

Studies in Computational Intelligence 862

Oscar Castillo
Patricia Melin
Janusz Kacprzyk *Editors*

Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications

 Springer

Studies in Computational Intelligence

Volume 862

Series Editor

Janusz Kacprzyk, Polish Academy of Sciences, Warsaw, Poland

The series “Studies in Computational Intelligence” (SCI) publishes new developments and advances in the various areas of computational intelligence—quickly and with a high quality. The intent is to cover the theory, applications, and design methods of computational intelligence, as embedded in the fields of engineering, computer science, physics and life sciences, as well as the methodologies behind them. The series contains monographs, lecture notes and edited volumes in computational intelligence spanning the areas of neural networks, connectionist systems, genetic algorithms, evolutionary computation, artificial intelligence, cellular automata, self-organizing systems, soft computing, fuzzy systems, and hybrid intelligent systems. Of particular value to both the contributors and the readership are the short publication timeframe and the world-wide distribution, which enable both wide and rapid dissemination of research output.

The books of this series are submitted to indexing to Web of Science, EI-Compendex, DBLP, SCOPUS, Google Scholar and Springerlink.

More information about this series at <http://www.springer.com/series/7092>

Oscar Castillo · Patricia Melin ·
Janusz Kacprzyk
Editors

Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications

 Springer

Editors

Oscar Castillo
Division of Graduate Studies and Research
Tijuana Institute of Technology
Tijuana, Baja California, Mexico

Patricia Melin
Division of Graduate Studies and Research
Tijuana Institute of Technology
Tijuana, Baja California, Mexico

Janusz Kacprzyk
Systems Research Institute
Polish Academy of Sciences
Warsaw, Poland

ISSN 1860-949X ISSN 1860-9503 (electronic)
Studies in Computational Intelligence
ISBN 978-3-030-35444-2 ISBN 978-3-030-35445-9 (eBook)
<https://doi.org/10.1007/978-3-030-35445-9>

© Springer Nature Switzerland AG 2020

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

We describe in this book, recent advances on fuzzy logic, neural networks, and optimization algorithms, as well as their hybrid combinations, and their application in areas, such as intelligent control and robotics, pattern recognition, medical diagnosis, time-series prediction, and optimization of complex problems. In particular, the enhancements of neural and optimization algorithms are given particular attention. The book is organized in ten main parts, which contain a group of papers around a similar subject. The first part consists of papers with the main theme of type-1 and type-2 fuzzy logic, which basically consists of papers that propose new concepts and algorithms based on type-1 and type-2 fuzzy logic and their applications. The second part contains papers with the main theme of intuitionistic fuzzy logic and applications, which are basically papers dealing with new concepts and algorithms in intuitionistic fuzzy logic applied in a wide variety of problems. The third part contains papers that present the theory and practice of metaheuristics in different areas of application. The fourth part presents diverse applications of fuzzy logic, neural networks, and hybrid intelligent systems in medical applications. The fifth part contains papers describing applications of fuzzy logic, neural networks, and metaheuristics in robotic problems. The sixth part contains papers that present the theory and practice of neural networks in different areas of application. The seventh part contains papers that present the theory and practice of optimization and evolutionary algorithms in different areas of application. The eighth part contains papers with the main theme of intelligent agents, which are basically papers dealing with new concepts and algorithms in intelligent agents applied in a wide variety of problems. The ninth part contains papers describing applications of fuzzy logic, neural networks, and metaheuristics in pattern recognition problems. The tenth part contains papers that present the theory and practice of hybrid intelligent systems in different areas of application.

In the first part of the book with five papers, we refer to theoretical aspects and applications of type-1 and type-2 fuzzy logic, which basically consists of papers that propose new concepts and algorithms based on type-1 and type-2 for different

applications. The aim of using type-2 fuzzy logic is to provide better uncertainty management in problems of control, pattern recognition, and prediction.

In the second part with the main theme of intuitionistic fuzzy logic and applications, there are eight papers that describe different contributions that propose new intuitionistic models, concepts, and algorithms for different applications. The aim of using intuitionistic fuzzy logic is to provide uncertainty handling capabilities to intelligent systems.

In the third part of the theory and practice of metaheuristics in different areas of application, there are four papers that describe different contributions that propose new models and concepts, which are also applied in diverse areas of application. The nature-inspired methods include variations of different methods as well as new nature-inspired paradigms.

In the fourth part presents diverse applications of fuzzy logic, neural networks, and hybrid intelligent systems in medical applications, there are four papers that describe different contributions on the application of these kinds of systems to solve complex real-world medical problems.

In the fifth part of fuzzy logic, neural networks, and metaheuristic applications in robotics, there are six papers that describe different contributions on the application of these kinds of intelligent models to solve complex real-world robotic problems.

In the sixth part of the theory and practice of neural networks in different areas of application, there are five papers that describe different contributions that propose new models and concepts, which are also applied in diverse areas of application. The neural network methods include variations of different methods as well as new neural paradigms.

In the seventh part of the theory and practice of optimization and evolutionary algorithms in different areas of application, there are five papers that describe different contributions that propose new models and concepts, which are also applied in diverse areas of application. The optimization and evolutionary algorithms include variations of different methods as well as new search paradigms.

In the eighth part with the main theme of intelligent agents, there are four papers that describe different contributions that propose new intelligent agents models, concepts, and algorithms for different applications. The aim of using intelligent agents is to provide increasing decision-making capabilities to intelligent systems.

In the ninth part presents diverse applications of fuzzy logic, neural networks, and hybrid intelligent systems in pattern recognition systems, there are five papers that describe different contributions on the application of these kinds of systems to solve complex real-world pattern recognition problems.

In the tenth part of the theory and practice of hybrid intelligent systems in different areas of application, there are six papers that describe different contributions that propose new models and concepts, which are also applied in diverse areas of application. The hybrid intelligent systems include variations of different methods as well as new paradigms.

In conclusion, the edited book comprises papers on diverse aspects of fuzzy logic, neural networks, and nature-inspired optimization metaheuristics and their application in areas, such as intelligent control and robotics, pattern recognition, time-series prediction, and optimization of complex problems. There are theoretical aspects as well as application papers.

Tijuana, Mexico
Tijuana, Mexico
Warsaw, Poland
July 2019

Oscar Castillo
Patricia Melin
Janusz Kacprzyk

Contents

Type-1 and Type-2 Fuzzy Logic

Parameter Adaptation in the Imperialist Competitive Algorithm Using Generalized Type-2 Fuzzy Logic	3
Emer Bernal, Oscar Castillo, José Soria and Fevrier Valdez	
Applying Fuzzy Logic to Identify Heterogeneity of the Allometric Response in Arithmetical Space	11
Cecilia Leal-Ramírez, Héctor Echavarría-Heras and Enrique Villa-Diharce	
Fireworks Algorithm (FWA) with Adaptation of Parameters Using Interval Type-2 Fuzzy Logic System	35
Juan Barraza, Fevrier Valdez, Patricia Melin and Claudia I. González	
Omnidirectional Four Wheel Mobile Robot Control with a Type-2 Fuzzy Logic Behavior-Based Strategy	49
Felizardo Cuevas, Oscar Castillo and Prometeo Cortés-Antonio	
Optimization for Type-1 and Interval Type-2 Fuzzy Systems for the Classification of Blood Pressure Load Using Genetic Algorithms	63
Juan Carlos Guzmán, Patricia Melin and German Prado-Arechiga	

Intuitionistic Fuzzy Logic

Interval Valued Intuitionistic Fuzzy Evaluations for Analysis of Students' Knowledge	75
Evdokia Sotirova, Anthony Shannon, Vassia Atanassova, Krassimir Atanassov and Veselina Bureva	
Generalized Net Model of the Network for Automatic Turning and Setting the Lighting in the Room with Intuitionistic Fuzzy Estimations	83
Tihomir Videv, Sotir Sotirov and Boris Bozveliev	

Generalized Net Model of Common Internet Payment Gateway with Intuitionistic Fuzzy Estimations	91
Boris Bozveliev, Sotir Sotirov, Stanislav Simeonov and Tihomir Videv	
Intuitionistic Fuzzy Neural Networks with Interval Valued Intuitionistic Fuzzy Conditions	99
Krassimir Atanassov, Sotir Sotirov and Nora Angelova	
Generalised Atanassov Intuitionistic Fuzzy Sets Are Actually Intuitionistic Fuzzy Sets	107
Peter Vassilev and Krassimir Atanassov	
The Numerical Solution of Intuitionistic Fuzzy Differential Equations by the Third Order Runge-Kutta Nyström Method	119
Bouchra Ben Amma, Said Melliani and S. Chadli	
Intuitionistic Fuzzy Linear Systems	133
Hafida Atti, Bouchra Ben Amma, Said Melliani and S. Chadli	
Nonlocal Intuitionistic Fuzzy Differential Equation	145
R. Ettoussi, Said Melliani and S. Chadli	
Metaheuristics: Theory and Applications	
Harmony Search with Dynamic Adaptation of Parameters for the Optimization of a Benchmark Controller	157
Cinthia Peraza, Fevrier Valdez and Oscar Castillo	
Evaluation of Parallel Exploration and Exploitation Capabilities in Two PSO Variants with Intra Communication	169
Yunkio Kawano, Fevrier Valdez and Oscar Castillo	
Chemical Reaction Algorithm to Control Problems	185
David de la O, Oscar Castillo and José Soria	
AMOSA with Analytical Tuning Parameters and Fuzzy Logic Controller for Heterogeneous Computing Scheduling Problem	195
Héctor J. Fraire Huacuja, Carlos Soto, Bernabé Dorronsoro, Claudia Gómez Santillán, Nelson Rangel Valdez and Fausto Balderas-Jaramillo	
Medical Applications	
A Modular Neural Network Approach for Cardiac Arrhythmia Classification	211
Eduardo Ramírez, Patricia Melin and German Prado-Arechiga	
Particle Swarm Optimization of Modular Neural Networks for Obtaining the Trend of Blood Pressure	225
Ivette Miramontes, Patricia Melin and German Prado-Arechiga	

Classification of X-Ray Images for Pneumonia Detection Using Texture Features and Neural Networks 237
 Sergio Varela-Santos and Patricia Melin

Segmentation and Classification of Noisy Thermographic Images as an Aid for Identifying Risk Levels of Breast Cancer 255
 Pilar Gomez-Gil, Daniela Reynoso-Armenta, Jorge Castro-Ramos, Juan Manuel Ramirez-Cortes and Vicente Alarcon-Aquino

Robotic Applications

Acceleration of Path Planning Computation Based on Evolutionary Artificial Potential Field for Non-static Environments 271
 Ulises Orozco-Rosas, Kenia Picos and Oscar Montiel

Multi-objective Evaluation of Deep Learning Based Semantic Segmentation for Autonomous Driving Systems 299
 Cynthia Olvera, Yoshio Rubio and Oscar Montiel

Towards Tracking Trajectory of Planar Quadrotor Models 313
 Prometeo Cortés-Antonio, Fevrier Valdez, Oscar Castillo and Patricia Melin

Autonomous Garage Parking of a Car-Like Robot Using a Fuzzy PD + I Controller 325
 Enrique Ballinas, Oscar Montiel and Yoshio Rubio

Analysis of P, PI, Fuzzy and Fuzzy PI Controllers for Control Position in Omnidirectional Robots 339
 Leticia Luna-Lobano, Prometeo Cortés-Antonio, Oscar Castillo and Patricia Melin

Fuzzy Logic Controller with Fuzzylab Python Library and the Robot Operating System for Autonomous Robot Navigation: A Practical Approach. 355
 Eduardo Avelar, Oscar Castillo and José Soria

Neural Networks Applications

Neural Evolutionary Predictive Control for Linear Induction Motors with Experimental Data 373
 Alma Y. Alanis, Nancy Arana-Daniel, Carlos Lopez-Franco and Jorge D. Rios

Filter Size Optimization on a Convolutional Neural Network Using FGSA 391
 Yutzil Poma, Patricia Melin, Claudia I. González and Gabriela E. Martínez

Evaluation and Analysis of Performances of Different Heuristics for Optimal Tuning Learning on Mamdani Based Neuro-Fuzzy System 405
 Sukey Nakasima-López, Mauricio A. Sanchez and Juan R. Castro

Direct and Indirect Evolutionary Designs of Artificial Neural Networks 431
 O. Alba-Cisneros, A. Espinal, G. López-Vázquez, M. A. Sotelo-Figueroa, O. J. Purata-Sifuentes, V. Calzada-Ledesma, R. A. Vázquez and H. Rostro-González

Studying Grammatical Evolution’s Mapping Processes for Symbolic Regression Problems 445
 B. V. Zuñiga-Nuñez, J. Martín Carpio, M. A. Sotelo-Figueroa, J. A. Soria-Alcaraz, O. J. Purata-Sifuentes, Manuel Ornelas and A. Rojas-Domínguez

Optimization and Evolutionary Algorithms

A Survey of Hyper-heuristics for Dynamic Optimization Problems 463
 Teodoro Macias-Escobar, Bernabé Dorronsoro, Laura Cruz-Reyes, Nelson Rangel-Valdez and Claudia Gómez-Santillán

The Dynamic Portfolio Selection Problem: Complexity, Algorithms and Empirical Analysis 479
 Daniel A. Martínez-Vega, Laura Cruz-Reyes, Claudia Guadalupe Gomez-Santillan, Fausto Balderas-Jaramillo and Marco Antonio Aguirre-Lam

A Novel Dynamic Multi-objective Evolutionary Algorithm with an Adaptable Roulette for the Selection of Operators 493
 Héctor Joaquín Fraire Huacuja, Eduardo Rodríguez del Angel, Juan Javier González Barbosa, Alejandro Estrada Padilla and Lucila Morales Rodríguez

Combinatorial Designs on Constraint Satisfaction Problem (VRP) 509
 Juan A. Montesino-Guerra, Héctor Puga, J. Martín Carpio, Manuel Ornelas-Rodríguez, A. Rojas-Domínguez and Lucero Ortiz-Aguilar

Comparative Analysis of Multi-objective Metaheuristic Algorithms by Means of Performance Metrics to Continuous Problems 527
 Carlos Juarez-Santini, J. A. Soria-Alcaraz, M. A. Sotelo-Figueroa and Esteves-Jiménez Velino

Intelligent Agents

Towards an Agent-Based Model for the Analysis of Macroeconomic Signals 551
 Alejandro Platas-López, Alejandro Guerra-Hernández, Nicandro Cruz-Ramírez, Marcela Quiroz-Castellanos, Francisco Grimaldo, Mario Paolucci and Federico Cecconi

Fuzzy Worlds and the Quest for Modeling Complex-Adaptive Systems 567
 Miguel Melgarejo

Procedural Generation of Levels for the Angry Birds Videogame Using Evolutionary Computation 581
 Jaime Salinas-Hernández and Mario Garcia-Valdez

A Multi-agent Environment Acting as a Personal Tourist Guide 593
 Asya Stoyanova-Doycheva, Todorka Glushkova, Vanya Ivanova, Lyubka Doukovska and Stanimir Stoyanov

Pattern Recognition

Comparing Evolutionary Artificial Neural Networks from Second and Third Generations for Solving Supervised Classification Problems 615
 G. López-Vázquez, A. Espinal, Manuel Ornelas-Rodríguez, J. A. Soria-Alcaraz, A. Rojas-Domínguez, Héctor Puga, J. Martín Carpio and H. Rostro-González

Gegenbauer-Based Image Descriptors for Visual Scene Recognition 629
 Antonio Herrera-Acosta, A. Rojas-Domínguez, J. Martín Carpio, Manuel Ornelas-Rodríguez and Héctor Puga

Bimodal Biometrics Using EEG-Voice Fusion at Score Level Based on Hidden Markov Models 645
 Juan Carlos Moreno-Rodríguez, Juan Manuel Ramirez-Cortes, Rene Arechiga-Martinez, Pilar Gomez-Gil and Juan Carlos Atenco-Vazquez

Towards a Quantitative Identification of Mobile Social Media UIDPs' Visual Features Using a Combination of Digital Image Processing and Machine Learning Techniques 659
 Viviana Yarel Rosales-Morales, Nicandro Cruz-Ramírez, Laura Nely Sánchez-Morales, Giner Alor-Hernández, Marcela Quiroz-Castellanos and Efrén Mezura-Montes

Fuzzy Modular Neural Model for Blinking Coding Detection and Classification for Linguistic Expression Recognition 675
 Mario I. Chacon-Murguía, Carlos E. Cañedo-Figueroa and Juan A. Ramirez-Quintana

Hybrid Intelligent Systems

A Genetic Algorithm Based Approach for Word Sense Disambiguation Using Fuzzy WordNet Graphs 693
 Sonakshi Vij, Amita Jain and Devendra Tayal

Configuration Module for Treating Design Anomalies in Databases for a Natural Language Interface to Databases 703
 Grigori Sidorov, Rodolfo A. Pazos R., José A. Martínez F., J. Martín Carpio and Alan G. Aguirre L.

Development of a Virtual View for Processing Complex Natural Language Queries 715
 José A. Martínez F., Rodolfo A. Pazos R., Héctor Puga and Juana Gaspar H.

Automated Ontology Extraction from Unstructured Texts using Deep Learning 727
 Raúl Navarro-Almanza, Reyes Juárez-Ramírez, Guillermo Licea and Juan R. Castro

Implementation of a Multicriteria Analysis Model to Determine Anthropometric Characteristics of an Optimal Helmet of an Italian Scooter 757
 Josué Cuevas, Alberto Ochoa, Juan Luis Hernandez, José Mejía, Liliana Avelar and Boris Mederos

Improving Segmentation of Liver Tumors Using Deep Learning 771
 José Mejía, Alberto Ochoa and Boris Mederos

Intuitionistic Fuzzy Sugeno Integral for Face Recognition 781
 Gabriela E. Martínez and Patricia Melin

Type-1 and Type-2 Fuzzy Logic

Parameter Adaptation in the Imperialist Competitive Algorithm Using Generalized Type-2 Fuzzy Logic



Emer Bernal, Oscar Castillo, José Soria and Fevrier Valdez

Abstract In this paper we propose the use of generalized type-2 fuzzy systems to dynamic adjustment the parameters of the imperialist competitive algorithm (ICA), and we take a type-1 fuzzy system as a basis to extend our proposal using generalized type-2 fuzzy logic. The ICA algorithm is based on the concept of imperialism in which the strongest countries try to take control of the weakest countries. In order to measure the performance of our proposed method different benchmark functions were used and finally, a comparison was made between the variants to observe their behavior applied to benchmark mathematical functions.

Keywords Imperialist competitive algorithm · Fuzzy system · Benchmark function · Generalized type-2 · Fuzzy logic

1 Introduction

Zadeh is considered the father of fuzzy logic and fuzzy sets. Fuzzy sets represent the imprecision of traditional sets assigning numerical values in the $[0,1]$ interval, in traditional sets it is difficult to establish or find an exact value for some type of measurement [1, 2].

Fuzzy systems are built on the basis of rules formed through human knowledge and heuristics based on human experience. Fuzzy logic is an effective tool for information management in rule-based systems due to its tolerance for imprecision and lack of information [2–5].

When the problem or situation contains a high degree of uncertainty, type-2 fuzzy systems can be used because they work better with high levels of uncertainty or lack of information [5, 6].

Type-2 fuzzy systems have emerged as a generalization of type-1 fuzzy systems based upon fuzzy sets [7, 8]. The main advantage of using generalized type-2 fuzzy

E. Bernal · O. Castillo (✉) · J. Soria · F. Valdez
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: ocastillo@tectijuana.mx

systems is that show a higher degree of stability and are more capable of handling uncertainty [9].

We propose the dynamic adjustment of parameters in the imperialist competitive algorithm using generalized type-2 fuzzy systems. A fuzzy systems was designed to control the beta parameter and to measure the performance of our proposed method fuzzy imperialist competitive algorithm (FICAGT2) using generalized type-2 fuzzy systems. We used a set of benchmark mathematical functions to test the proposed method. Fuzzy Logic has been previously used in other metaheuristics [1, 9, 10], to improve on methods with fixed parameters [6, 11–13].

The rest of the paper is structured as follows. Section 2 provides a review of imperialist competitive algorithm. Section 3 presents the methodology used in the proposed method. Section 4 presents the mathematical benchmark functions and the results obtained and finally Sect. 5 offers the conclusions.

2 Imperialist Competitive Algorithm

Atashpaz-Gargari and Lucas proposed the imperialist competitive algorithm (ICA) in 2007 [14, 15]. The algorithm is inspired by imperialism, where all the most powerful countries aspire to make a colony the less powerful countries and thus absorb them. In the area of metaheuristics, the imperialist competitive algorithm takes as a basis the social political progress unlike other metaheuristics or evolutionary algorithms [16–18].

In the ICA algorithm, a randomly generated population is used, in this context the individuals are considered as countries. The countries that have greater power will be considered as imperialist countries and the rest as imperialist colonies. The colonies are divided among the imperialist countries based on their power or fitness value. [16].

After the population was divided into imperialist and colonies, the colonies begin to move towards the imperialist country and this is known as assimilation process. The colonies move x distance towards their imperialist country, where x is a random number that is generated to hear a uniform distribution in the interval $(0, \beta d)$.

$$x \sim U(0, \beta d) \quad (1)$$

where β is a number in the range of 1–2 and d is the distance between the colony and its imperialist country [15, 18] (Fig. 1).

The power of an empire is calculated by the power of its imperialist and the power of its colonies. Each empire will try to take possession of other empires, this causes an imperialist competition in which weaker empires diminish in power and stronger empires increase in power, this causes weaker empires to collapse and their colonies become part of the strongest empires [17].

After a period of time all the empires began to collapse little by little, except the strongest empire and so all the colonies of the other empires will belong and

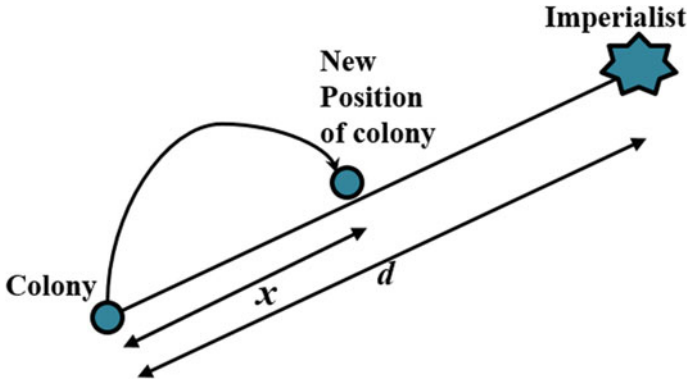


Fig. 1 Movement of the colonies toward the imperialist

be controlled by an imperialist country and this leads to the end of the imperialist competition [15].

3 Proposal Methodology

We propose a method based on generalized type-2 fuzzy systems for dynamic adjustment the parameters of the imperialist competitive algorithm. A type-1 fuzzy system as a basis to extend our proposal using generalized type-2 fuzzy systems was used [18].

The fuzzy system taken as a base consists of an input variable labeled as Decades and with the Beta as output variable, each of the variables are granulated with three triangular membership functions labeled “Low”, “Medium” and “High” [1].

The proposal based on generalized type 2 fuzzy systems in the competitive imperialist algorithm for dynamic adaptation the parameters can be seen in Fig. 2.

The proposal using generalized type-2 fuzzy system is designed with Gaussian membership functions that are shown in Fig. 3 [7–9].

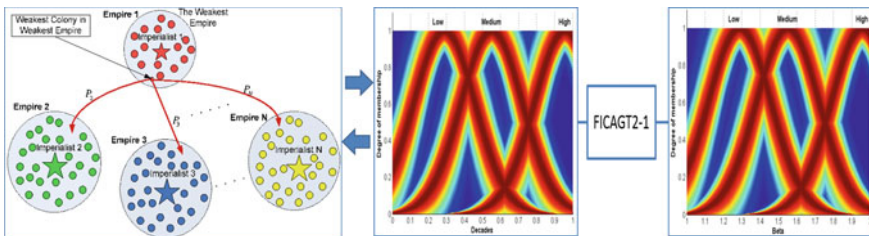


Fig. 2 Proposed approach

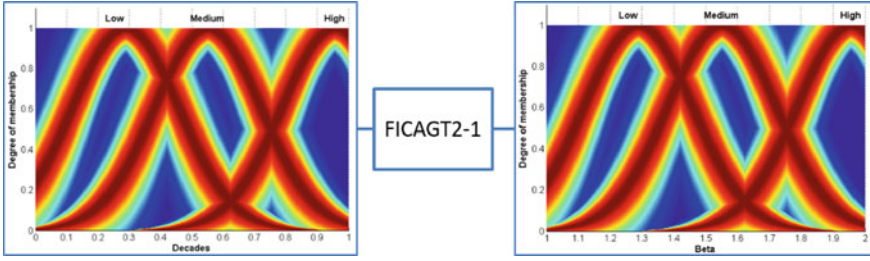


Fig. 3 FICAGT2-1 generalized type-2 fuzzy system

The rules of generalized type 2 fuzzy systems are the same as those used in the type-1 fuzzy system. Where main idea is to start exploring and as the decades progress, exploit as shown below.

Rules of the fuzzy system for Beta:

1. If (Decades is Low) then (Beta is Low).
2. If (Decades is Medium) then (Beta is Medium).
3. If (Decades is High) then (Beta is High).

4 Benchmark Mathematical Functions and the Experimental Results

The use of benchmark mathematical functions to test the performance of metaheuristic algorithms is very common [19–24]. In this work they are used to measure the performance of the proposal based on generalized type 2 fuzzy system in the ICA algorithm for dynamic adjustment of the beta parameter.

In this paper we used 10 mathematical functions to measure the performance of our proposal. Below are listed with their mathematical expression, optimal value and search space [1].

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad (2)$$

with $x_j \in [-5.12, 5.12]$ and $f(x^*) = 0$

$$f_2(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i| \quad (3)$$

with $x_j \in [-10, 10]$ and $f(x^*) = 0$

$$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i x_j \right)^2 \quad (4)$$

with $x_j \in [-100, 100]$ and $f(x^*) = 0$

$$f_4(x) = \max_i \{|x_i|, 1 \leq i \leq n\}$$

with $x_j \in [-100, 100]$ *and* $f(x^*) = 0$

(5)

$$f_5(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$$

with $x_j \in [-30, 30]$ *and* $f(x^*) = 0$

(6)

$$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$$

with $x_j \in [-100, 100]$ *and* $f(x^*) = 0$

(7)

$$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random} [0, 1]$$

with $x_j \in [-1.28, 1.28]$ *and* $f(x^*) = 0$

(8)

$$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{|x_i|})$$

with $x_j \in [-500, 500]$ *and* $f(x^*) = -418.9829$

(9)

$$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$$

with $x_j \in [-5.12, 5.12]$ *and* $f(x^*) = 0$

(10)

$$f_{10}(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$$

with $x_j \in [-32, 32]$ *and* $f(x^*) = 0$

(11)

The results obtained by the ICA algorithm and our proposal are shown in separate tables with the respect to the number of decades, respectively. Each table contains the average obtained after 30 executions.

Values for the parameters used with the ICA algorithm with fixed parameters and the proposed FICAGT2 with parameter adaptation using interval type-2 fuzzy systems.

Table 1 shows the values for the parameters used during the experiments with the ICA algorithm, FICA and the proposal FICAGT2 for 1000, 2000 and 3000 decades.

Table 1 Values of the parameters for ICA, FICA and FICAGT2

Parameters	ICA	FICA	FICAGT2
Dimensions	30	30	30
No. countries	200	200	200
No.i	10	10	10
Revolution rate	0.2	0.2	0.2
Xi	0.02	0.02	0.02
Beta	1.4	Dynamic	Dynamic

Table 2 Results for 1000 decades

Function	ICA [1]	FICA [1]	FICAGT2-1
f1	1.12E-23	8.79E-25	6.9596E-25
f2	3.39E-56	4.41E-52	1.8227E-50
f3	5822.35897	5909.17803	3886.2518
f4	8.60023687	0.83147595	2.87315123
f5	24.1293542	171.579304	293.879952
f6	6.59E-17	1.06E-22	4.22E-16
f7	9.76E-41	11.6622472	11.7137065
f8	2535.76878	2195.79881	2150.82545
f9	123.807243	101.011796	89.6214181
f10	13.0151646	12.90137	11.1177291

Bold indicates best results

Table 3 Results for 2000 decades

Function	ICA [1]	FICA [1]	FICAGT2-1
f1	9.26E-51	3.24E-51	3.6695E-46
f2	2.22E-141	4.42E-116	4.1E-112
f3	7928.65015	3638.93304	8307.63181
f4	0.61591754	0.0102226	0.08621043
f5	11.799352	8.03929545	429.544444
f6	1.06E-17	2.47E-20	1.93E-16
f7	9.95E-92	11.2356722	11.3343513
f8	2349.08987	1889.56348	2043.27965
f9	114.795875	70.1576847	40.9412707
f10	12.8467107	12.9715618	11.7740274

Bold indicates best results

Table 2 shows the results of the ICA algorithm and our proposal FICAGT-2 and we can observe the average of the experiments for all the functions with 1000 decades.

Table 3 shows the results of the ICA algorithm and our proposal FICAGT-2 and we can observe the average of the experiments for all the functions with 2000 decades.

Table 4 shows the results of the ICA algorithm and our proposal FICAGT-2 and we can observe the average of the experiments for all the functions with 3000 decades.

5 Conclusions

We can conclude that some significant improvements have been obtained with type-2 fuzzy logic, that the proposed method has better performance than the original ICA algorithm in some test functions as shown in Tables 2, 3 and 4.

Table 4 Results for 3000 decades

Function	ICA [1]	FICA [1]	FICAGT2-1
f1	1.18E−29	1.51E−75	1.7274E−17
f2	7.50E−231	3.19E−183	5.282E−165
f3	6863.925722	7563.274419	5979.00292
f4	0.010497636	0.000138756	0.00266397
f5	13.04971328	9.935683436	554.315711
f6	5.40E−19	1.86E−19	1.07E−14
f7	9.87E−144	10.95189055	11.1528643
f8	2345.035647	1804.997716	2062.16087
f9	114.289348	68.03826234	56.660041
f10	15.34792029	11.98271395	11.8930438

Bold indicates best results

The use of generalized type-2 fuzzy systems has shown to be a good alternative to achieve improvements to dynamic adjustment the parameters of the imperialist competitive algorithm.

Acknowledgements We would like to express our gratitude to the CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Bernal, E., Castillo, O., Soria, J.: Fuzzy logic for dynamic adaptation in the imperialist competitive algorithm. In: IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7. IEEE (2017)
2. Melin, P., Olivas, F., Castillo, O., Valdez, F., Soria, J., Valdez, M.: Optimal design of fuzzy classification systems using PSO with dynamic parameter adaptation through fuzzy logic. *Expert. Syst. Appl.* **40**(8), 3196–3206 (2013)
3. Jang, J.S.R., Sun, C.T., Mizutani, E.: *Neuro-fuzzy and soft computing: a computational approach to learning and machine intelligence* (1997)
4. Bernal, E., Castillo, O., Soria, J., Valdez, F.: A variant to the dynamic adaptation of parameters in galactic swarm optimization using a fuzzy logic augmentation. In: Proceedings of the 2018 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1–7 (2018)
5. Mahmoodabadi, M.J., Jahanshahi, H.: Multi-objective optimized fuzzy-PID controllers for fourth order nonlinear systems. *Eng. Sci. Technol. Int. J.* **18**, 1084–1098 (2016)
6. Milajić, A., Beljaković, D., Davidović, N., Vatin, N., Murgul, V.: Using the big bang-big crunch algorithm for rational design of an energy-plus building. *Procedia Eng.* **117**, 916–923 (2015)
7. Sanchez, M.A., Castillo, O., Castro, J.R., Melin, P.: Fuzzy granular gravitational clustering algorithm for multivariate data. *Inf. Sci.* **279**, 498–511 (2014)
8. Sanchez, M.A., Castillo, O., Castro, J.R.: Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. *Expert. Syst. Appl.* **42**(14), 5904–5914 (2015)
9. Castillo, O., Amador-Angulo, L., Castro, J.R., Garcia-Valdez, M.: A comparative study of type-1 fuzzy logic systems, interval type-2 fuzzy logic systems and generalized type-2 fuzzy logic systems in control problems. *Inf. Sci.* **354**, 257–274 (2016)

10. Bernal, E., Castillo, O., Soria, J.: A fuzzy logic approach for dynamic adaptation of parameters in galactic swarm optimization. In: Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS), pp. 1–6, IEEE (2017)
11. Cheng, M.Y., Prayogo, D.: Symbiotic organisms search: a new metaheuristic optimization algorithm. *Comput. Struct.* **139**, 98–112 (2014)
12. Engelbrecht, A.P.: *Computational Intelligence*. Wiley, Pretoria, South Africa (2007)
13. Mitchell, M.: *An Introduction to Genetic Algorithms*. MIT Press, Cambridge (1999)
14. Atashpaz-Gargari, E., Hashemzadeh, F., Rajabioun, R., Lucas, C.: C. Colonial competitive algorithm: a novel approach for PID controller design in MIMO distillation column process. *Int. J. Intell. Comput. Cybern.* **1**, 337–355 (2008)
15. Atashpaz-Gargari, E., Lucas, C.: Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition. *Evol. comput.* 4661–4667 (2007)
16. Bernal, E., Castillo, O., Soria, J.: Imperialist competitive algorithm applied to the optimization of mathematical functions: a parameter variation study. In: *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization*, vol. 601, pp. 219–232, Springer International Publishing, Berlin (2015)
17. Bernal, E., Castillo, O., Soria, J.: Imperialist competitive algorithm with dynamic parameter adaptation applied to the optimization of mathematical functions. In: *Nature-Inspired Design of Hybrid Intelligent Systems*, vol. 667, pp. 329–341. Springer International Publishing, Berlin (2017)
18. Bernal, E., Castillo, O., Soria, J., Valdez, F.: Imperialist competitive algorithm with dynamic parameter adaptation using fuzzy logic applied to the optimization of mathematical functions. *Algorithms* **10**(1), 18 (2017)
19. Haupt, R.L., Haupt, S.E.: *Practical Genetic Algorithms*, 2nd edn. Wiley, Hoboken, NJ, USA (2004)
20. Hedar, A.R.: Test functions for unconstrained global optimization. [Online] Egypt, Assiut University, Available http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedar_files/TestGO.htm
21. Muthiah-Nakarajan, V., Noel, M.M.: Galactic swarm optimization: a new global optimization metaheuristic inspired by galactic motion. *Appl. Soft Comput.* **38**, 771–787 (2016)
22. Oftadeh, R., Mahjoob, M.J., Shariatpanahi, M.: A novel meta-heuristic optimization algorithm inspired by group hunting of animals: hunting search. *Comput. Math Appl.* **60**(7), 2087–2098 (2010)
23. Sedighzadeh, M., Bakhtiary, R.: Optimal multi-objective reconfiguration and capacitor placement of distribution systems with the hybrid big bang-big crunch algorithm in the fuzzy framework. *Ain. Shams. Eng. J.* **7**, 113–129 (2016)
24. Valdez, F., Melin, P., Castillo, O.: An improved evolutionary method with fuzzy logic for combining particle swarm optimization and genetic algorithms. *Appl. Soft. Comput.* **11**(2), 2625–2632 (2011)

Applying Fuzzy Logic to Identify Heterogeneity of the Allometric Response in Arithmetical Space



Cecilia Leal-Ramírez, Héctor Echavarría-Heras and Enrique Villa-Diharce

Abstract Customary approaches in allometric examination include linear regression in geometrical space, as well as, nonlinear regression in the original scale of data. These protocols could not produce consistent results in a circumstance in which the allometric response manifest heterogeneity as the covariate changes. The paradigm of log-nonlinear allometry offers a mechanism for the analysis of heterogeneity in geometric space. However, the use of a logarithmic transformation in allometry is controversial. In this contribution, we present a fuzzy approach aimed to examination of allometric heterogeneity in direct arithmetical space. Offered construct relies on a hybrid procedure integrating crisp cluster analysis and a fuzzy inference system of Mamdani type. Calibration aims depended on an extensive data set composing measurements of eelgrass leaf biomass and their corresponding areas. Results on raw data suggest heterogeneity more clearly manifest in the normalization constant than in the allometric exponent. Nevertheless, differences in normalization constant values among clusters are only slight for data remaining after removal of inconsistent replicates. This suggests heterogeneity produced by intrinsic factors of leaf growth.

Keywords Allometry · Heterogeneity · Mamdani type fuzzy model · Average leaf biomass eelgrass

C. Leal-Ramírez · H. Echavarría-Heras (✉)
Centro de Investigación Científica y de Educación Superior de Ensenada, Carretera
Ensenada-Tijuana, No 3918, Zona Playitas, Ensenada 22860, BC, Mexico
e-mail: hetxavar@cicese.mx

C. Leal-Ramírez
e-mail: cleal@cicese.mx

E. Villa-Diharce
Centro de Investigación en Matemáticas, A.C. Jalisco s/n, Mineral Valenciana, Guanajuato Gto
36240, Mexico
e-mail: villadi@cimat.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_2

1 Introduction

An allometric relationship between measured quantities w and a is commonly expressed through a power function of the form

$$w = \beta a^\alpha, \quad (1)$$

with α and β referred as the allometric parameters α known as the allometric exponent and β termed normalization constant. This model appears in research problems of different knowledge areas, like physics, biology, resource management and earth sciences [1–9].

The model of Eq. (1) allows convenient estimations of the dry weight (biomass) of a plant unit w based on measurements of a trait a such as length, width or diameter of suitable plant components [10]. For instance, estimates of the parameters α and β in the model of Eq. (1) and measurements of leaf area a produce indirect estimations of eelgrass leaf biomass w [11]. Moreover, for eelgrass, estimates of the parameters α and β are time invariant within a given geographical region [12]. Thus, allometric projections for leaf biomass w gotten from Eq. (1) by means of available parameter estimates and currently measured values of leaf area a are truly nondestructive. This way, we can derive concomitant surrogates for aggregated variables such as mean leaf biomass in shoots. These are essential in assessment of an eelgrass population [12–14]. But, on spite of statistical invariance, environmental influences could induce a relative extent of variability on local estimates of α and β [12]. Besides, from a general standpoint the value of the response given the covariate in a scaling relationship such as Eq. (1) is very sensitive on variations of the parameters α and β . This could propagate uncertainties on allometric projections of individual eelgrass leaf biomass w to such an extent, that related surrogates of mean leaf biomass in shoots could get significantly biased [11]. But, besides uncertainties on parameter estimates owing to environmental fluctuation, there are factors on methodological grounds that can also influence precision. We refer to factors like, data quality, sample size, and analysis method [15–17].

There are also effects on suitability of allometric surrogates that could associate to insufficiency of the contemplated scaling model. For instance, complexity inherent in Eq. (1) could not provide a consistent fit in the presence of curvature as it is understood in allometric examination. This bias inducing circumstance refers to non-log linear allometry [18]. This allows consideration of heterogeneity of the log-transformed response as its covariate goes through different growth stanzas [19, 20]. Customarily, this paradigm associates to polynomial or piecewise-linear-regression protocols in geometrical space. Nevertheless, there are views asserting that, the use of log-transformed data leads to biased results, and that direct nonlinear regression in allometric examination should be preferred [17, 21]. In any event, a statistical approach in dealing with complex allometries could be modeling the conditional distribution of the response given the descriptor as a mixture. This way, the covariate domain can be clustered and associated to local regression models concurrently fitted.

Moreover, fuzzy logic provides a non-probabilistic interpretation of uncertainty [22, 23]. The assembly of a Takagi-Sugeno-Kang (TSK) fuzzy model [24, 25] involves a partitioning of the covariate domain into a number of fuzzy subdomains. Each cluster or partition region associates to a fuzzy IF-THEN rule [26] that specifies a local model as an output. Echavarría-Heras et al. [27] illustrates the use of the Takagi-Sugeno-Kang device to model heterogeneity in the allometric response in arithmetical space. Nevertheless, this approach identifies a mean response function $f(a)$ for w , which does not allow to characterize heterogeneity as established by scaling parameters taking different values over a composite of growth stanzas. In this revision, we extend these methods in order to consider an analytical construct that could identify an allometric paradigm based on a collection of sub models of the form given by Eq. (1) that allows contemplating heterogeneity in the parameters α and β . Estimation of named allometric parameters is carried away by means of a hybrid procedure that combines crisp cluster analysis and a fuzzy inference system of Mamdani type. The present approach, considers parameters as fuzzy sets. Moreover, both the allometric response and its covariate can fit in fuzzy sets according to different degrees of membership. This allows to consider influences due to individual contribution of the response and the covariate. Calibration of the procedure relied on *Zostera marina* leaf biomass and area data collected in San Quintin Bay Baja California and reported in [9].

In Sect. 2, we describe involved data sets and the formal set up backing the present adaptation of the Mamdani fuzzy inference system for allometric examination. Section 3 explains the performance of the fuzzy allometric paradigm compared to nonlinear regression of the customary bivariate allometric model in direct arithmetical space. A discussion section elaborates on advantages and weakness of the present fuzzy approach. A final section summarizes gains derived from the present approach and suggest matters of further research.

2 Methods

2.1 Data

For the present research, we relied on data composing individual measurements of leaf biomass (g) corresponding length (mm) and width (mm) obtained from 10,412 individual eelgrass leaves collected in San Quintin Bay Baja California as previously reported in [9]. Leaf area (mm^2) was indirectly obtained by using the length times width proxy [27]. In what follows this primary data will be referred as the raw data set. Removal of inconsistent replicates according to the Median Absolute Deviation (MAD) procedure described in [9] supplied a quality controlled data set that we further on refer as the processed data set. This was used for comparison aims.

2.2 Notions of Fuzzy Set Theory

We present notions of fuzzy set theory that support our construct. For further reading we recommend Refs. [28–30].

We denote by means of the symbol U the universal set containing all probable elements that are relevant in a field of interest. A fuzzy subset \mathbf{A} of U is typified by its membership function μ_A . This ascribes to an element x of U a number $\mu_A(x)$ in the interval $[0, 1]$. Then, $\mu_A(x)$ provides the grade of membership of x in \mathbf{A} . An ordinary subset \mathbf{A} of U is a fuzzy set for which the membership function is the indicator function of \mathbf{A} defined by: $\mu_A: U \rightarrow \{0, 1\}$, where

$$\mu_A(x) = \begin{cases} 1 & \text{if } x \in \mathbf{A} \\ 0 & \text{otherwise} \end{cases}. \quad (2)$$

A triangular fuzzy number is a fuzzy subset $\mathbf{A} \subset \mathbb{R}$ characterized by a membership function $\mu_A: \mathbb{R} \rightarrow [0, 1]$ defined by:

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ \frac{x-a}{m-a} & \text{if } a \leq x \leq m \\ \frac{b-x}{b-m} & \text{if } m \leq x \leq b \\ 0 & \text{if } x \geq b \end{cases}, \quad (3)$$

where a , b and m are real numbers such that $a < m < b$. The numbers a and b are called the lower and upper limits and m is known as the modal value defining the triangular fuzzy number, which is symbolically represented by $\tilde{A} = [a, b, m]$.

A Gaussian fuzzy number is a fuzzy subset $\mathbf{A} \subset \mathbb{R}$ characterized by a membership function $\mu_A: \mathbb{R} \rightarrow [0, 1]$ defined by

$$\mu_A(x) = e^{-c(x-m)^2}, \quad (4)$$

both m and k are real numbers, m is known as the medium value and k is called the opening of the bell. Moreover, m and k are sufficient to define the Gaussian fuzzy number, which is symbolically represented by means of $\tilde{A} = [c, k]$.

A *H-shaped* fuzzy number is a fuzzy subset $\mathbf{A} \subset \mathbb{R}$ characterized by a membership function $\mu_A: \mathbb{R} \rightarrow [0, 1]$ defined as follows:

$$\mu_A(x) = \begin{cases} 0 & \text{if } x \leq a \\ 2\left[\frac{x-a}{b-a}\right]^2 & \text{if } a \leq x \leq \frac{a+b}{2} \\ 1 - 2\left[\frac{x-b}{b-a}\right]^2 & \text{if } \frac{a+b}{2} \leq x \leq b \\ 1 & \text{if } b \leq x \leq c \\ 1 - 2\left[\frac{x-c}{d-c}\right]^2 & \text{if } c \leq x \leq \frac{c+d}{2} \\ 2\left[\frac{x-d}{d-c}\right]^2 & \text{if } \frac{c+d}{2} \leq x \leq d \\ 0 & \text{if } x \geq d \end{cases}, \quad (5)$$

where the parameters a, b, c and d are all real numbers. These suffice in defining the respective H -shaped fuzzy number, which is denoted through $\tilde{A} = [a, b, c, d]$.

Let $f: R \rightarrow R$ be an ordinary real function and \mathbf{A} be a fuzzy subset of R . Then, the **image** of the fuzzy set \mathbf{A} under the function f is the fuzzy subset \mathbf{B} of \mathbf{R} whose membership function is defined by

$$\mu_B(x) = \begin{cases} \max\{\mu_A(x)\} & \text{if } \{x|y = f(x)\} \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Let $A \subset U$ be a fuzzy set of U defined by the membership function $\mu_A: U \rightarrow [0, 1]$. The support set of A is defined as the crisp set that contains all the elements of U that have nonzero membership values in A , that is,

$$support(A) = \{x \in U; \mu_A(x) > 0\}. \tag{7}$$

We define the π -area of a fuzzy set A as a set $|A|^\pi$ containing all points (x, z) in U given by

$$|A|^\pi = \left\{ (x, z) \left| \begin{array}{l} z \leq \mu_A(x) \text{ if } \mu_A(x) \leq \pi \\ z \leq \pi \quad \text{if } \mu_A(x) > \pi \end{array} \right. \right\}. \tag{8}$$

For fuzzy sets A and B on a reference set U , typified by corresponding membership function $\mu_A(x)$ and $\mu_B(x)$ the associated standard set operations $A \cap B$ and $A \cup B$ are fuzzy sets that have respective membership functions

$$\begin{aligned} \mu_{A \cap B}(x) &= \min[\mu_A(x), \mu_B(x)] && \text{fuzzy intersection function} \\ \mu_{A \cup B}(x) &= \max[\mu_A(x), \mu_B(x)] && \text{fuzzy union function} \end{aligned} \tag{9}$$

2.3 Mamdani Fuzzy Inference System (MFIS)

Mamdani fuzzy inference systems are expert systems with approximate reasoning that map a vector of inputs (it can be a single input) to a single output (it can be a vector of outputs) [29]. In this examination, we rely on this type of a fuzzy inferential paradigm in order to characterize variability of an allometric response w as its descriptor a changes. Figure 1 displays the modules composing architecture of a MFIS and how they are related.

The fuzzification step identifies fuzzy subsets that pertain in the considered analysis. This usually accommodates linguistic terms allowing characterization of input data in terms of fuzzy sets. Then, customarily this is achieved by determining the proportion of similarity of data to each linguistic term. This also fixes the degree on which the input data belong to each of the considered fuzzy sets. This amounts to

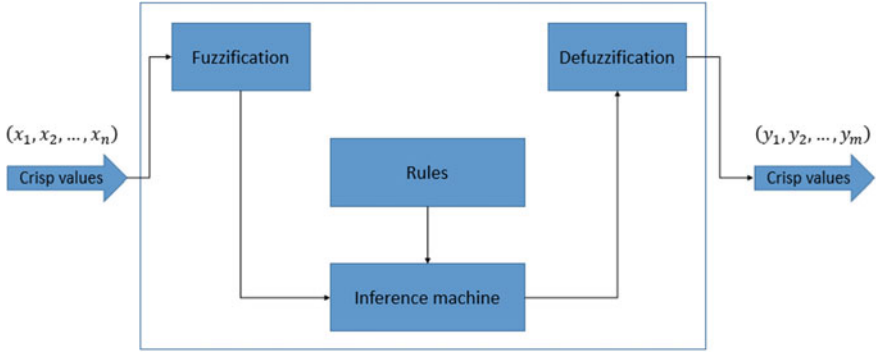


Fig. 1 Architecture of a mamdani fuzzy inference system, where n and m are the number of inputs and outputs of the system

characterizing the membership functions of the involved fuzzy sets (e.g. triangular defined by Eq. (3), Gaussian defined by Eq. (4) or *H-shaped* defined by Eq. (5)). The inference system takes on antecedents commonly conceived as conjunctions of one or more input fuzzy sets. Application of IF-THEN rules associate antecedents to a fuzzy output called consequent. For instance, a fuzzy rule, symbolized through ϕ^1 could involve an antecedent composed by a conjunction of two fuzzy sets A_1 and A_2 in a universe X that is associated to a fuzzy consequent B^1 in a universe Y , namely

$$\phi^1: \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ then } \tilde{y} \text{ is } B^1, \quad (10)$$

where x_1, x_2 are linguistic terms conceived as an input vector and \tilde{y} a fuzzy number. In the above rule fuzzy sets A_1 and A_2 and B^1 associate to membership functions $\mu_{A_1}(x_1)$, $\mu_{A_2}(x_2)$ and $\mu_{B^1}(y_1)$ one to one. In the general settings, we consider several input values x_1, \dots, x_m associated to linguistic terms A_1, \dots, A_m and a set Φ including n inference rules, that is $\Phi = \bigcup_i \phi^i$ with $1 \leq i \leq n$. The number N is further on referred as the cardinality of Φ . Usually, rule ϕ^i has a form

$$\phi^i: \text{If } x_1 \text{ is } A_1 \text{ and } x_2 \text{ is } A_2 \text{ and } \dots \text{ and } x_m \text{ is } A_m \text{ then } \tilde{y} \text{ is } B^1, \quad (11)$$

with $i = 1, 2, 3, \dots, n$.

The degree of certainty or activation of the antecedents in rule ϕ^i , is here represented by means of the symbol $\Omega_{\phi^i}(x_1, \dots, x_m)$. For instance, if the rules are represented in the canonical form given by Eq. (10) the minimum (fuzzy intersection in Eq. (9)) yields

$$\Omega_{\phi^i}(x_1, x_2) = \min\{\mu_{A_1}(x_1), \mu_{A_2}(x_2)\}. \quad (12)$$

In the general settings, the execution of rule ϕ^i will be done by applying a fuzzy intersection operator (fuzzy intersection function in Eq. (9)) whose arguments are

the value of the antecedents. The output of each rule will be the fuzzy number \tilde{y} resulting from the implication (π -area operator in Eq. (8)). The aggregation process is given by

$$\tilde{y} = |B^1|^{\Omega_{\phi^1}(x_1, \dots, x_m)} \oplus |B^2|^{\Omega_{\phi^2}(x_1, \dots, x_m)} \oplus \dots \oplus |B^n|^{\Omega_{\phi^n}(x_1, \dots, x_m)}, \quad (13)$$

where for $i = 1, 2, \dots, n$, $|B^i|^{\Omega_{\phi^i}(x_1, \dots, x_m)}$ is the $\Omega_{\phi^i}(x_1, \dots, x_m)$ -area of B^i defined by Eq. (8) and \oplus stands for the maximum operator (fuzzy union function in Eq. (9)). Although, there are other operators available, the maximum is the most used to perform this operation.

A defuzzification process converts the fuzzy values of the output variable \tilde{y} resulting from the inference process in precise information, this expressed by a crisp value y . Selection of defuzzification method can play a decisive role in the synthesis of fuzzy models for many application areas. This mainly depends on the expert. In this paper, we take into account the center of gravity defuzzification method for a fuzzy inference system involving a set of n rules. The center of gravity defuzzification method yields a crisp output y given by

$$y = \frac{\sum_{i=1}^n (\Omega_{\phi^i}(x_1, \dots, x_m) \mu_{B^i}(\Omega_{\phi^i}(x_1, \dots, x_m)))}{\sum_{i=1}^n \mu_{B^i}(\Omega_{\phi^i}(x_1, \dots, x_m))}. \quad (14)$$

2.4 Fuzzy Identification of the Allometric Model

In order to conceive the present fuzzy analysis, we initially utilized a k-means method as implemented on the function `kmeans` (Matlab R2018a). This device allows each pair of leaf biomass w and area a measurements to become an object with spatial differentiation. The procedure separates data into clusters C_k with $k \in \{1, 2, \dots, p\}$, each one composing of objects as close to each other as possible, and, as far as, possible from objects in the remaining mutually exclusive clusters. Each cluster C_k is defined by its member objects and by its center. The center of each cluster minimizes the sum of distances from all objects contained in it. The coordinates of the center (also called centroid) of the cluster C_k are through represented by the pair (\bar{w}_k, \bar{a}_k) . The number of clusters determines cardinality n that is number of IF-THEN rules creating Φ in the fuzzy inference system. Moreover, the allometric model of Eq. (1) allows the consideration of a function

$$\beta = f(\alpha), \quad (15)$$

where

$$f(\alpha) = \frac{w}{a^\alpha}. \quad (16)$$

Finally, entering a pair (w, a) in a cluster C_k into the fuzzy inference engine, returns a crisp estimate α . Then, Eq. (16) produces correspondent crisp value of β . Once this procedure exhausted all points (w, a) in C_k minimum and maximum values of calculated values of α and β estimates determine variation ranges for these parameters in C_k .

We evaluated agreement between observed and projected values, by examining values of Lin's Concordance Correlation Coefficient (CCC) [31]. This index is ordinarily designated by means of the symbol ρ . Agreement will be defined as poor whenever $\rho < 0.90$, moderate for $0.90 \leq \rho < 0.95$, good for $0.95 \leq \rho < 0.99$ or excellent for $\rho \geq 0.99$ [27].

The inference engine was implemented in Matlab 2018a software Version R2018, activated by using the function **evalfis**. Also, all statistical tasks were performed using the MATLAB.

3 Results

For raw data the k-means method, identified three clusters C_1 , C_2 and C_3 (Table 1; Fig. 2). This may indicate heterogeneity of the allometric response w as the covariate a changes. Echavarría-Heras et al. [27] compared two methods for the identification of the model of Eq. (1). One was linear regression of log-transformed data and a second one contemplated direct nonlinear regression in the original scale of data. They found the traditional approach producing biased results, but reported consistency of the nonlinear protocol. In any event addressed regression schemes could not detect heterogeneity. In order to explore its display we can count on the adaptation of a Mamdani fuzzy inference system for the present allometric set up.

In order to construct the present fuzzy allometric model, we depart from expert systems information. We offer a cluster distinction through linguistic terms adapted to cover the whole variation range of observed leaf weights and corresponding areas. This way, for fuzzification aims leaf area clusters in Fig. 2, from leftmost to rightmost are set to correspond to linguistic terms A_1 , A_2 and A_3 one to one. Additionally, we characterized them by means of Gaussian membership functions set by Eq. (4), that is (Fig. 3),

$$A_1 = [100, 276.9], \quad A_2 = [400, 1381] \quad \text{and} \quad A_3 = [580, 3370].$$

Table 1 Coordinates of centroids of clusters identified in eelgrass leaf weight to area pairs in raw data using the k-means method

Variable	Centroid of C_1	Centroid of C_2	Centroid of C_3
Leaf area (mm ²)	276.905	1380.748	3372.824
Leaf weight (g)	0.00473	0.0252	0.0696

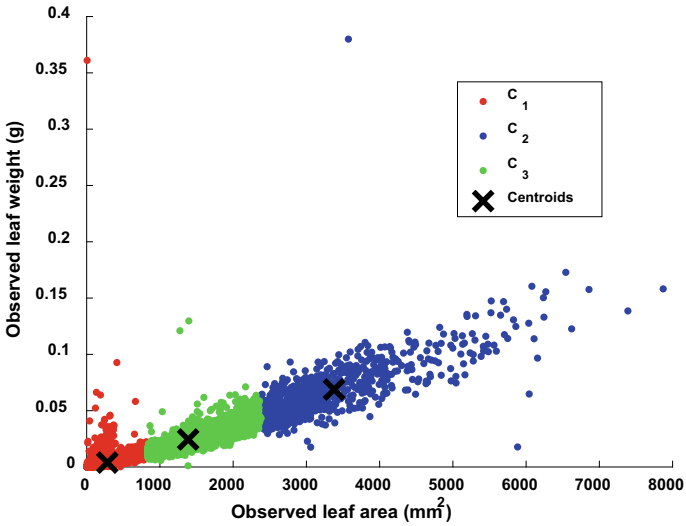


Fig. 2 Clusters identified by the k-means method for pairs of eelgrass leaf biomass and area composing raw data. An X shows the centroid of each cluster

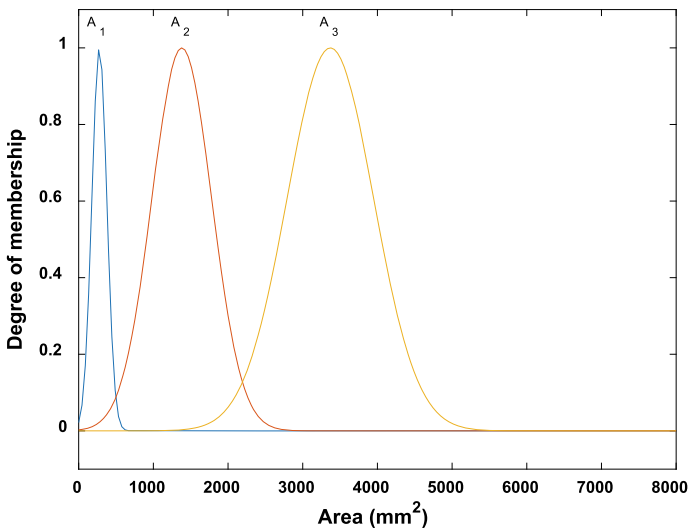


Fig. 3 Linguistic terms A₁, A₂ and A₃ for fuzzification of eelgrass leaf area \tilde{Area} in present data, which were characterized by Gaussian membership functions, A₁ = [100, 276.9], A₂ = [400, 381] and A₃ = [580, 3370]

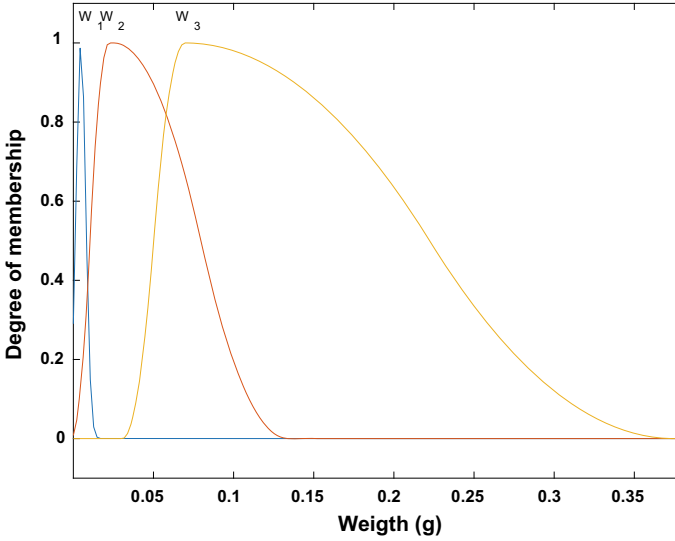


Fig. 4 Linguistic terms W_1 , W_2 and W_3 for fuzzification of eelgrass leaf weight $\tilde{\text{Weight}}$ in present data. We typified W_1 by means of a Gaussian membership function, $W_1 = [0.003, 0.00473]$, and W_2 and W_3 using *H-shaped* membership functions, $W_2 = [0.0015, 0.00223, 0.00223, 0.0134]$, $W_3 = [0.03072, 0.0696, 0.0696, 0.375]$

In the same way, for fuzzification of leaf weight clusters, the leaf weight linguistic variable is partitioned as W_1 , W_2 and W_3 . We characterized W_1 by means of a Gaussian membership function, $W_1 = [0.003, 0.00473]$, and used the *H-shaped* membership function to characterize W_2 and W_3 , that is (Fig. 4),

$$W_2 = [0.0015, 0.00223, 0.00223, 0.0134]$$

and

$$W_3 = [0.03072, 0.0696, 0.0696, 0.375].$$

Echavarría-Heras et al. [11] present estimates of the parameters α and β in the allometric model of Eq. (1). Reported confidence intervals allow a fuzzy characterization $\tilde{\alpha}$ of the parameter α through the contemplated fuzzy inference system of Mamdani type. Accordingly, fuzzy typifying depended on one linguistic term \mathbf{B} modeling a variation range of α as determined by the lower and upper limits of the associated confidence interval. A triangular membership function defined by Eq. (3) characterized \mathbf{B} as a fuzzy set (Fig. 5).

Building the Mamdani fuzzy method took into account a set of inference rules $R = \bigcup_i R^i$ with $i = 1, 2, 3, \dots, 9$ to form the associated inference system, namely

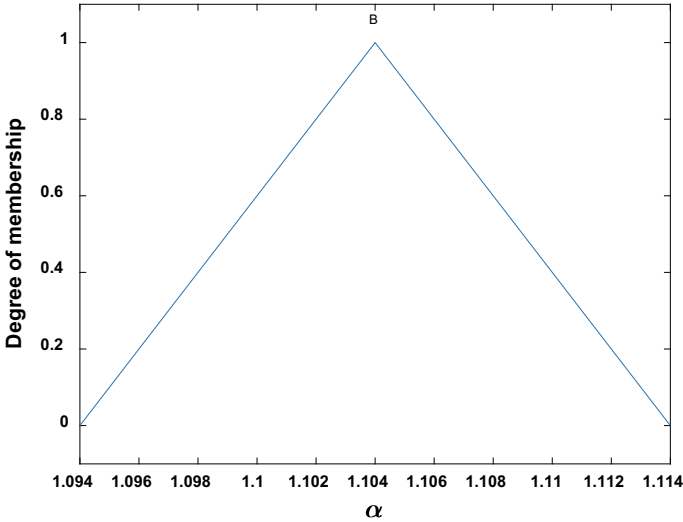


Fig. 5 Triangular membership function was used for fuzzification of the parameter α with only one linguistic term, $\tilde{B} = [1.0941.1041.114]$

- R^1 : IF (a_1 is A_1) AND (w_1 is W_1) THEN ($\tilde{\alpha}$ is B)
- R^2 : IF (a_1 is A_2) AND (w_1 is W_1) THEN ($\tilde{\alpha}$ is B)
- R^3 : IF (a_1 is A_3) AND (w_1 is W_1) THEN ($\tilde{\alpha}$ is B)
- R^4 : IF (a_1 is A_1) AND (w_1 is W_2) THEN ($\tilde{\alpha}$ is B)
- R^5 : IF (a_1 is A_2) AND (w_1 is W_2) THEN ($\tilde{\alpha}$ is B)
- R^6 : IF (a_1 is A_3) AND (w_1 is W_2) THEN ($\tilde{\alpha}$ is B)
- R^7 : IF (a_1 is A_1) AND (w_1 is W_3) THEN ($\tilde{\alpha}$ is B)
- R^8 : IF (a_1 is A_2) AND (w_1 is W_3) THEN ($\tilde{\alpha}$ is B)
- R^9 : IF (a_1 is A_3) AND (w_1 is W_3) THEN ($\tilde{\alpha}$ is B)

The minimum operator (fuzzy intersection function in Eq. (9)) evaluated the AND in the antecedents. Similarly, the THEN implication in the consequents resulted from application of the π -area, function (Eq. (8)). Aggregation of consequents depended on application of the maximum operator (fuzzy union function in Eq. (9)). And the defuzzification procedure carried away through the center of gravity method [32] applied to the support of the fuzzy set resulting from the aggregation step (Eq. (14)).

Entering a pair (w, a) in a cluster C_k into the inference engine, returned a calculated crisp value $\alpha(w, a)$. Then Eq. (15) produced a corresponding value $\beta(w, a)$ for the normalization constant β . Once this procedure exhausted all points (w, a) in C_k minimum and maximum values of $\alpha(w, a)$ and $\beta(w, a)$ determined variation ranges for the estimators of the parameters α and β in C_k [33]. Moreover, measured leaf areas and each pair of calculated parameters gave back allometric projections of observed leaf biomass values. Among pairs of calculated parameters we selected one $(\hat{\alpha}_k, \hat{\beta}_k)$

Table 2 Comparison of optimal concordance parameter estimates $\hat{\alpha}_k$ and $\hat{\beta}_k$ for cluster C_k obtained through the Mamdani Fuzzy Model applied on raw data

Cluster	$\hat{\beta}_k$	Variation range	$\hat{\alpha}_k$	Variation range
C_1	$1.166e-05$ $\Delta\beta = 0$	($8.266e-08$, 0.0499)	1.1040 $\Delta\alpha = 0$	(1.104 , 1.104)
C_2	$9.068e-06$ $\Delta\beta = 2.6e-6$	($5.944e-07$, $4.561e-05$)	1.1040 $\Delta\alpha = 0$	(1.104 , 1.104)
C_3	$9.068e-06$ $\Delta\beta = 2.6e-6$	($1.268e-06$, $4.553e-05$)	1.1040 $\Delta\alpha = 0$	(1.104 , 1.104)

We include variation intervals for calculated values of α and β . $\Delta\beta$ stands for the distance between a value of β and the greatest estimate of this parameter. Similar explanation for $\Delta\alpha$

yielding the maximum value of Lin's Concordance Correlation Coefficient (ρ) [31] between observed and projected values in cluster C_k (Table 2). This sets what we refer in what follow as an optimal concordance criterion to produce estimates $\hat{\alpha}_k$ and $\hat{\beta}_k$ for α and β in the cluster C_k . The average of optimal concordance estimates is denoted through $(\bar{\alpha}, \bar{\beta})$.

Table 2 show optimal concordance parameter estimates $\hat{\alpha}_k$ and $\hat{\beta}_k$ obtained for raw data through the optimal concordance criterion. Estimates of the parameter α are similar among clusters. But, we can ascertain differences for the parameter β for cluster 1. This on its own could imply heterogeneity in the allometric response. Moreover, concordance correlation coefficient for cluster one attained a low value of $\rho = 0.5307$, correspondingly, clusters 2 and 3 recorded $\rho = 0.7759$ and $\rho = 0.7103$ one to one. Besides, Fig. 6 displays matching allometric projections of individual leaf biomass values in clusters. We can ascertain a relatively abrupt transition between allometric mean responses of clusters one and two. In comparison transition between clusters two and three is fairly smooth. We can judge that clusters two and three exhibit allometric correspondence. Thus, results seem to indicate heterogeneity mainly determined by a differential allometric pattern in cluster one.

In turn lines in Fig. 7 suggest differences between an allometric pattern identified by the fuzzy inference system protocol, and that resulting from direct nonlinear regression. We also calculated the allometric-mean response function derived from parameter estimates $(\bar{\alpha}, \bar{\beta})$. These were obtained by averaging cluster level optimal concordance estimates of $(\hat{\alpha}_k, \hat{\beta}_k)$. This so obtained mean response associated to a measure of agreement between observed and projected values of $\rho = 0.9287$. Meanwhile, the mean response curve yield by nonlinear regression corresponded to $\rho = 0.9307$, thus, differing only slightly from its fuzzy counterpart. But, we can learn that fuzzy projection overestimates the nonlinear regression one. Interestingly, shown in between curve gotten by averaging fuzzy derived and nonlinear regression projections associates to a higher reproducibility strength $\rho = 0.9323$. Thus, we can hypothesize that reproducibility strength derived from the fuzzy model, could get higher by accommodating suitable forms of membership functions. Contrasting

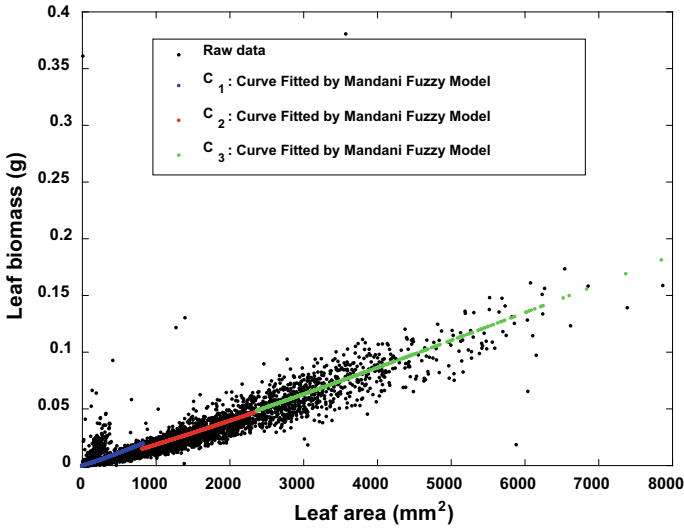


Fig. 6 Comparison of allometric projections of individual leaf biomass values in identified clusters. We can ascertain a relatively abrupt transition between allometric mean responses tied to clusters one and two. In comparison corresponding transition between clusters two and three is relatively smooth

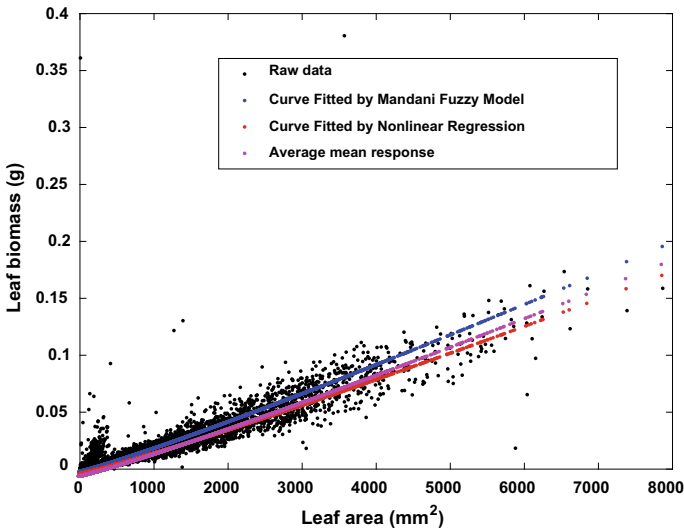


Fig. 7 Allometric mean response functions obtained from parameters α and β gotten by averaging fuzzy-optimal concordance estimates for clusters in raw data (blue lines, $\rho = 0.9287$) compared with those produced by direct nonlinear regression (red line, $\rho = 0.9307$). Averaged mean response between the fuzzy model and nonlinear regression ones (pink lines). This resulted from parameters α and β acquired by averaging parameter estimates produced by the fuzzy and nonlinear regression methods. This arrangement associated to a largest reproducibility ($\rho = 0.9323$).

reproducibility strength resulting from nonlinear regression of the model of Eq. (1) keeps fixed.

Detected heterogeneity in allometry of individual leaf biomass values points toward differences in concomitant projections of mean leaf biomass at cluster level. Assessing such a discrepancy circumscribes to consideration of cluster one and a composite of clusters two and three, since parametric differences occurred only for cluster one. Figure 8 panel (a) compares observed mean biomass in leaves composing cluster 1 with corresponding allometric projections produced by the fuzzy method ($\rho = 0.3539$). Panel (c) pertains to similar comparison involving nonlinear regression ($\rho = 0.4172$). In turn panel (b) presents comparison of observed and fuzzy–method projected mean biomass for leaves in clusters 2 and 3 combined ($\rho = 0.9794$). Panel (d) pertains to conforming results based on nonlinear regression ($\rho = 0.9307$). We can learn that reproducibility in cluster 1 is poor irrespective

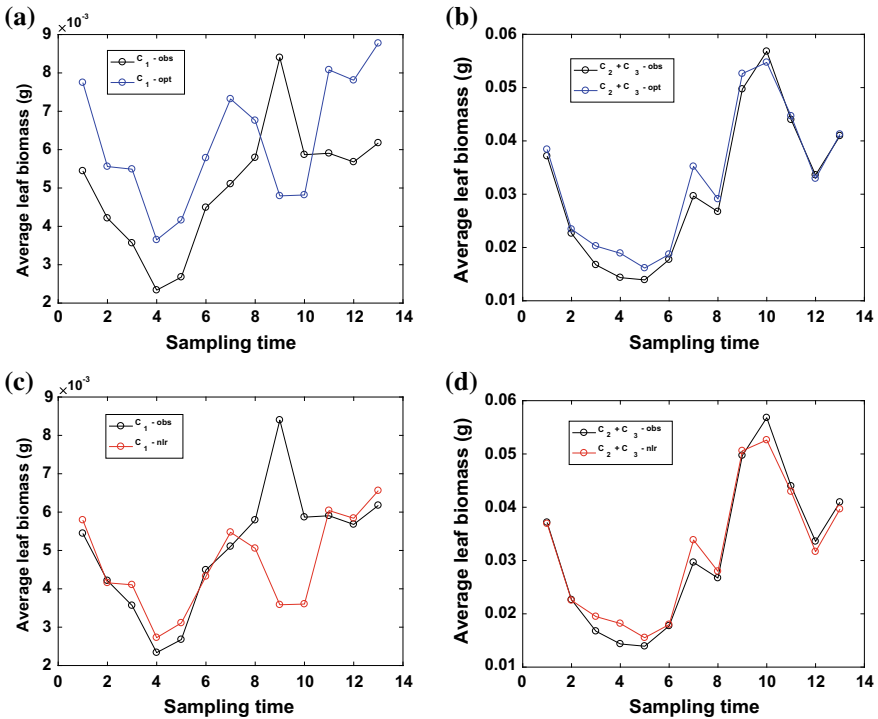


Fig. 8 Average leaf biomass projected for raw data. Panel (a) shows observed mean biomass in leaves composing cluster 1 and corresponding allometric projections produced by the fuzzy method. Panel (c) pertains to parallel comparison involving nonlinear regression. Panel (b) presents observed and fuzzy–method projected mean biomass for leaves in clusters 2 and 3 combined. Panel (d) displays related results based on nonlinear regression. In legends opt means mean response produced by using optimal concordance parameter estimates ($\hat{\beta}_k, \hat{\alpha}_k$) and nlr stands for nonlinear regression

Table 3 Coordinates of centroids of cluster identified in eelgrass leaf weight to area pairs in processed data using the k-means method

Variable	Centroid of C ₁	Centroid of C ₂	Centroid of C ₃
Leaf area (mm ²)	295.699	1497.024	3456.218
Leaf weight (g)	0.0044	0.0274	0.07136

of projection method. Meanwhile, reproducibility strength of allometric proxies for leaves in clusters 2 and 3 derived from both methods remarkably improved, being the fuzzy approach the one resulting in a higher suitability. Cluster one comprises smaller leaves in data which normally associate to extremely reduced biomasses. This hints on errors while estimating corresponding dry weights at laboratory. Thus, a data quality issue could rise relevant in explaining heterogeneity for present raw data [34].

In order to explore data quality driven heterogeneity, we compare performances of fuzzy and nonlinear regression projections acquired from processed data. Again, the k-means method, identified three clusters C₁, C₂ and C₃ (Table 3; Fig. 9). As it occurred for raw data, spread in Fig. 9 may suggests heterogeneity.

Figures 10 and 11 display membership functions for fuzzification of leaf biomass and area input variables in processed data. Correspondingly, Fig. 12 shows the membership function for fuzzification of the allometric exponent α . Table 4 show parameter estimates $\hat{\alpha}_k$ and $\hat{\beta}_k$ obtained by means of the fuzzy method for processed data

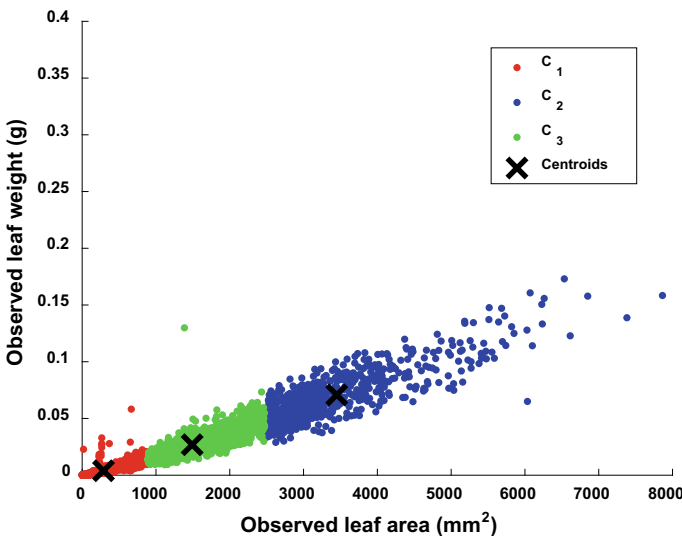


Fig. 9 Clusters identified by the k-means method for pairs of eelgrass leaf biomass and area values composing the processed data set. An X shows the centroid of each cluster

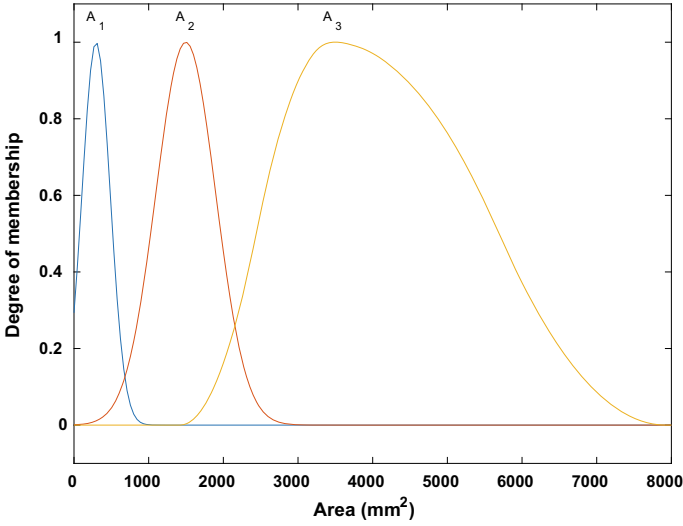


Fig. 10 Linguistic terms A_1 , A_2 and A_3 for fuzzification of eelgrass leaf area \bar{Area} in present data. A_1 and A_2 were characterized by Gaussian membership functions, $A_1 = [189, 296]$ and $A_2 = [400, 1497]$ respectively. And were characterized by a *H-shaped* membership function $A_3 = [1420, 3460, 3460, 7926]$

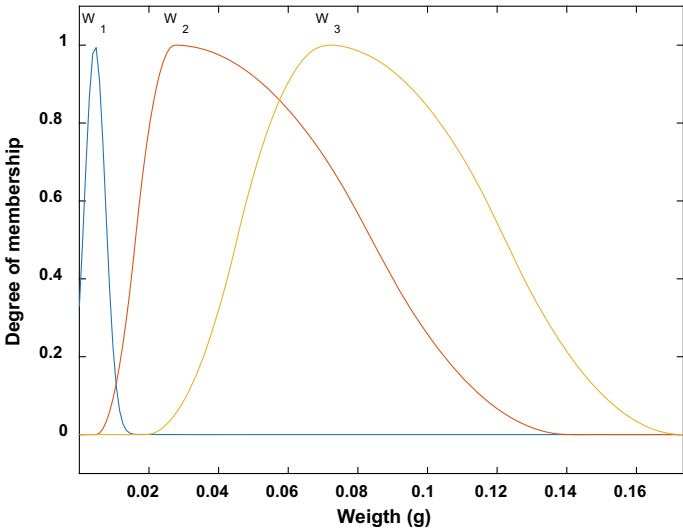


Fig. 11 Typified linguistic terms W_1 , W_2 and W_3 for fuzzification of eelgrass leaf weight \bar{Weigth} in processed data. We characterized W_1 through a Gaussian membership function: $W_1 = [0.003, 0.00449]$. Meanwhile, a *H-shaped* membership function form characterized W_2 and W_3 , namely $W_2 = [0.004830.0274, 0.0274, 0.1407]$ and $W_3 = [0.0191, 0.0714, 0.0714, 0.1733]$

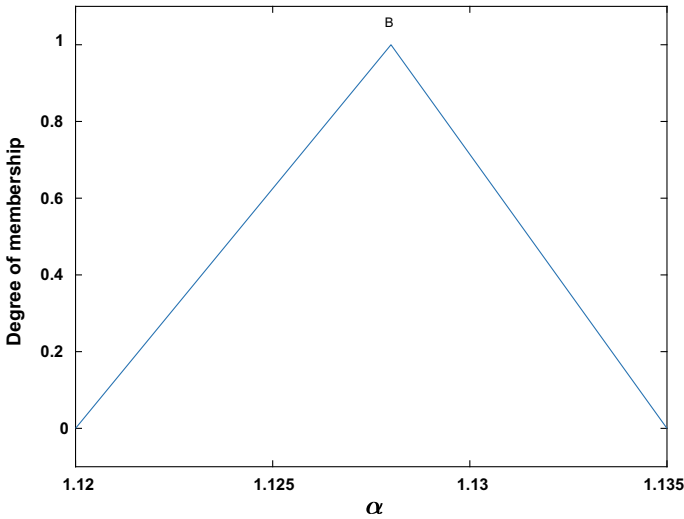


Fig. 12 Fuzzification of the parameter α required only one triangular linguistic term, $\tilde{\mathbf{B}} = [1.121, 1.128, 1.135]$

Table 4 Comparison of optimal concordance parameter estimates $\hat{\alpha}_k$ and $\hat{\beta}_k$ for cluster C_k obtained through the Mamdani Fuzzy Model applied on processed dat

Cluster	$\hat{\beta}_k$	Variation range	$\hat{\alpha}_k$	Variation range
C_1	$8.488e-06$ $\Delta\beta = 0$	$(7.461e-08, 0.0008)$	1.1276 $\Delta\alpha = 0$	$(1.12753, 1.12766)$
C_2	$7.142e-06$ $\Delta\beta = 1.346e-6$	$(3.477e-06, 3.723e-05)$	1.1276 $\Delta\alpha = 0$	$(1.12753, 1.12766)$
C_3	$287e-06$ $\Delta\beta = 1.201e-6$	$(3.568e-06, 1.094e-05)$	1.1276 $\Delta\alpha = 0$	$(1.12750, 1.12766)$

We include variation intervals for calculated values of α and β . $\Delta\beta$ stands for the distance between a value of β and the greatest estimate of this parameter. Similar explanation for $\Delta\alpha$

and through the optimal correspondence criterion. Estimates of the parameter α are similar among clusters. Nevertheless, we can be acquainted that differences for the parameter β remain among clusters. These deviations imply heterogeneity in the allometric mean response function among clusters. Moreover, compared to raw data reproducibility strength of the fuzzy method for individual leaf biomass in cluster one improved to $\rho = 0.92399$. Correspondingly, clusters 2 and 3 recorded $\rho = 0.8117$ and $\rho = 0.8274$ one to one. Besides, Fig. 13 still displays a relatively abrupt transition between allometric mean responses of clusters one and two. In comparison corresponding transition between clusters two and three is smooth.

Correspondingly, lines in Fig. 14 show similarity in mean response curves acquired from the fuzzy-allometric (blue lines) and nonlinear regression (red lines)

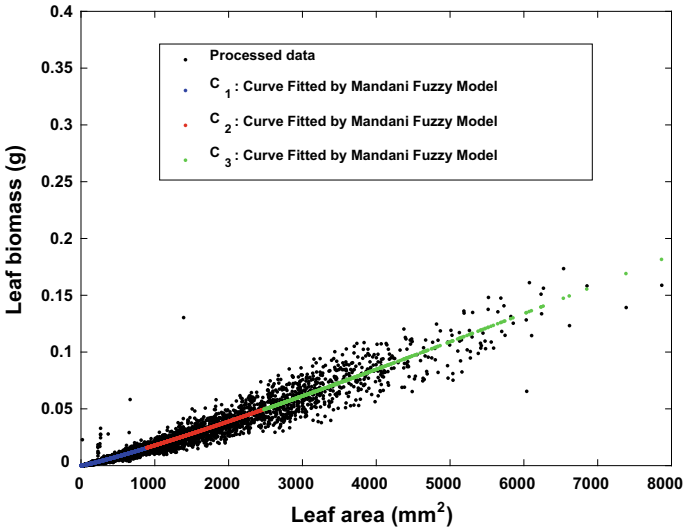


Fig. 13 Comparison of allometric projections of individual leaf biomass values in clusters identified for processed data. We can ascertain a relatively abrupt transition between allometric mean responses of clusters one and two. In comparison corresponding transition between clusters two and three is smooth. In legends opt means mean response produced by using optimal concordance parameter estimates $(\hat{\beta}_k, \hat{\alpha}_k)$

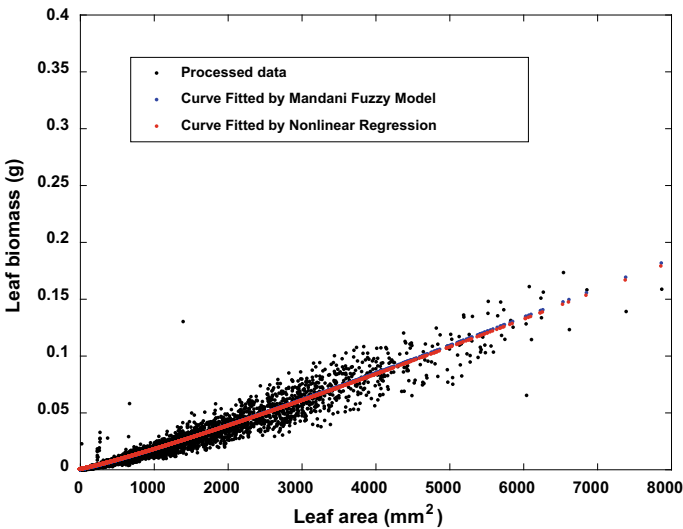


Fig. 14 Allometric mean response corresponding to estimates $\bar{\beta}$ and $\bar{\alpha}$, gotten by averaging fuzzy-optimal correspondence estimates for clusters, $\hat{\alpha}_k$ and $\hat{\beta}_k$, in processed data (blue lines, $\rho = 0.9738$) compared with those produced by direct nonlinear regression (red line, $\rho = 0.9736$). Allometric mean response lines coincide

protocols. The fuzzy-allometric projections associate to a value of $\rho = 0.9738$ suggesting remarkable agreement between projected and observed values. Meanwhile, nonlinear regression corresponded to $\rho = 0.9736$. Comparing to raw data removing inconsistent replicates markedly improved reproducibility of both fuzzy and nonlinear regression protocols. We can learn that for this try fuzzy projections coincide with those tied to nonlinear regression. However, compared with nonlinear regression displaying high reproducibility even for raw data enhanced reproducibility the fuzzy method seems to be essentially dependent on data quality.

We now turn to examine differences in projections of mean leaf biomass at cluster level due to heterogeneity in processed data. Results in Table 4 show that differences between values of β estimated for clusters 2 and 3 amount to $0.14e-6$. Meanwhile, differences between these estimates and that corresponding to cluster one averaged $1.27e-6$, about 10 times the firstly quoted deviation. Then as it occurred for raw data assessing divergences in projections of average leaf biomass circumscribes to consideration of cluster one and a composite of clusters two and three.

Figure 15 panel (a) compares observed mean biomass in leaves comprising cluster 1 in processed data with corresponding allometric projections produced by the fuzzy method ($\rho = 0.9653$). Panel (c) pertains to similar comparison involving nonlinear regression ($\rho = 0.9708$). In turn panel (b) presents comparison of observed mean biomass for leaves in clusters 2 and 3 combined and corresponding fuzzy-method projected values ($\rho = 0.9876$). Panel (d) pertains to compatible results based on nonlinear regression ($\rho = 0.9879$). We can learn that reproducibility in all clusters is excellent irrespective of projection method. Then, removal of inconsistent replicates in cluster one remarkably improved consistency of the fuzzy projection method. Thus, data quality is indeed a factor raising consistency of the fuzzy approach. But in any event, even taking data quality control into account, heterogeneity persisted. This hints on this driven by intrinsic factors in leaf growth rather than being generated by inconsistent replicates in raw data.

4 Discussion

Eelgrass is an important seagrass species distributed in estuaries and nearshore environments worldwide, providing habitat and foraging grounds for marine animals, buffering the shoreline from erosion, filter the water, and oxygenate the sediments [9]. Anthropogenic stress nowadays threatens eelgrass permanence so that aforementioned ecological services could be lost [27]. For this reason, eelgrass has been the subject of intense research and conservation efforts. The measurement of pertinent mean leaf biomass in shoots makes available key information for the evaluation of its overall status. Traditional assessments rely on tedious and time consuming dry weight estimations at laboratory. Besides, these methods are destructive, this particularly affecting developing conservation plots. Allometric projection offers a non-destructive alternative. Nevertheless, allometric proxies of average leaf biomass in shoots derive from aggregations of allometric projections of individual leaf biomass.

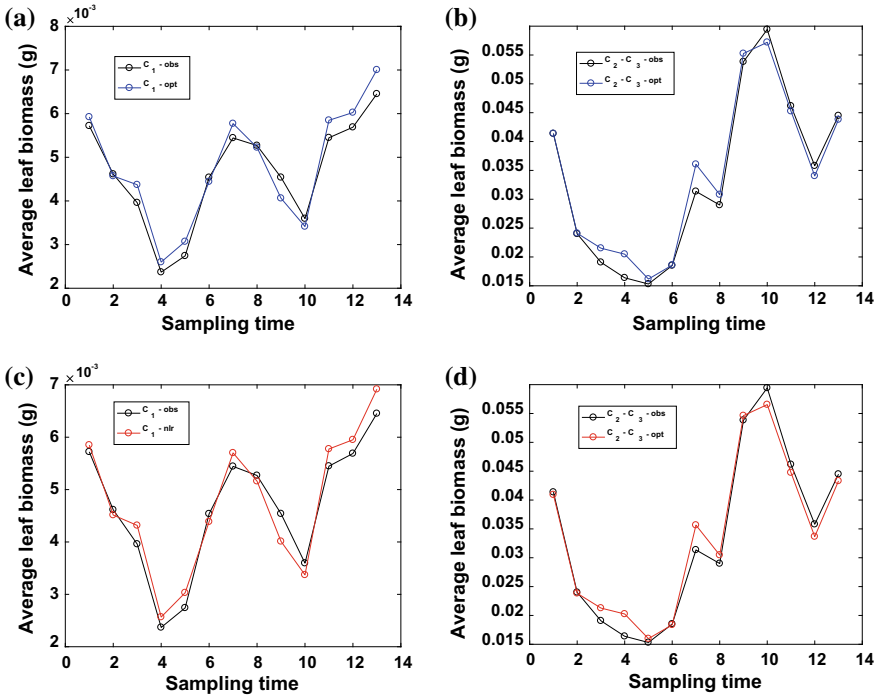


Fig. 15 Average leaf biomass projected for processed data. Panel (a) shows observed mean biomass in leaves composing cluster 1 and corresponding allometric projections produced by the fuzzy method. Panel (c) pertains to parallel comparison involving nonlinear regression. Panel (b) presents observed and fuzzy-method projected mean biomass for leaves in clusters 2 and 3 combined. Panel (d) displays related results based on nonlinear regression. In legends opt means mean response produced by using optimal concordance parameter estimates ($\hat{\beta}_k, \hat{\alpha}_k$) and nlr stands for nonlinear regression

Hence, there are concerns of propagation of error in parameter estimates affecting adequacy of derived nondestructive constructs [35]. Since, they derive from allometric projections of individual leaf biomass factors like data quality, sample size, and analysis method could be relevant [15–17].

Concerning effects of analysis method on allometric examination Packard and Birchard [17] and Packard and Boardman [21] concluded that a fit of Eq. (1) via direct nonlinear regression provides better estimates of α and β than the traditional scheme of linear regressions operating on log-transformed data. Thus, suitability of analysis method for the identification of the allometric model of Eq. (1), being this log-linear regression or direct non-linear regression, turns out to be an important research problem. For present eelgrass data, Echavarría-Heras et al. [27] concluded that the traditional method produced biased projections of average leaf biomass in shoots, and that only a protocol based on the model of Eq. (1) fitted directly in arithmetical space by nonlinear regression methods could grant consistent predictions of

observed leaf biomass values. Therefore, choosing a pertinent analysis method for allometric examination of present data seems to be a controversial matter. Moreover, selection of the suitable approach mainly depend on appropriate understanding of the scale on which random errors are distributed. Usually, examination of the allometric dependence of a response w in terms of a covariate a departs from a regression model $w = h(a)e^\epsilon$ in arithmetical space, being $h(a)$ a given power function of a and with e^ϵ identified as a multiplicative error factor characterized by a random variable ϵ . This is customarily assumed to be normally distributed with zero mean and homogeneous variance. Thus such a regression arrangement sets the function $h(a)$ as the related allometric mean response function. The correspondingly regression model in geometrical space takes on the form $\log w = \log h(a) + \epsilon$, that sets an additive error term ϵ . Alternatively, we can consider a nonlinear regression model with additive random error term ϵ in direct arithmetical scale, that is, $w = h(a) + \epsilon$, as necessary. But, this approach can also lead to inconvenience [36]. Moreover, suitability of analysis approach depends on imbedded error distribution. Indeed, practice elucidates that fitting the allometric model (1) with an additive error is not equivalent to fitting a linearized model after a log–transformation. Thus, choosing among these regression protocols must follow explicit examination of underlying error structure in data [37]. In the other hand the present approach based on a fuzzy inference system proved to be useful even though, we dealt with an indeterminate error structure. Moreover, the concordance correlation coefficient for allometric proxies of mean leaf biomass acquired from raw data and the Mamdani fuzzy model recorded $\rho = 0.9840$. This entails excellent reproducibility strength. Moreover, Echavarria-Heras et al. [27] considered a Takagui-Sugeno-Kendal fuzzy protocol with linear consequents in arithmetical space, aimed to model the intrinsic nonlinear functional relationship $w = f(a)$ between eelgrass leaf biomass w and area a in arithmetical space. It turns out that this protocol also entailed excellent reproducibility. But, the parameter β is assumed to be strongly influenced by environmental conditions thereby remaining non-static through the analysis [6]. Correspondingly, the allometric exponent α could be allowed to change either continuously, or abruptly as the covariate grows [18]. This heterogeneity bearing condition is referred as complex allometry [38]. Besides, the general output of the Takagi-Sugeno fuzzy model in [27] involves two linear consequents weighted at each value of the covariate by corresponding normalized firing strength factors. This structure provides an efficient characterization of a nonlinear function $f(a)$ that describes variation of w in terms of a . But, this scheme does not allow to typify heterogeneity of w as established by scaling parameters taking different values over corresponding growth stanzas. Beyond, the named Takagi-Sugeno approach the present Mamdani fuzzy model offers estimates $(\hat{\alpha}_k, \hat{\beta}_k)$ of the vector (α, β) in cluster C_k . This delivers a direct interpretation of allometric heterogeneity. Moreover, the presently offered fuzzy model permitted to untie a differential allometric pattern for leaves on a first cluster that associates to the smallest leaves. This could have not been achieved by relying only in direct nonlinear regression working over the whole range of the covariate. Certainly, a linear regression method and the

Mamdani fuzzy model entailed similar reproducibility strengths. But, the interpretative power of the later is far beyond that of the conventional one. Indeed, the present fuzzy method being based in a thorough partitioning of data brings heterogeneity into the analysis through a direct and simplified approach.

5 Conclusion

The present results show that identification of heterogeneity of the allometric response as its covariate changes, may possibly enhance accuracy of derived proxies, but could also extend our understanding of the true association between these variables. Regular practice in allometric examination bears direct nonlinear regression based on the model of Eq. (1) as a required protocol to avoid bias concomitant to a linear regression approach in geometrical space. But, regardless granting a relative gain in predictive power, complexity of the former approach will undoubtedly fail to detect heterogeneity inherent in the allometric relationship. An improvement of the predictive strength of the referred nonlinear paradigm seems feasible upon consideration of a suitable error structure. This eventually will bring a representation of the distribution error term as a mixture thereby complicating identification matters. On the contrary the intuitive-flexible and efficient arrangement conforming the presently offered Mamdani fuzzy model proved to be a convenient tool to characterize heterogeneity and enhance efficiency of considered allometric proxies. In summary, when the allometric exponent α is a continuously changing function of the allometric covariate or when it changes abruptly, the offered fuzzy logic approach could identify fit accurate concomitant values for the parameter β . This allow to obtain estimates of the allometric parameters valid for each cluster conforming heterogeneity of the response. Moreover, this last approach will allow a sound identification of parameter variability whatever error structure is imbedded in the data.

References

1. Newman, M.E.J.: Power laws, pareto distributions and Zipf's law. *Contemp. Phys.* **46**, 323–351 (2005)
2. Marquet, P.A., Quiñones, R.A., Abades, S., Labra, F., Tognelli, M.: Scaling and power-laws in ecological systems. *J. Exp. Biol.* **208**, 1749–1769 (2005)
3. West, G.B., Brown, J.H.: The origin of allometric scaling laws in biology from genomes to ecosystems: towards a quantitative unifying theory of biological structure and organization. *J. Exp. Biol.* **208**, 1575–1592 (2005)
4. Maritan, A., Rigon, R., Banavar, J.R., Rinaldo, A.: Network allometry. *Geophys. Res. Lett.* **29**(11), 1–4 (2002)
5. Filgueira, R., Labarta, U., Fernández-Reiriz, M.J.: Effect of condition index on allometric relationships of clearance rate in *mytilus galloprovincialis lamarck*, 1819. *Rev. Biol. Mar. Oceanogr.* **43**(2), 391–398 (2008)

6. Kaitaniemi, P.: How to derive biological information from the value of the normalization constant in allometric equations. *PLoS One* **3**(4), e1932 (2008)
7. Martin, R.D., Genoud, M., Hemelrijk, C.K.: Problems of allometric scaling analysis: examples from mammalian reproductive Biology. *J. Exp. Biol.* **208**, 1731–1747 (2005)
8. De Robertis, A., Williams, K.: Weight-length relationships in fisheries studies: the standard allometric model should be applied with caution. *Trans. Am. Fish. Soc.* **137**(3), 707–719 (2008)
9. Echavarría-Heras, H.A., Leal-Ramírez, C., Villa-Diharce, E., Cazarez-Castro, N.R.: The effect of parameter variability in the allometric projection of leaf growth rates for eelgrass (*Zostera marina* L.) II: the importance of data quality control procedures in bias reduction. *Theor. Biol. Med. Model.* **12**(30) (2015)
10. García-Soria, D., Abanto-Rodríguez, A., Del Castillo, D.: Determinación de ecuaciones alométricas para la estimación de biomasa aérea de *Guadua sacocarpa* Lodoño & Peterson de la comunidad nativa bufeo pozo, Ucayali, Perú. *Folia Amazonica* **24**(2) 139–144 (2015)
11. Echavarría Heras, H.A., Leal Ramírez, C., Villa Diharce, E., Cazarez Castro, N.R.: On the suitability of an allometric proxy for nondestructive estimation of average leaf dry weight in eelgrass shoots I: sensitivity analysis and examination of the influences of data quality, analysis method, and sample size on precision. *Theoret. Biol. Med. Model.* **15**(4), 20 (2018)
12. Solana-Arellano, M.E., Echavarría-Heras, H.A., Leal-Ramírez, C., Lee, K.S.: The effect of parameter variability in the allometric projection of leaf growth rates for eelgrass (*Zostera marina* L.). *Lat. Am. J. Aquat. Res.* **42**(5), 1099–108 (2014)
13. Echavarría-Heras, H.A., Lee, K.S., Solana-Arellano, M.E., Franco-Vizcaino, E.: Formal analysis and evaluation of allometric methods for estimating above-ground biomass of eelgrass. *Ann. Appl. Biol.* **159**(3), 503–515 (2011)
14. Echavarría-Heras, H.A., Solana-Arellano, M.E., Franco-Vizcaino, E.: An allometric method for the projection of eelgrass leaf biomass production rates. *Math. Biosci.* **223**(1), 58–65 (2010)
15. Savage, V.M., Gillooly, J.F., Woodruff, W.H., West, G.B., Allen, A.P.: The predominance of quarter-power scaling in biology. *Funct. Ecol.* **18**, 257–282 (2004)
16. Hui, D., Jackson, R.B.: Uncertainty in allometric exponent estimation: a case study in scaling metabolic rate with body mass. *J. Theor. Biol.* **249**, 168–177 (2007)
17. Packard, G.C., Birchard, G.F.: Traditional allometric analysis fails to provide a valid predictive model for mammalian metabolic rates. *J. Exp. Biol.* **211**, 3581–3587 (2008)
18. Packard, G.C.: Is non-loglinear allometry a statistical artifact? *Biol. J. Lin. Soc.* **107**(4), 764–773 (2012)
19. Hartnoll, R.G.: The determination of relative growth in Crustacea. *Crustaceana* **34**(3), 282–293 (1978)
20. Barradas, J.R.S., Lermen, I.S., Larre, G.G., Martins, T.P., Fontura, N.F.: Polyphasic growth in fish: a case study with *Corydoras paleatus* (Siluriformes, Callichthyidae). *Ser. Zool, Iheringia* (2016)
21. Packard, G.C., Boardman, T.J.: Model selection and logarithmic transformation in allometric analysis. *Physiol. Biochem. Zool.* **81**, 496–507 (2008)
22. Zadeh, L.A.: Fuzzy sets. *Inf. Control.* **8**(3), 338–353 (1965)
23. Zimmerman, H.J.: *Fuzzy Set Theory and Its Applications*, 2nd edn. Kluwer, Boston MA (1991)
24. Takagi, T., Sugeno, M.: Fuzzy identifications of systems and its applications to modeling and control. *IEE Trans Syst. MAN Cybern.* **15**(1), 116–132 (1985)
25. Sugeno, M., Kang, G.T.: Structure identification of fuzzy model. *Fuzzy Sets Syst.* **28**, 15–33 (1988)
26. Bezdek, J.C., Pal, S.K.: *Fuzzy Models for Pattern Recognition*. IEEE Press, New York (1992)
27. Echavarría-Heras, H.A., Leal-Ramírez, C., Castro-Rodríguez, J.R., Villa-Diharce, E., Castillo, O.: A Takagi-Sugeno-Kang fuzzy model formalization of eelgrass leaf biomass allometry with application to the estimation of average biomass of leaves in shoots: comparing the reproducibility strength of the present fuzzy and related crisp proxies. In: Castillo, O., Melin, P., Kacprzyk, P. (eds.) *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, 2nd edn. Springer, Berlin (2018)

28. Zadeh, L.A.: Fuzzy logic = computing with words. *IEEE Trans. Fuzzy. Syst.* **4**(2), 103–111 (1996)
29. Pedrycz, W., Gomide, F.: *An Introduction to Fuzzy Sets: Analysis and Design*. The MIT Press, Massachusetts (1998)
30. Barros, L.C., Bassanezi, R.C.: *Tópicos em lógica fuzzy e biomatemática*, 2nd edn., p. 344. UNICAMP/IMECC, Campinas (2010)
31. Lin, L.I.K.: A concordance correlation coefficient to evaluate reproducibility. *Biometrics* **45**, 255–268 (1989)
32. Wang, L.X., Mendel, J.M.: Fuzzy basis functions, universal approximation, and orthogonal least-squares learning. *IEEE Trans. Neural Networks.* **3**(5), 807–814 (1992)
33. Bitar, S.D., Campos, C.P., Freitas, C.E.C.: Applying fuzzy logic to estimate the parameters of the length-weight relationship. *Braz. J. Biol.* 1–8 (2016)
34. Mascaro, J., Litton, C.M., Hughes, R.F., Uowolo, A., Schnitzer, S.A.: Is logarithmic transformation necessary in allometry? Ten, one-hundred, one-thousand-times yes. *Biol. J. Linn. Soc.* **111**, 230–233 (2014)
35. Echavarría-Heras, H.A., Solana-Arellano, M.E., Leal-Ramírez, C., Franco-Vizcaino, E.: An allometric method for measuring leaf growth in eelgrass, *Zostera marina*, using leaf length data. *Bot. Mar.* **56**(3), 275–86 (2013)
36. Lai, J., Yang, B., Lin, D., Kerkhoff, A.J, Ma., K.: The allometry of coarse root biomass: log-transformed linear regression or nonlinear regression? *PLoS ONE* (2013)
37. Xiao, X., White, E.P., Hooten, M.B., Durham, S.L.: On the use of log-transformation vs. non-linear regression for analyzing biological power laws. *Ecology* **92**(10), 1887–1894 (2011)
38. Jolicoeur, P.: A simplified model for bivariate complex allometry. *J. Theor. Biol.* **140**, 41–49 (1989)

Fireworks Algorithm (FWA) with Adaptation of Parameters Using Interval Type-2 Fuzzy Logic System



Juan Barraza, Fevrier Valdez, Patricia Melin and Claudia I. González

Abstract The main goal of this paper is to improve the performance of the Fuzzy Fireworks Algorithm (FFWA), which is a variation of conventional Fireworks Algorithm (FWA). In previous work the FFWA was proposed on Type-1 Fuzzy Logic to adjust parameters dynamically and the difference now in this work is that we use the Interval Type-2 Fuzzy Logic for and adjust the parameter of the explosion amplitude of each firework, and this variation, we called as Interval Type-2 Fuzzy Fireworks Algorithm and we denoted as IT2FFWA. To evaluate the performance of FFWA and IT2FFWA we tested both algorithms with 12 mathematical Benchmark functions.

Keywords Fireworks algorithm · Fuzzy parameter adaptation · Type-2 fuzzy logic

1 Introduction

Soft computing has the aim of achieving intelligence for machines; also the concept of artificial intelligence is used to when a machine is capable of showing that can imitate some functions of humans. In computational science the functions more important to imitate for machines are the learning and solve problems, for consequent, there are methodologies as neural networks, fuzzy logic or algorithms based on nature, swarm intelligence, or the physical, which are used on the artificial intelligence to imitate the functions of humans mentioned above [1].

This paper concentrates on the methodologies of the algorithms based on swarm intelligence [2–4], because de conventional FWA is an algorithm that simulates the explosion of the fireworks and are based on swarm intelligence, in this case, the swarm is imitated with the sparks that are generated around of the each firework [5–7].

J. Barraza · F. Valdez (✉) · P. Melin · C. I. González
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: fevrier@tectijuana.mx

J. Barraza
e-mail: Jbarrazagerardo@gmail.com

P. Melin
e-mail: pmelin@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_3

On the other hand we can say that fuzzy logic is a methodology developed by L. Zadeh that allows obtaining a conclusion or conclusions based on the input information that could be ambiguous, imprecise, with noise or incomplete, actually, in the area of computational, is called as Type-1 Fuzzy Logic [8, 9].

Interval Type-2 Fuzzy logic is a variation of Type-1 fuzzy logic, with the difference that in Type-2 the degree of uncertainty is higher than in Type-1, which means that, in Type-1 the degree of uncertainty is of two dimensions and the Type-2 is represented in three dimensions where the third dimension is the value of the membership function at each point on its two-dimensional domain that is called its footprint of uncertainty (FOU) [10, 11].

This paper is organized as follows: Sect. 2 shows the Conventional Fireworks algorithm (FWA), the next Section (Sect. 3) shows a previous work of FWA (FFWA). Our proposed method is in Sects. 4 and 5 describes the Benchmark Functions; Sect. 6 shows the experiments and results, and Sect. 7 the conclusions.

2 Fireworks Algorithm (FWA)

The conventional Fireworks Algorithm (FWA) is a swarm intelligence algorithm as mentioned above, and it is composed for 4 general steps: initialization of locations, calculation of the number of sparks, calculation of the explosion amplitude for each firework and selection of the best location [12].

In this work we only explained the calculation of explosion amplitude because is the parameter that we modify, and for more explanation of this algorithm (FWA) the reader is refer to the previous papers [13].

Explosion Amplitude

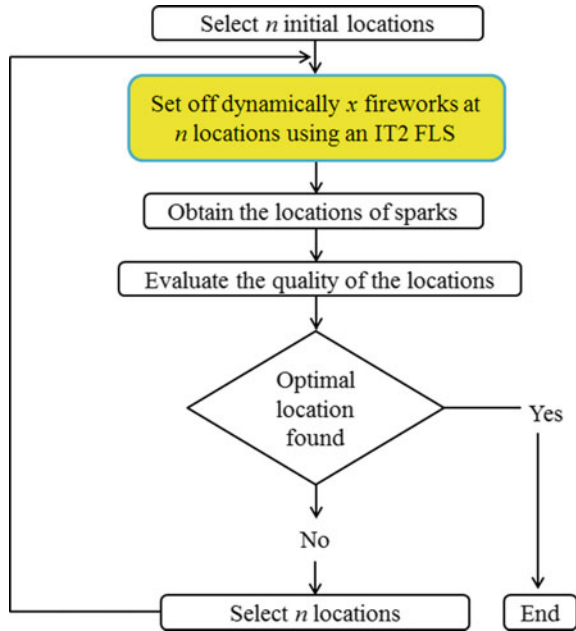
The explosion amplitude for each firework is calculated as follows:

$$A_i = \hat{A} \cdot \frac{f(x_i) - y_{\min} + \epsilon}{\sum_{i=1}^n (f(x_i) - y_{\min}) + \epsilon} \quad (1)$$

where \hat{A} is a constant parameter that controls the maximum amplitude of each firework, $y_{\min} = \min(f(x_i)) (i = 1, 2, 3, \dots, n)$, indicate the minimum value (best) of the objective function among n fireworks and ϵ indicates the smallest constant in the computer, and it is utilized with the goal that an error of division by zero does not occur [14].

Into the FWA, there are three algorithms, the algorithm for calculation of the number of sparks for each firework (algorithm 1), the algorithm to generate Gaussian sparks (algorithm 2) and the general algorithm (algorithm 3) that shows the pseudo code of the conventional fireworks algorithm [15]. The performance of algorithms 1 and 2 we can be found in paper [13] or [16], and both algorithms are used to generate the sparks for each firework, the unique difference is that in algorithm 1 we generate normal sparks and the algorithm 2 we generate Gaussian sparks.

Fig. 1 Flow chart of IT2FFWA



We show the algorithm 3 in Fig. 1 [17, 18]

Algorithm 3. Pseudo code of the FWA

```

Randomly select: n locations for Fireworks;
While stop criteria = false do
  Set off x fireworks at the n locations:
  for each firework  $x_i$  do
    Calculate the number of sparks  $\widehat{S}_i$  to the fireworks, Eq. 3
    Obtain locations of  $\widehat{S}_i$  sparks using Algorithm 1;
  end for
  for k = 1:  $\widehat{m}$  do
    Randomly select a firework  $x_j$ ;
    Generate sparks (Gauss) using Algorithm 2;
  end for
  Select the best location and keep it for next explosion;
  Randomly select n - 1 locations from the two types of
  sparks and the current fireworks according to the probability
  given in Eq. 7
end while
  
```

The numbers of the equations and algorithms used in pseudo code of FWA are based on a paper of the developer of this metaheuristic, Ying Tan and Yuanchen Zhu in 2009 [16, 17].

3 Fuzzy Fireworks Algorithm (FFWA)

The Fuzzy Fireworks Algorithm (FFWA) is a variant of the FWA, and in this variant, we converted the coefficient of explosion amplitude, is to say, the coefficient of explosion amplitude in FWA is static (40) and in FFWA is controlled dynamically between a range of 2 and 40. The parameter was originally controlled with a Type-1 Fuzzy Inference System of Mamdani type and we can find the details in the paper [13].

The equation of the explosion amplitude in FFWA is shown below:

$$FA_i = \widehat{FA} \cdot \frac{f(x_i) - y_{\min} + \epsilon}{\sum_{i=1}^n (f(x_i) - y_{\min}) + \epsilon}$$

where $\widehat{FA} = [2, 40]$.

4 Proposed Method (IT2FFWA)

In this proposed method we modify the explosion amplitude, in particular, the coefficient that controls the explosion amplitude of each firework. In recent works [13, 16], we show that adjustment of parameters with Type-1 fuzzy inference system is satisfactory, and then, we decided to improve to control of the same parameter but in this variant we are using Interval Type-2 Fuzzy Logic (IT2FLS), the reason because we are now using IT2FLS is to model a higher degree of uncertainty it contains, so, we can see to we can to control the exploration and exploitation of the algorithm using a fuzzy inference of this type (IT2 FLS).

The proposed method is called as Interval Type 2 Fuzzy Fireworks Algorithm and denoted as IT2FFWA.

Figure 1 shows the enhancement that was made to the FFWA.

The features of the Interval Type-2 Fuzzy System that we added in the Flow chart in IT2FFWA are the following:

IT2FIS

The IT2FIS is of Mamdani type with one input variable and one output variable as shown in Fig. 2.

Input Variable

The input variable is Evaluation in a range of 0 and 1, Evaluations has a partitions of three Gaussian membership functions, which are Low, Medium and High in a range of 0 and 1 (Fig. 3).

Output Variable The output variable is the coefficient of amplitude explosion (AmplitudeCoefficient) in a range of 2 and 40; the AmplitudeCoefficient has a partition of three Gaussian membership functions, which are Small, Medium and Big in a range of 2 and 40 (Fig. 4).

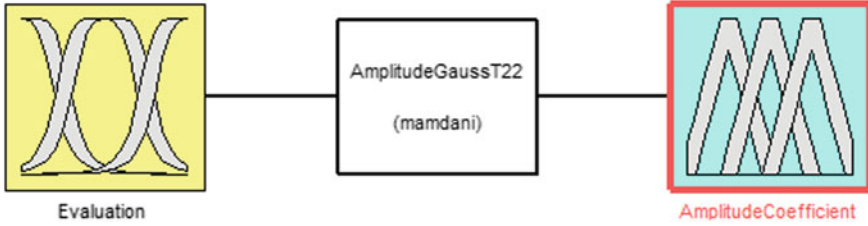


Fig. 2 Graphical representation of IT2FIS

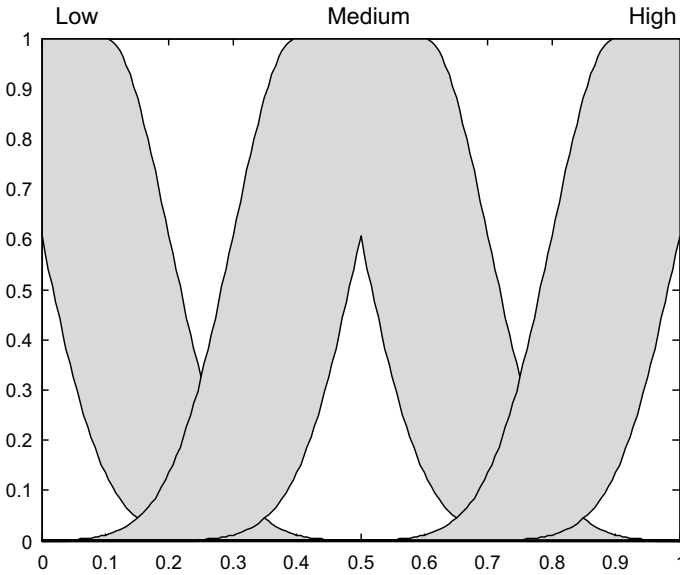


Fig. 3 Input Variable of IT2FIS (Evaluation)

Fuzzy Rules The fuzzy rules of the Interval Type 2 Fuzzy Inference System are the following (Fig. 5).

The aim to use the Interval Type 2 Fuzzy Inference System is to improve the control of the exploration and exploitation into the algorithm [19, 20], in other words, based on fuzzy rules we can control the amplitude coefficient a decrease way [21].

5 Benchmark Functions

The benchmark functions are presented in Tables 1 and 2, respectively.

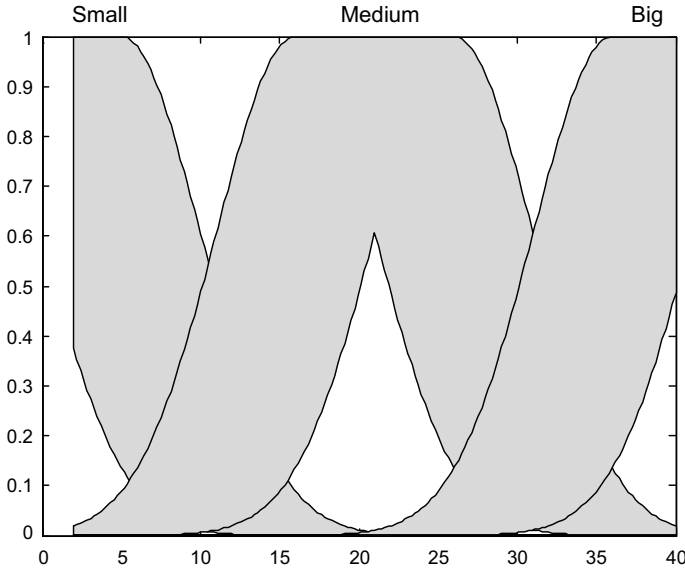


Fig. 4 Output variable of IT2FIS (AmplitudeCoefficient)

Fig. 5 Fuzzy rules of IT2FIS

1. If (Evaluation is Low) then (AmplitudeCoefficient is Big)
2. If (Evaluation is Medium) then (AmplitudeCoefficient is Medium)
3. If (Evaluation is High) then (AmplitudeCoefficient is Small)

6 Experiments and Results

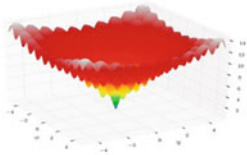
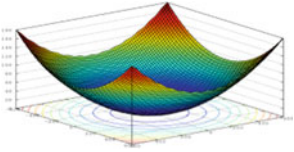
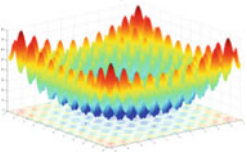
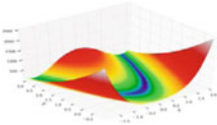
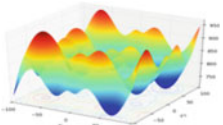
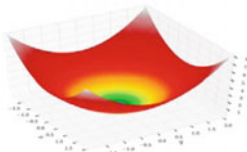
We tested with 12 benchmark functions the performance of IT2FFWA, the first 6 functions have an optimum value in zero and the rest of the functions have an optimum value different to zero. In Table 3 we enlist the name of each benchmark function sequentially from 1 to 12; we also enlist the range of initialization and number of dimensions for each function.

Where the functions: Ackley, Griewank, Rastrigin, Rosenbrock, Schwefel and Sphere have an optimum value equal to zero. The Styblinski function has an optimum value equal to -39.659 , Michalewicz function with optimum value of -9.66 , the optimum value of Easom function is -1 , Goldstein Price has an optimum value in 3 , and the finally the optimum values of Penalized Function P 16 and Six Hump are -330 and -1.0316 , respectively.

As we can see in Table 4, we used 5000, 10000 and 15000 evaluations functions to evaluate both algorithms (FFWA and IT2FFWA) and comparing them.

We also see that IT2FFWA shows a better performance than FFWA, and only in Rosenbrock function with 10000 evaluations function and Schwefel function with 10000 evaluations functions FFWA is better than IT2FFWA.

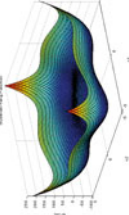
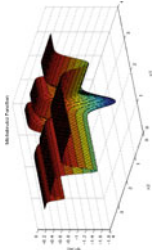
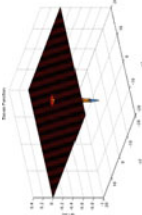
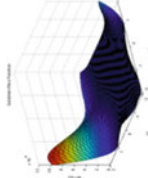
Table 1 Benchmark mathematical functions with optimum equal to zero

Function	Graphic representation
$f_1(x) = -20 \exp\left(-0.2\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	
$f_2(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	
$f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	
$f_4(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_1 - 1)^2]$	
$f_5(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	
$f_6(x) = \sum_{i=1}^n x_i^2$	

The letters in bold shows the better result for the benchmark function with its respective number of evaluations function, the results are the average obtained of the 30 independent runs.

In Table 5 we made a comparison between the conventional Fireworks Algorithm (FWA) and IT2FFWA but with benchmark functions that does not have their optimum value equal to zero. And we can see that in 2 benchmark functions (Styblinski and Michalewicz) the performance of FWA is better than IT2FFWA but in 4 of 6

Table 2 Benchmark mathematical functions with optimum different to zero

Function	Graphic representation
$f_7(x) = \frac{1}{2} \sum_{i=1}^n (x_i^4 - 16x_i^2 + 5x_i)$	
$f_8(x) = - \sum_{i=1}^n \sin(x_i) \sin^{2m} \left(\frac{ix_i^2}{\pi} \right)$	
$f_9(x) = - \cos(x_i) \cos(x_j) (\exp(x_1 - \pi)^2 - (x_2 - \pi)^2)$	
$f_{10}(x) = \left[1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \right] \\ \times \left[30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 - 48x_2 - 36x_1x_2 + 27x_2^2) \right]$	

(continued)

Table 2 (continued)

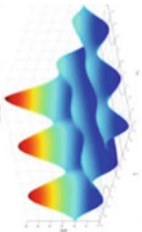
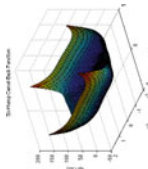
Function	Graphic representation
$f_{11}(x) = 0.1 \left\{ 10 \sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 \left[1 + 10 \sin^2(3\pi x_{i+1}) \right] + (x_d - 1)^2 \left[1 + \sin^2(2\pi x D) \right] \right\} + \sum_{i=1}^n \mu(x_i, 5, 100, 4)$	
$f_{12}(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3} \right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2) x_2^2$	

Table 3 Benchmark functions to simulate results

Function	Range of initialization	Dimensions
Ackley	$[80, 100]^D$	30
Griewank	$[30, 50]^D$	30
Rastrigin	$[80, 100]^D$	30
Rosenbrock	$[30, 50]^D$	30
Schwefel	$[15, 30]^D$	30
Sphere	$[80, 100]^D$	30
Styblinski	$[-5, 5]^D$	30
Michalewicz	$[30, 50]^D$	10
Easom	$[-100, 100]^D$	2
Goldstein price	$[-2, 2]^D$	2
Penalized function p16	$[-50, 50]^D$	30
Six hump	$[-5, 5]^D$	2

Table 4 Comparing of results between FFWA and IT2FFFWA

Function	Evaluations	FFWA	IT2FFFWA
	5000	9.04E-02	1.90E-02
Ackley	10000	6.13E-02	1.17E-02
	15000	5.26E-02	8.79E-03
	5000	2.20E-03	1.25E-04
Griewank	10000	1.20E-03	1.10E-04
	15000	6.00E-04	6.50E-05
	5000	3.903	3.48E-01
Rastrigin	10000	2.294	2.10E-01
	15000	1.694	1.40E-01
	5000	31.654	27.142
Rosenbrock	10000	27.611	27.805
	15000	27.554	24.313
	5000	1.75E-02	9.54E-03
Schwefel	10000	1.50E-03	7.89E-03
	15000	1.30E-03	1.11E-03
	5000	4.21E-02	2.10E-03
Sphere	10000	2.39E-02	1.52E-03
	15000	1.91E-02	1.01E-03

Bold indicates best results

Table 5 Comparing of results between FWA and IT2FFFWA

Function	Evaluations	FWA	IT2FFFWA
Styblinski	10000	-945.722188	-962.48587
Michalewicz	10000	-1.801211	-7.821872
Easom	10000	-0.966446	-0.999826
Goldstein price	10000	3.002846	3.002744
Penalized P16	10000	0.757192	0.624896
Six hump	10000	-1.031554	-1.031566

Bold indicates best results

benchmark functions (Easom, Goldstein Price, Penalized P16 and Six hump) the performance of IT2FFFWA is better than FWA.

7 Conclusions

A way to conclude this work, we can say that used the Interval Type-2 Fuzzy Logic System to control or adjustment the parameters into the some algorithm is quite satisfactory, in this case, we presented in this job, where we implemented Interval Type-2 to adjustment the parameter coefficient of the explosion amplitude into the conventional Fireworks Algorithm and the Fuzzy Fireworks Algorithm.

The results obtained shows that the Interval Type-2 Fuzzy Fireworks Algorithm (IT2FFFWA) have a better performance than conventional FWA or inclusive, a recently variation of the FWA, which is Fuzzy Fireworks Algorithm (FFWA). It is probable that the degree of uncertainty help us to obtained a better control between exploration and exploitation of the algorithm. We also shown that in benchmark functions with optimum equal to zero we obtained the better results with IT2FFFWA. and with the functions with their optimum different to zero, we achieve to obtained better results in half of functions that we tested, 3 of 6 benchmark functions to be exact.

To future work could be test IT2FFFWA with more complex benchmark functions and to perform a comparison with others metaheuristic, as well as, we implement the IT2FFFWA in a control plant. Finally, the use of interval type-2 could be extended to generalized type-2 to verify if results can be improved even more, as in other works [22–31].

References

1. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948 (1995)
2. Zheng, Y., Song, Q., Chen, Y.S.: Multiobjective fireworks optimization for variable-rate fertilization in oil crop production. Appl. Soft Comput. **13**, 4253–4263 (2013)

3. Das, S., Abraham, A., Konar, A.: Swarm intelligence algorithms in bioinformatics. *Stud. Comput. Intell.* **94**, 113–147 (2008)
4. Dorigo, M., Maniezzo, V., Coloni, A.: Ant system: optimization by a colony of cooperating agents. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* **26**(1), 29–41 (1996)
5. Li, J., Zhang, S.: Adaptive fireworks algorithm. *IEEE Congr. Evol. Comput. (CEC)*, pp. 3214–3221 (2014)
6. Tan, Y.: *Fireworks Algorithm*, pp. 355–364. Springer, Berlin, Heidelberg (2015)
7. Tan, Y., Zheng, S.: Enhanced fireworks algorithm. In: *IEEE Congress on Evolutionary Computation*, pp. 2069–2077 (2013)
8. Zadeh, L.A.: Knowledge representation in fuzzy logic. *IEEE Trans. Knowl. Data Eng.* **1**(1), 89–100, Mar 1989
9. Simoes, M., Bose, K., Spiegel, J.: Fuzzy logic based intelligent control of a variable speed cage machine wind generation system. *IEEE Trans. Power Electron.* **12**(1), 87–95 (1997)
10. Rubio, E., Castillo, O.: Interval type-2 fuzzy possibilistic C-means optimization using particle swarm optimization. In: *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 63–78 (2017)
11. Soto, J., Melin, P.: Optimization of the interval type-2 fuzzy integrators in ensembles of ANFIS models for time series prediction: case of the mexican stock exchange. In: *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization*, pp. 27–45 (2015)
12. Zheng, Y., Xu, X., Ling, H., Sheng-Yong, C.: A hybrid fireworks optimization method with differential evolution operators. *Neurocomputing* **148**, 75–82 (2015)
13. Barraza, J., Melin, P., Valdez, F., Gonzalez, C.: Fuzzy FWA with dynamic adaptation of parameters. *IEEE CEC*, pp. 4053–4060 (2016)
14. Tan, Y., Zheng, S.: Dynamic search in fireworks algorithm. In: *Evolutionary Computation (CEC 2014)*, pp. 3222–3229
15. Tan, Y., Zhu, Y.: *Fireworks Algorithm for Optimization*, pp. 355–364. Springer, Berlin, Heidelberg (2010)
16. Barraza, J., Valdez, F., Melin, P., Gonzalez, C.: Fireworks Algorithm (FWA) with adaptation of parameters using fuzzy logic. In: *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 313–327 (2017)
17. Abdulmajeed, N.H., Ayob, M.: A firework algorithm for solving capacitated vehicle routing problem. *Int. J. Adv. Comput. Technol. (IJACT)* **6**(1), 79–86 (2014)
18. Ding, K., Zheng, S., Tan, Y.: A GPU-based Parallel Fireworks Algorithm for Optimization, GECCO'13. Amsterdam, The Netherlands, 6–10 July 2013
19. Liu, M., Mernik, S.H.: Exploration and exploitation in evolutionary algorithms, a survey. *ACM Comput. Surv.* **45**(3), 35, 32 (2013)
20. Perez, J., Valdez, F., Castillo, O.: Modification of the bat algorithm using type-2 fuzzy logic for dynamical parameter adaptation. In: *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 343–355 (2017)
21. Rodriguez, L., Castillo, O., Soria, J.: Grey Wolf Optimizer (GWO) with dynamic adaptation of parameters using fuzzy logic. *IEEE CEC*, pp. 3116–3123 (2016)
22. Rodríguez, L., Castillo, O., Soria, J.: A study of parameters of the grey wolf optimizer algorithm for dynamic adaptation with fuzzy logic. In: *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 371–390 (2017)
23. Castillo, O., Melin, P.: A review on the design and optimization of interval type-2 fuzzy controllers. *Appl. Soft Comput.* **12**(4), 1267–1278 (2012)
24. Castillo, O., Martinez-Marroquin, R., Melin, P., et al.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. *Inf. Sci.* **192**, 19–38 (2012)
25. Castillo, O., Melin, P.: Optimization of type-2 fuzzy systems based on bio-inspired methods: A concise review. *Inf. Sci.* **205**, 1–19 (2012)

26. Melin, P., Castillo, O.: A review on type-2 fuzzy logic applications in clustering, classification and pattern recognition. *Appl. Soft Comput.* **21**, 568–577 (2014)
27. Castillo, O., Melin, P.: A review on interval type-2 fuzzy logic applications in intelligent control. *Inf. Sci.* **279**, 615–631 (2014)
28. Melin, P., Mendoza, O., Castillo, O.: An improved method for edge detection based on interval type-2 fuzzy logic. *Expert Syst. Appl.* **37**(12), 8527–8535 (2010)
29. Hidalgo, D., Melin, P., Castillo, O.: An optimization method for designing type-2 fuzzy inference systems based on the footprint of uncertainty using genetic algorithms. *Expert Syst. Appl.* **39**(4), 4590–4598 (2012)
30. Melin, P., Gonzalez, C.I., Castro, J.R., et al.: Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Trans. Fuzzy Syst.* **22**(6), 1515–1525 (2014)
31. Melin, P., Castillo, O.: A review on the applications of type-2 fuzzy logic in classification and pattern recognition. *Expert Syst. Appl.* **40**(13), 5413–5423 (2013)

Omnidirectional Four Wheel Mobile Robot Control with a Type-2 Fuzzy Logic Behavior-Based Strategy



Felizardo Cuevas, Oscar Castillo and Prometeo Cortés-Antonio

Abstract The main goal considered in this paper is to maintain a specific location and behavior for a robot by using type-2 fuzzy logic for controlling its behavior. We describe the basic concepts about type-2 fuzzy logic, the virtualization of the mobile robot and its modeling according to real situations. Kinematic equations and fuzzy logic are used to control the motor speed of the wheels. The proposed control system is developed in Matlab-Simulink, the system can model and guide a mobile robot, successfully in simulated and real environments. Comparisons of the system performance are made by changing the number of the membership functions in the Type-2 Fuzzy Logic Controller of the robot. The results of the simulation show the advantages of the proposed approach. Also a Mecanum-wheeled robot benefits from great omni-direction maneuverability.

Keywords AMR (Autonomous mobile Robots) · T2FS (Type-2 fuzzy Systems) · T2FLC (Type-2 fuzzy logic Controller) · OMR (Omnidirectional mobile Robot) · Mecanum wheels · Robotics navigation

1 Introduction

In environments where there are obstacles preventing mobility, a platform equipped with mecanum-type wheels allows omnidirectional movements, and with these, greater maneuverability is achieved in highly congested situations [1].

Recent studies on practical applications of vehicles with mecanum type wheels better known as Swedish that are used in environments for load or storage (where spaces sometimes are too small), and this shows how popular the use and study of

F. Cuevas · O. Castillo (✉) · P. Cortés-Antonio
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: ocastillo@tectijuana.mx

F. Cuevas
e-mail: felizardo.cuevas@tectijuana.edu.mx

P. Cortés-Antonio
e-mail: prometeo.cortes@tectijuana.edu.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_4

vehicles have become. In this case autonomous robots with omnidirectional wheels can be very good in these types of environments where maneuvers can be complicated and risky, in the movement of loads [2, 3].

The mobile platform is equipped with four Mecanum wheels (see Fig. 1a). Each wheel has six symmetric bodies of revolution which are 3 rollers placed at an angle around its periphery. The angle between the roller axes and the wheel plane is 45° . However, the control of the movements is difficult, since there are extenuating factors of the speeds, such as the slip due to the type of surface, the reduced contact between the floor and the wheels, and the effect of the distribution of the mass of the robotic platform [4]. Various strategies have been addressed to achieve an appropriate control of robots with mecanum wheels.

In [5–7] some examples are offered. In all of them, kinematic control is preferred over dynamic, which consists on controlling the speed of the motors to achieve the desired movement. The strategy used in this work is to control the robot with mecanum wheels using its kinematics and not its dynamics. The motor control system is based on the use of type-2 fuzzy logic, which has been successfully used in many applications with robots at different levels. Some examples are offered in [8–10].

The design for the first time of the controller by fuzzy logic was proposed initially by Mamdani and Assilian in 1974 [11]. This type of control is used in intuitive problems with expert knowledge and has the disadvantage that it requires a high computational effort.

Currently, one of the main challenges of mobile robotics is the design of models that reliably perform complex tasks in the face of the uncertainty of the environment. For this reason, it is interesting to carry out research works in this area [12]. This makes the realization of works in this field quite interesting.

Figure 1 shows the robotic platform and the navigation scenario in which the mobile platform moves and follows the desired path.

The behavior is an important part for the design of the control to achieve an objective in a set of situations with restrictions. It is here where the need arises in mobile robots when performing a specific task, the development and cooperation of diverse behaviors.

Each behavior is implemented for the control of a task such as, the following of a line, obstacle avoidance of and search for an objective [13].

The main contribution of this work is the application of a Type-2 Fuzzy Logic Controller with the goal of minimizing the orientation and position error within the line that serves as a path to the vehicle. The mathematical models and the control routines were simulated and validated in Matlab/Simulink.

This work is structured as follows: Sect. 2 presents the basic concepts of the 4-wheel omnidirectional robot kinematics. Section 3 presents the design, implementation and simulation of the interval type-2 fuzzy logic controller for the mobile robot. Section 4 presents the results and discussions of the achievements. Section 5 concludes with some final clarifications, conclusions and future work.

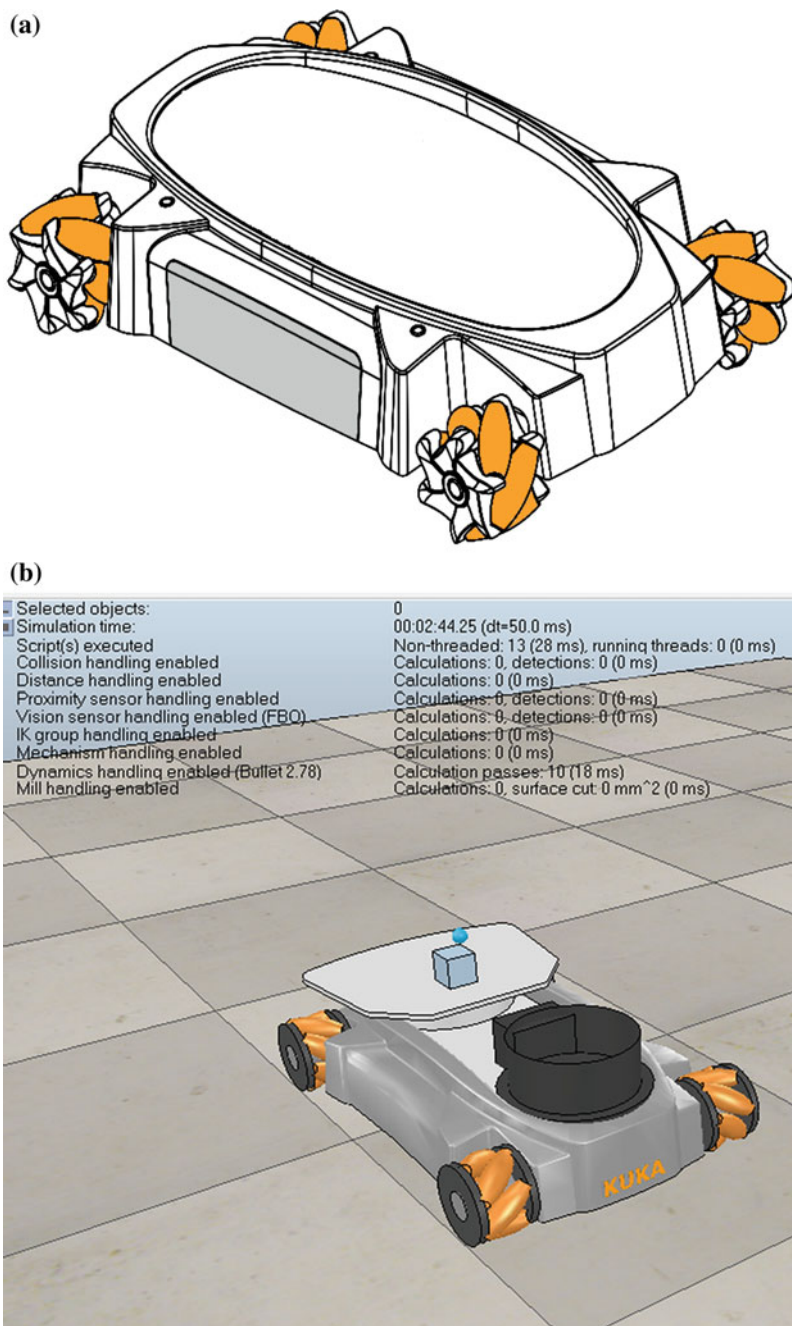


Fig. 1 a Mobile omnidirectional Platform, b Navigation scenario where mobile platform moves

2 Kinematic Model

By varying the directions and speeds of the wheels, the robotic platform can perform omnidirectional movements (see Fig. 2).

Among the applications of kinematics is the possibility of using it to have an initial mathematical model for designing the controller. The following statements outline the limitations in the construction of the kinematic model [6]:

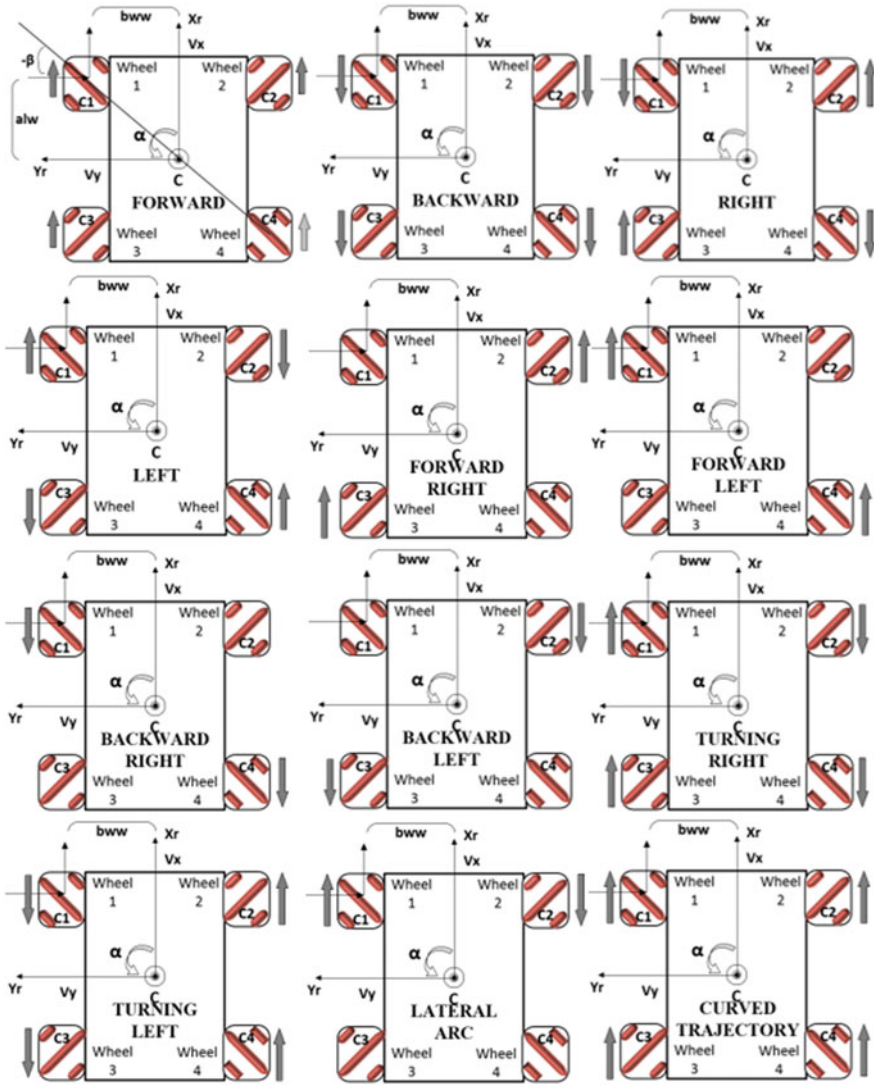


Fig. 2 Movement of the robot according to the direction and angular velocity of the wheels

- The displacement of the robot is assumed to be on a flat surface.
- The structure of the robot does not have flexible elements.
- The steering axes of the wheels are almost always present on the wheels, making them perpendicular to the surface where they are applied.
- Friction is ignored.

2.1 Kinematic Behavior

In the kinematic scheme of the platform presented in Fig. 5b, the rollers of the wheels 1, 2, 3 and 4 in contact with the underlying surface are represented by rectangles; alw denotes half the distance between the axes of the wheel pairs, and bww denotes half the distance between the contact points of the opposite wheels. The radius of the outer surface of the Mecanum wheel is $CiPi = R$, and the radius of the cross section of the roller at the contact point is $KiPi = r$. We assume the contact between the Mecanum wheels and the underlying surface to be ideal: the rollers move without losing of contact with the surface and without slipping, and the center Ci of the i th wheel, the center of the axis of the roller Ki , and the contact point Pi between the roller and the floor are on the same straight line at each instant of time (see Fig. 3).

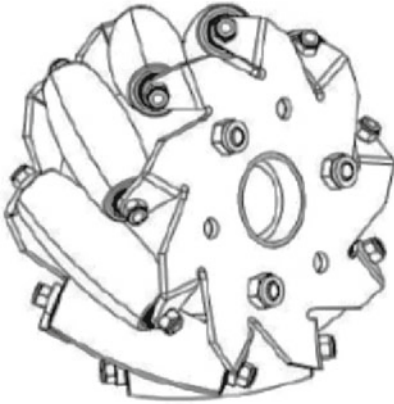
The wheels act as a flat union and this is described below. The wheel is considered to be a rigid element, and it is always in contact with the ground in a unique position that serves as the origin of the reference system explained in Fig. 2. The directions v_x and v_y determine the position or direction of the wheel and represent the linear velocities $x - y$, and w_z is the angular velocity when the robot turns. The conventional wheel in question, has an element v_x that is null, although there are other wheels that provide a different behavior, as shown in Fig. 4.

An omnidirectional wheel is constituted as a standard wheel, one that is equipped with a ring with rollers, with active actuator, passive direction and the rotation axes are perpendicular to the forward direction [9, 10].

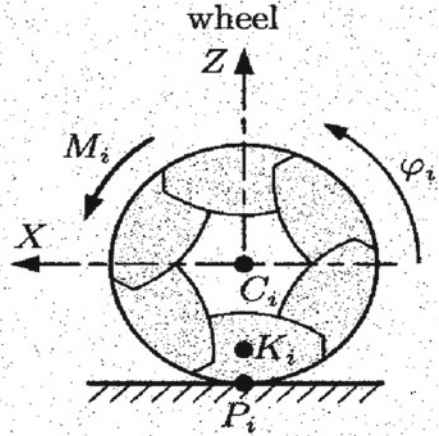
2.2 Kinematic Model

The variables that capture the real speeds to activate the mobile robot are the angular velocities at each of its wheels and are represented by ω_i ($i = 0, 1, 2, 3$) in the kinematics model shown in Fig. 5. Each one of the wheels is driven individually so that the mobile robot is able to perform a lateral or radial movement. The mathematical model of the movement is expressed in the inverse kinematics model of the robotic platform as follows [13].

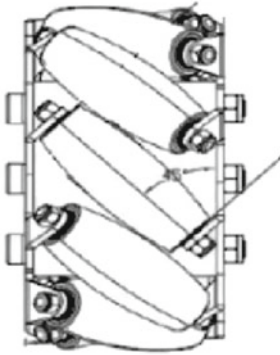
The Inverse Kinematic Model is expressed in the following equation:



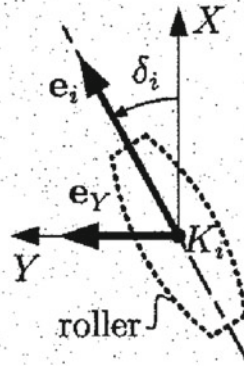
(a) 9 Rollers at a 45° angle



(b) Wheel contact with the floor



(c) Wheel Mounted rollers



(d) Roller kinematics on Wheel

Fig. 3 Representation of Mecanum wheel contact with the floor. **a** Mecanum Wheel. **b** Mecanum wheel contact with the floor. **c** Rollers 45° the mecanum wheel **d** Roller kinematics

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \omega_3 \end{pmatrix} = \begin{pmatrix} 1 & -1 & \frac{(L_x+L_y)}{2} \\ 1 & 1 & \frac{(L_x+L_y)}{2} \\ 1 & -1 & \frac{(L_x+L_y)}{2} \end{pmatrix} \begin{pmatrix} V_X \\ V_Y \\ \omega \end{pmatrix} \tag{1}$$

The Forward Kinematic Model is presented in the following expression:

$$\begin{pmatrix} V_X \\ V_Y \\ \omega \end{pmatrix} = \frac{R}{4} \begin{pmatrix} 1 & 1 & 1 & 1 \\ -1 & 1 & 1 & -1 \\ -\frac{2}{(L_x+L_y)} & \frac{2}{(L_x+L_y)} & -\frac{2}{(L_x+L_y)} & \frac{2}{(L_x+L_y)} \end{pmatrix} \tag{2}$$

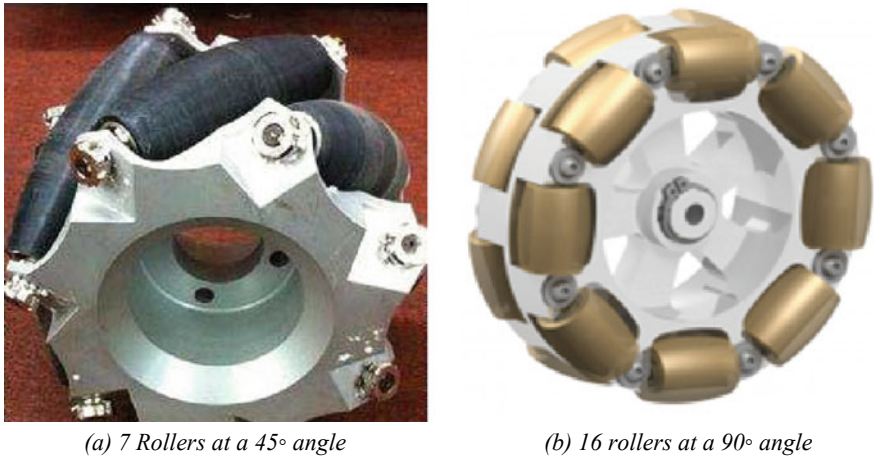


Fig. 4 Omnidirectional wheels types. **a** Mecanum wheel **b** double omnidirectional wheel

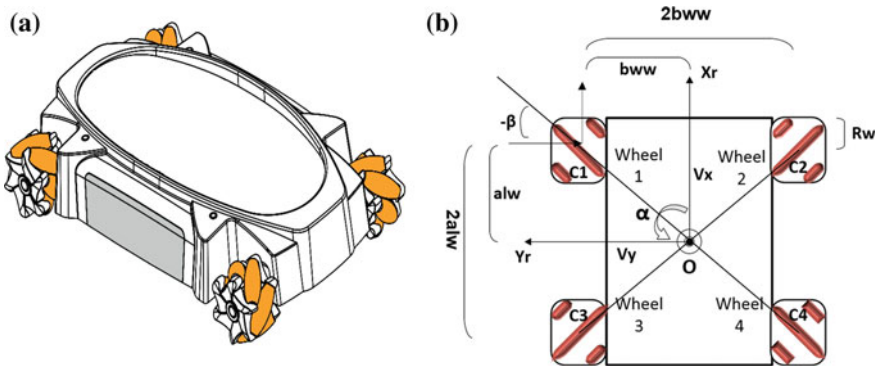


Fig. 5 **a** image of the platform, **b** Kinematic scheme of the robotic platform

where, R_w is the radius of the wheel, L_x and L_y denote the distances from the center of the robot frame to the center of the wheels with respect to the X_r and Y_r axes, respectively. X_r denotes the longitudinal velocity while Y_r denotes the transversal velocity of the mobile robot. ω denotes the angular velocity or the rate on which the mobile robot is turning around its central axis.

3 Interval Type-2 Fuzzy Systems

A fundamental requirement of autonomous mobile robots is a good behavior to avoid obstacles. In the interest of considering the development of an efficient mobile robot

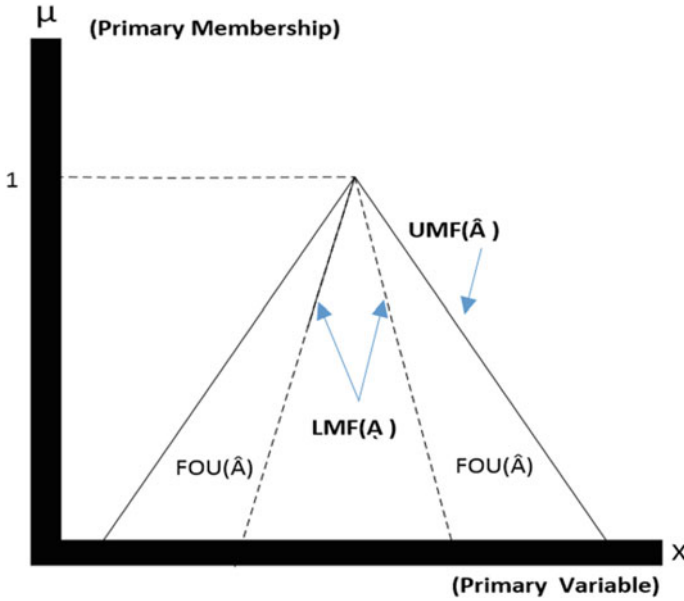


Fig. 6 Upper and Lower MFs for an interval type-2 fuzzy set

controller, a T2FLS is used [12]. It is expected that the T2FLS control algorithm will produce an efficient controller, where we can have the ability to overcome uncertainty and having this ability the robot can plan its movements running efficiently in an unknown and changing environment [14].

For the present work, it is proposed to use the architecture of a type-2 fuzzy Inference System, as illustrated in Fig. 6.

The upper and lower MFs for an interval type-2 fuzzy set (Fig. 6) can be written in Eqs. (3) and (4) respectively. Assuming that we have N rules in a fuzzy rule base of Type 2, they take the following form [15] of Eq. (6):

$$F_{\hat{A}}(x) = \begin{cases} 0, & x < l_1 \\ \frac{x-l_1}{p_1-l_1}, & l_1 < x < p_1 \\ 1, & x \geq p_1 \\ \frac{(r_2-x)}{(r_2-p_2)}, & x \geq p_2 \\ 0, & x \geq r_2 \end{cases} \quad (3)$$

$$F_{\underline{A}}(x) = \begin{cases} 0, & x < l_2 \\ \frac{x-l_1}{p_1-l_2}, & x \leq \frac{r_1(p_2-l_2)}{(p_2-l_2)} + \frac{l_2(r_1-p_1)}{(r_1-p_1)} \\ \frac{(r_2-x)}{(r_2-p_2)}, & x > \frac{r_1(p_2-l_2)}{(p_2-l_2)} + \frac{l_2(r_1-p_1)}{(r_1-p_1)} \\ 0, & x > r_2 \end{cases} \quad (4)$$

$$R^i : \text{If } x_1 \text{ is } X_1^l \text{ and...and } \dots, x_p \text{ is } X_p^l \text{ then } (y \text{ is } Y^l) \quad (5)$$

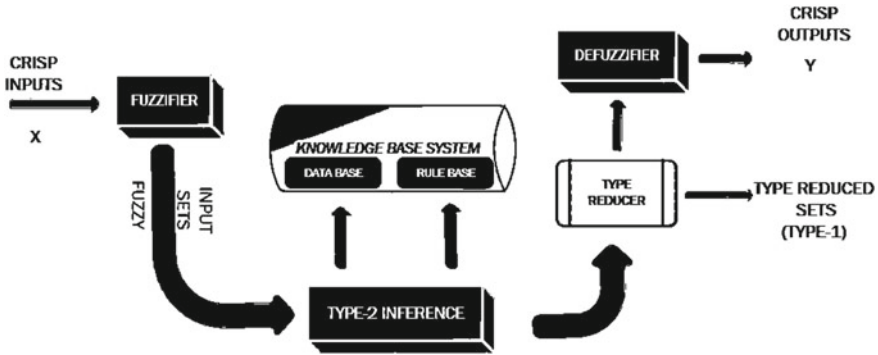


Fig. 7 IT2FLS Architecture

where

$$X_1^l (i = 1, ..p) \text{ and } Y^l \text{ are type - 2 fuzzy sets, and } x = (x^1, \dots, x^l) \text{ and } y \text{ are linguistic variables.}$$

The firing set is defined by interval:

$$F^i(x) = [f^i(x), \bar{f}^i(x) \equiv [f^i, \bar{f}^i] \tag{6}$$

- This IT2FLS handles the uncertainty in the system, while the T1FLS simply does not. Although it is described similarly to a T1FLS, an IT2FLS has some differences when it is described in a block diagram, presented in Fig. 7. The change being shown as it can be seen in the output zone or area, where a Type-Reducer reduces IT2FS to a T1FS, which is then introduced to a Defuzzifier to obtain a crisp result [15].

There are many ways to implement an IT2FLS type reducer and at the moment [12], one of these techniques is the most used, is the center of gravity (cos), where Y_{COS} is an interval, described by the points $[Y_l^i, Y_r^i]$, which are calculated with Eqs. (7) and (8).

Fuzzification

$$y_l = \frac{\sum_{i=1}^M f_l^i y_l^i}{\sum_{i=1}^M f_l^i} \tag{7}$$

$$y_r = \frac{\sum_{i=1}^M f_r^i y_r^i}{\sum_{i=1}^M f_r^i} \tag{8}$$

The defuzzification technique is performed by Eq. (9), which achieves obtaining a crisp output value.

$$y = \frac{y_l + y_r}{2} \quad (9)$$

The variability in each of the actions is modeled with interval type-2 fuzzy sets (T2FS). The T2FS linguistic inputs variables and their ranges are used for the trajectory planning, as shown in Fig. 8a and b, with two outputs which are the linear V_x and V_y speeds which are shown in Fig. 8c and d.

We know that the T2FS is located in a region built by a main type-1 membership function (T1MF). T2FS is obtained by using fuzzy sets to partition the input domains of the base line T1FS with footprint of uncertainty (FOU) as shown in Fig. 6. Consequently, the T1MF is extended to T2MF by adding the FOU in the antecedent and consequent parts of each rule [15].

Therefore, membership functions have values distributed with uncertainty. As well as those that belong to the antecedents also in the consequent parts. From the design, five MFs are for the ERROR X input, five MFs for the ERROR CHANGE, 5 MFs for the Linear V_x speed output and 5 MFs for the Linear V_y Speed output, are used.

Fixed values are assigned in a range of -5 to 5 volts for the ERROR, -5 to 5 volts for CHANGE ERROR to measure how the error changes dynamically, $0-150$ mm/s for the Linear speed V_x and $0-150$ mm/s for the Linear speed V_y depending on the displacement of the mobile robot.

In this work, Twenty-Five rules were developed for T2FLS in the trajectory planning. Rules are used in the control of V_x-V_y linear speeds, shown in Table 1.

These are obtained from the combination of Two inputs with five membership functions as shown in Table 1.

4 Results and Discussion

In this work an evaluation is made to for analyzing the performance of the mobile robot based on T2FLS shown in Fig. 5. The evaluation is carried out in the environment of Matlab/Simulink [15].

The data that should be stored in this simulation is the voltage of the input Error and Error Change, and the linear velocities V_x and V_y . All the data is obtained from the simulation with the help of Simulink, each output data is recorded per unit of time, which in this case is 1 mm/s (Fig. 9).

The control surface of T2FLS with 25 rules is included, whose inputs are the Error and Error Change and the output is the linear V_x and V_y speed, that represent the linear velocities $x - y$, and w_z is the angular velocity. In the development of T2FLS a smooth surface is obtained, which is characterized by the slopes on its surface and

Fig. 8 **a** ERROR X as input MFs. **b** ERROR CHANGE as input MFs. **c** Linear Speed Vx as output MFs. **d** Angular Speed Vy as output MFs

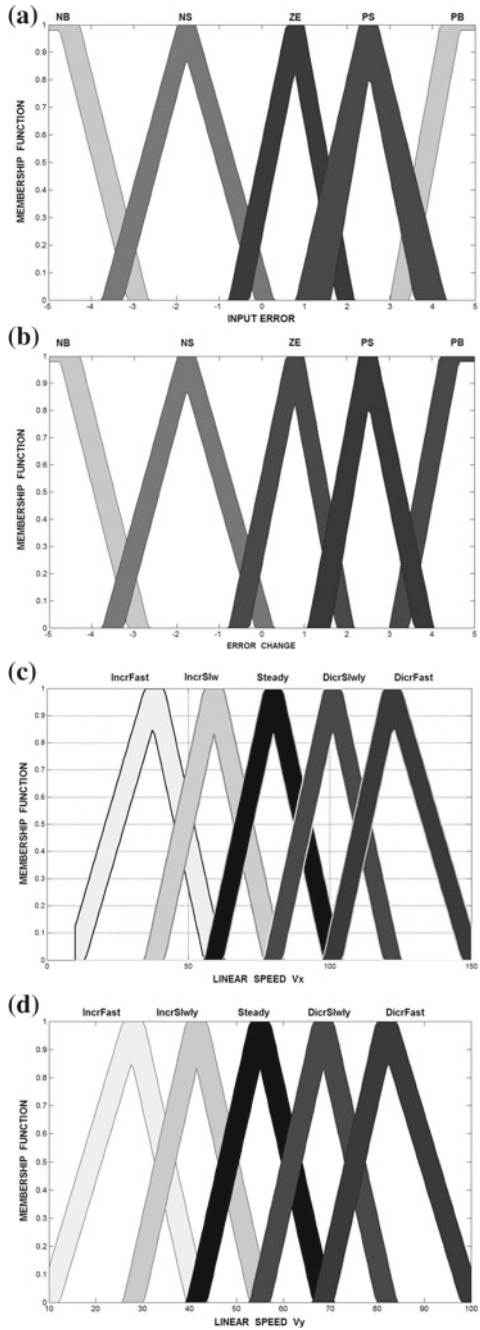


Table 1 Set of fuzzy logic rules with 5 membership functions for e and \hat{e}

e/\hat{e}	NB	NS	ZE	PS	PB
NS	IF/IF	IS/IF	STD/STD	DS/DS	DF/DF
NB	IS/IF	STD/DS	DS/DF	DF/DS	IF/IS
ZE	STD/IS	DS/IS	DF/IS	IF/IS	IS/IF
PS	DS/IS	DF/IS	IF/DF	IS/DS	STD/DS
PB	DF/DS	IF/IS	IS/DS	STD/IS	DS/IS

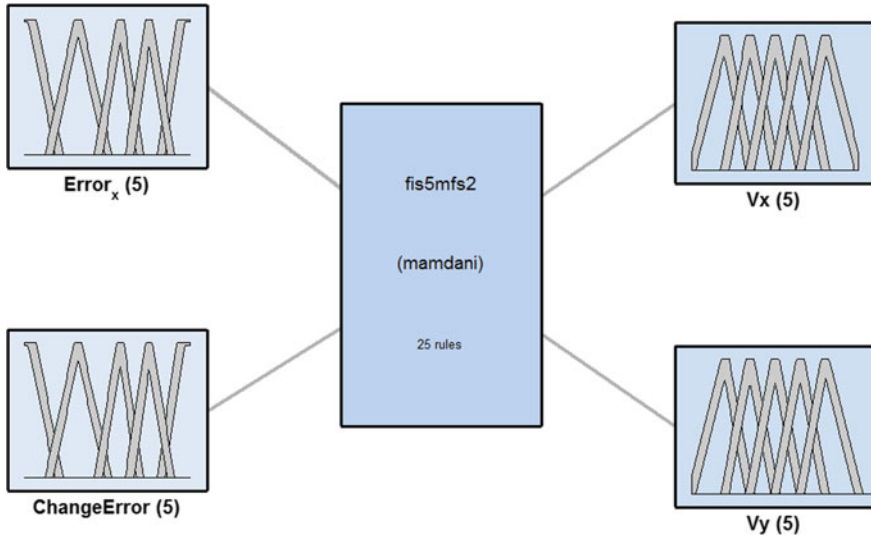


Fig. 9 Interval type-2 fuzzy system with 2 inputs and 2 outputs

each slope has a gradual change represented by the linear velocities, which is shown in Fig. 10a, b.

5 Conclusions

The design of a controller is presented using type-2 fuzzy logic based on the Mamdani method. This design starts from the concept of type-2 fuzzy logic, which is used for the controlling the robotic platform in an efficient way. The Mamdani controller is a good alternative compared to the traditional PID. The fuzzy logic control system with five membership functions is more robust than the control system with fewer membership functions, but with the peculiarity that the computational load increases.

In future work we will use Genetic Algorithms and PSO to optimize and adjust the membership functions of the fuzzy systems and we will also work with neural

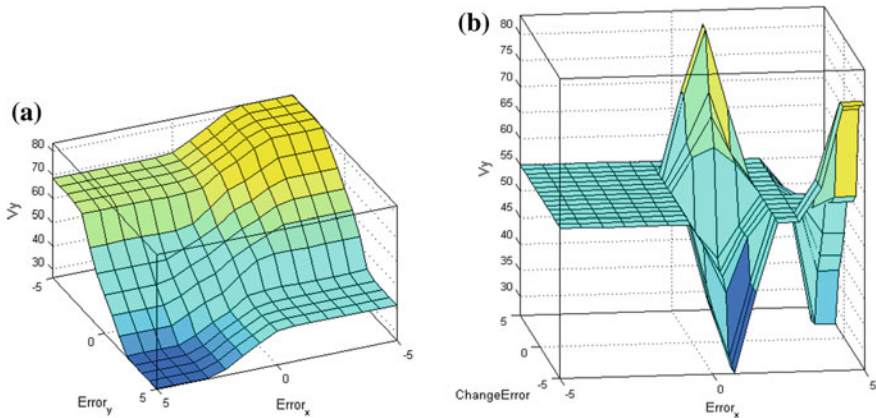


Fig. 10 The control surface of T2FLS with 25 rules, **a** Lineal Speed. **b** Angular speed

networks and perform more experiments. We envision also using generalized type-2 fuzzy logic as in [16–19].

In addition, we can consider in the future other applications of control, like in [21–28] or pattern recognition [29].

References

1. Lih-Chang, L., Hao-Yin, S.: Modeling and adaptive control of an omni-mecanum-wheeled robot. *Intell. Control Autom.* **04**, 166–179 (2013)
2. Taheri, H., Qiao, B., Ghaeminezhad, N.: Kinematic model of a four mecanum wheeled mobile robot. *Int. J. Comput. Appl.* **113**(3), 6–9 (2015)
3. Cuevas, F., Castillo, O.: Design and Implementation of a Fuzzy Path Optimization System for Omnidirectional Autonomous Mobile Robot Control in Real-Time. *Studies in Computational Intelligence*, vol. 749, pp. 241–252. Springer-Verlag (2018)
4. Salih, J.E.M., Rizon, M., Yaacob, S.: Designing omni-directional mobile robot with mecanum wheel. *Am. J. Appl. Sci.* **12**(5), 1831–1835 (2006)
5. Tlale, N., de Villiers, M.: Kinematics and dynamics modeling of a mecanum wheeled mobile platform. *Counc. Sci. Ind. Res.* (2006)
6. Zhao, Y., BeMent, S.L.: Kinematics, dynamics and control of wheeled mobile robots. In: *Proceedings 1992 IEEE International Conference on Robotics and Automation*, pp. 91–96 (1992)
7. Jungmin, K., Jungje, P., Kim, S.: Inertial navigation system for omni-directional AGV with mecanum wheel. *Adv. Mech. Eng.* **2**(1) (2012)
8. S. Jae-Bok, B. Kyung-Seok: Design and control of an omnidirectional mobile robot with steerable omnidirectional wheels. *Mobile robots, moving intelligence*, Buchli J. (ed.) Pro Literatur Verlag, pp. 224–240 (2006)
9. Kanjanawanishkul, K.: Omnidirectional wheeled mobile robots: wheel types and practical applications. *Int. J. Adv. Mechatron. Syst.* **6**, 289 (2015)
10. Assilian, S., Mamdani, E.H.: An experiment in linguistic synthesis with a fuzzy logic controller. *Int. J. Man Mach. Stud.* **7**(1), 1–13 (1974)

11. Castillo, O., Melin, P.: A review on the design and optimization of interval type-2 fuzzy controllers. *Appl. Soft Comput. J.* **12**, 1267–1278 (2012)
12. Castillo, O., Melin, P., Pedrycz, W.: Design of interval type-2 fuzzy models through optimal granularity allocation. *Appl. Soft Comput. J.* **11**, 5590–5601 (2011)
13. P. Ochoa, Castillo, O., Soria, J.: Differential evolution with dynamic adaptation of parameters for the optimization of fuzzy controllers. In: *Studies in Computational Intelligence*, vol. 547, pp. 275–288. Springer-Verlag (2014)
14. Wu, D.: On the fundamental differences between interval type-2 and Type-1 fuzzy logic controllers. *IEEE Trans. Fuzzy Syst.* **20**(5), 832–848 (2012)
15. Diegel, O., Badve, A., Bright, G., Potgieter, J., Tlale, S.: Improved mecanum wheel design for omni-directional robots. In: *Australasian Conference on Robotics and Automation Auckland*, pp. 27–29 (2002)
16. González, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
17. González, C.I., Melin, P., Castro, J.R., Mendoza, O., Castillo, O.: An improved sobel edge detection method based on generalized type-2 fuzzy logic. *Soft. Comput.* **20**(2), 773–784 (2016)
18. Ontiveros, E., Melin, P., Castillo, O.: High order α -planes integration: a new approach to computational cost reduction of general type-2 fuzzy systems. *Eng. Appl. of AI* **74**, 186–197 (2018)
19. Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Applied Soft Computing* **53**, 74–87 (2017)
20. Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C.I., Martinez, G.E., Soto, J.: A fuzzy hierarchical operator in the grey wolf optimizer algorithm. *Appl. Soft Comput. J.* **57**, 315–328 (2017)
21. Martínez-Soto, R., Castillo, O., Aguilar, L.T., Rodríguez, A.: A hybrid optimization method with PSO and GA to automatically design Type-1 and Type-2 fuzzy logic controllers. *Int. J. Mach. Learn. Cybern.* **6**, 175–196 (2015)
22. Amador-Angulo, L., Castillo, O.: Optimization of the type-1 and type-2 fuzzy controller design for the water tank using the bee colony optimization. In: *2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW)*, pp. 1–8. IEEE (2014)
23. Bhojar, D.V., Chilke, B.J., Kemekar, S.S.: Design and analysis of fuzzy PID controllers using genetic algorithm. *Int. Res. J. Eng. Technol. (IRJET)*, **3**, 135–138 (2016)
24. Caraveo, C., Valdez, F., Castillo, O.: Optimization of fuzzy controller design using a new bee colony algorithm with fuzzy dynamic parameter adaptation. *Appl. Soft Comput.* **43**, 131–142 (2016)
25. J. Pérez, Valdez, F., Castillo, O.: Modification of the bat algorithm using type-2 fuzzy logic for dynamical parameter adaptation. In: *Studies in Computational Intelligence*. vol. 667, pp. 343–355. Springer (2017)
26. Castillo, O., Martínez-Marroquín, R., Melin, P., Valdez, F., Soria, J.: Comparative study of bio-inspired algorithms applied to the optimization of type-1 and type-2 fuzzy controllers for an autonomous mobile robot. *Inf. Sci. (Ny)* **192**, 19–38 (2012)
27. Amador-Angulo, L., Mendoza, O., Castro, J., Rodríguez-Díaz, A., Melin, P., Castillo, O.: Fuzzy sets in dynamic adaptation of parameters of a bee colony optimization for controlling the trajectory of an autonomous mobile robot. *Sensors* **16**(9), 1458 (2016)
28. Melin, P., Astudillo, L., Castillo, O., Valdez, F., Garcia, M.: Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm. *Expert Syst. Appl.* **40**, 3185–3195 (2013)
29. Melin, P., González, C.I., Castro, J.R., Mendoza, O., Castillo, O.: Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Trans. Fuzzy Syst.* **22**(6), 1515–1525 (2014)

Optimization for Type-1 and Interval Type-2 Fuzzy Systems for the Classification of Blood Pressure Load Using Genetic Algorithms



Juan Carlos Guzmán, Patricia Melin and German Prado-Arechiga

Abstract In this paper, the design of type-1 and interval type-2 fuzzy systems using genetic algorithm is defined. Fuzzy systems are built from the experience of an expert, in this case a cardiologist. The main contribution of this work is to provide the optimization structure for the classification of the blood pressure load in a patient. The decision was made to use genetic algorithms for the optimization of membership functions, which help to improve the classification and provide a better diagnosis to the patient. In addition, the fuzzy systems have fuzzy rules, which are designed from the categories already defined by an expert. After performing some experiments with different type-1 and type-2 fuzzy systems for the classification of blood pressure load, it was concluded that it is necessary to optimize the membership functions to have better results.

Keywords Fuzzy system · Hypertension · Diagnosis · Blood pressure load · Genetic algorithm

1 Introduction

One of the most frequent and dangerous diseases is hypertension, which is a silent disease, most of the patients who do not have constant blood pressure checks, do not realize that they have the disease, until they have some type of cardiovascular event that puts them on alert or in the worst case happens an unexpected death. For this reason, it is important to perform constant check-ups at home and periodically a 24-h monitoring of blood pressure, in order to have knowledge of the behavior of systolic and diastolic blood pressure, as well as the day and night pressure load, and in this way avoid some problem in the future [1–5].

J. C. Guzmán · P. Melin (✉)
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: pmelin@tectijuana.mx

G. Prado-Arechiga
Excel Medical Center, Tijuana, Mexico

© Springer Nature Switzerland AG 2020
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_5

People check their blood pressure with the specialist, in which the doctor tells them if they have high blood pressure or not, but they omit to check all the other hours during the day and night, which is why the 24-h check-up is a good choice as it allows to obtain valuable information both during day and night, and this helps to know if there are blood pressure loads in these periods of time and to be able to treat them and to prevent possible cardiovascular events in the periods of time with more blood pressure loads [6, 7]. This work is focused on the optimal design of the membership functions of the fuzzy system for the classification of the blood pressure load, in which it is expected that with the implementation of the genetic algorithms, a better result will be obtained [8–13].

The blood pressure load is the percentage above the established ranges which indicate a range for the day and another range for the night. The ranges set for day time blood pressure load is $\geq 135/85$ mmHg and the range set for night time blood pressure load is $\geq 120/70$ mmHg [14–18].

Some previous studies have mentioned the relationship that may exist between the blood pressure load and determination of the damage that an organ can have, such as the left ventricular, among others. Most studies have not given much explanation regarding the correlation that may exist between the blood pressure level and the blood pressure load. That is why in this study we chose to use some intelligent computing techniques to determine an accurate diagnosis [19–22]. The main contribution of this work is to provide the optimization structure for the classification of the blood pressure load in a patient.

The paper is organized as follows: in Sect. 2 the methodology is described, in Sect. 3 the design of the optimization of type-1 and type-2 fuzzy system for the classification of blood pressure load is described, in Sect. 4 Results are summarized and in Sect. 5 the Conclusions are presented.

2 Methodology

2.1 Blood Pressure Load

The measurement of the blood pressure load is a set of measures that results in the 24-h monitoring device, in which the number of readings is analyzed above the ranges established by Table 1. These categories are presented in Table 1.

Table 1 Definitions and classification of the blood pressure load

Blood pressure load categories	Blood pressure load ranges
Normal load	<20
Intermediate load	20–40
High load	>40

Table 2 Example of a summary of blood pressure load record

Day and night period	Time	Interval (min)	Readings	Readings with BP_load	Awake	Asleep
Day period	07–22	20	35	3	8.60%	
Night period	22–07	30	18	15		83.33%
Day BP load			(% of day readings $\geq 135/85$ mmHg)			
Night BP load			(% of night readings $\geq 120/70$ mmHg)			

The Blood pressure (BP) load is obtained from the number of readings over the range established for day and night. In Table 2 we can find an example of a summary of a patient with 35 readings during the day, of which 3 readings are indicated as BP load, and during the night it has 18 readings, of which 15 are taken as BP load. The fuzzy system takes these percentages as inputs and based on the highest percentage given as input, is the level of BP load that is diagnosed.

3 Optimization of Type-1 and Type-2 Fuzzy System for the Classification of Blood Pressure Load

The structure of the type-1 fuzzy system as shown in Fig. 1 consists of the following inputs: Diurnal_Load and Nocturnal_Load and an output called Level_Load. The Diurnal_Load input has the following membership functions (MFs): Normal_Diurnal_Load, Intermediate_Diurnal_Load and High_Diurnal_Load. The Nocturnal_Load input has the following membership functions: Normal_Nocturnal_Load, Intermediate_Nocturnal_Load and High_Nocturnal_Load. For the output Level_Load has the following MFs: Normal,

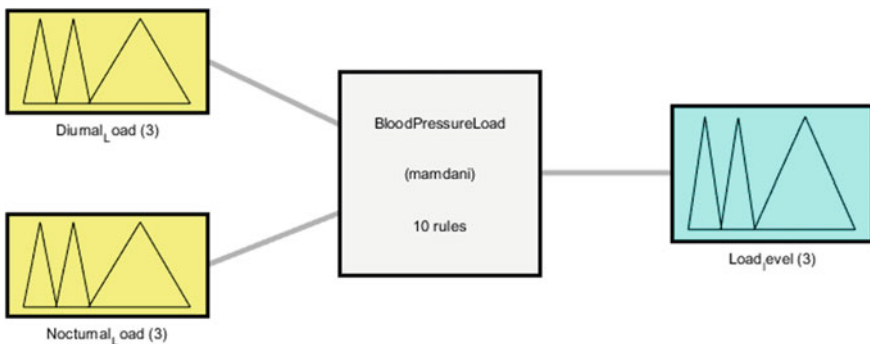


Fig. 1 Structure of the type-1 fuzzy system for classification of blood pressure load with Triangular membership functions

Intermediate and High. The fuzzy system is Mamdani type and has ten fuzzy rules [23, 24].

In the Genetic algorithm, it is necessary to define a chromosome to optimize, in this case the Membership Functions (MFs) as shown in Fig. 2 and the chromosome has 27 genes that help to optimize the MFs, Genes 1–9 allow to manage the parameters of the Diurnal_Load input, Genes 10–18 allow to manage the parameters of the Nocturnal_Load input and Genes 19–27 allow to manage the parameters of the Load_Level output. The following Fig. 2 shows the structure of the chromosome [25–28].

The structure of the interval type-2 fuzzy system is illustrated in Fig. 3 has the following inputs: Diurnal_Load and Nocturnal_Load and an output called Level_Load. The Diurnal_Load input has the following membership functions (MFs): Normal_Diurnal_Load, Intermediate_Diurnal_Load and High_Diurnal_Load. The Nocturnal_Load input has the following membership functions: Normal_Nocturnal_Load, Intermediate_Nocturnal_Load and High_Nocturnal_Load. For the output Level_Load has the following MFs: Normal, Intermediate and High. The fuzzy system is Mamdani type and has ten fuzzy rules.

Diurnal_Load								
Normal_Diurnal_Load			Intermediate_Diurnal_Load			High_Diurnal_Load		
1	2	3	4	5	6	7	8	9
Nocturnal_Load								
Normal_Nocturnal_Load			Intermediate_Nocturnal_Load			High_Nocturnal_Load		
10	11	12	13	14	15	16	17	18
Load_Level								
Normal			Intermediate			High		
19	20	21	22	23	24	25	26	27

Fig. 2 Structure of the chromosome of the type-1 for classification of blood pressure load with Triangular membership functions

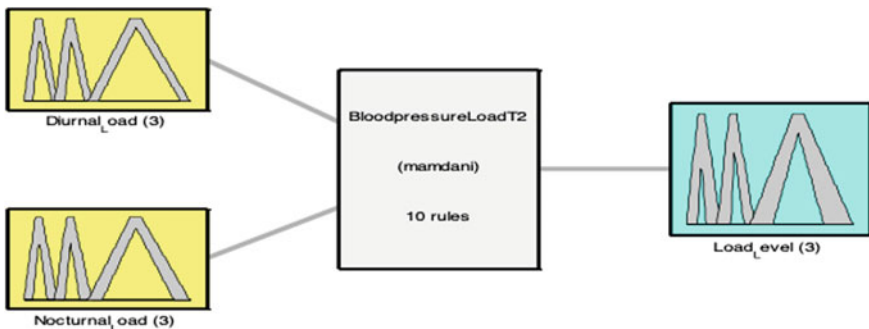


Fig. 3 Structure of the interval type-2 fuzzy system for the classification of blood pressure load

Diurnal_Load																	
Normal_Diurnal_Load						Intermediate_Diurnal_Load						High_Diurnal_Load					
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Nocturnal_Load																	
Normal_Nocturnal_Load						Intermediate_Nocturnal_Load						High_Nocturnal_Load					
19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36
Load_Level																	
Normal						Intermediate						High					
37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54

Fig. 4 Structure of the chromosome for the type-2 for classification of blood pressure load with Triangular membership functions

The chromosome has 54 genes and this help to optimize the MFs, Genes 1–18 allow managing the parameters of the Diurnal_Load input, Genes 19–36 allow managing the parameters of the Nocturnal_Load input and Genes 37–54 allow managing the parameters of the Load_Level output. The following Fig. 4 shows the structure of the chromosome of the type-2 fuzzy system for classification of blood pressure load with Triangular membership functions.

4 Results

Below are the experimented results produced by the type-1 and type-2 BP load fuzzy systems without optimization, and it is expected that when implementing the optimization with genetic algorithms the results will improve, as happened in previous works. It is important to mention that there have been experiments with more patients and some erroneous classifications have been obtained, that is why we opted for the optimization of membership functions. Table 3 shows the results of the type-1 fuzzy system for the classification of blood pressure load without optimization. Finally, Table 4 shows the results for the interval type-2 fuzzy system for the classification of blood pressure load without optimization. row marked in red means classification error for both tables.

5 Conclusions

Fuzzy systems use the intelligence of an expert and perform reasoning through a set of fuzzy rules. This type of systems helps us to be more efficient in less time and thus have an accurate diagnosis of the disease that you want to diagnose, in this case

Table 3 The results for the type-1 fuzzy system for the classification of blood pressure load without optimization

Patient	Diurnal_Load (%)	Nocturnal_Load (%)	Load_Level
1	2	55.3	High
2	8.3	47.8	High
3	55.2	41.8	High
4	8.0	42.0	Intermediate
5	77.0	82.9	High
6	0.0	2.3	Normal
7	76.0	33.3	High
8	0.0	5.0	Normal
9	53.0	56.1	High
10	32.0	23.0	Intermediate
11	46.9	93.9	High
12	54.2	90.0	High
13	62.4	23.0	High
14	3.3	33.8	Intermediate
15	28.6	57.3	High
16	36.8	21.2	Intermediate
17	7.9	5.3	Normal
18	5.3	14.7	Normal
19	46.8	18.8	High
20	17.7	31.8	Intermediate
21	9.9	27.6	Intermediate
22	36.3	16.7	Intermediate
23	9.3	0.0	Normal
24	7.3	31.3	Intermediate
25	55.5	0.0	High
26	71.6	47.1	High
27	23.6	33.3	Intermediate
28	5.8	15.3	Normal
29	61.0	63.0	High
30	0.0	12.0	Normal

the blood pressure load gives enough information to be able to make an accurate decision and avoid a cardiovascular event. We can note that the blood pressure load is very variable both in the day and in the night for some patients, but for others it can be a constant peak in their daytime and nighttime samples, which is why it is important to give emphasis to this type of blood pressure load since this can cause a sudden cardiovascular problem. This is why the use of these fuzzy systems can help the expert to make better decisions at the time of the uncertainty that this

Table 4 The results for the interval type-2 fuzzy system for the classification of blood pressure load without optimization

Patient	Diurnal_Load (%)	Nocturnal_Load (%)	Load_Level
1	2	55.3	High
2	8.3	47.8	High
3	55.2	41.8	High
4	8.0	42.0	Intermediate
5	77.0	82.9	High
6	0.0	2.3	Normal
7	76.0	33.3	High
8	0.0	5.0	Normal
9	53.0	56.1	High
10	32.0	23.	Intermediate
11	46.9	93.9	High
12	54.2	90.0	High
13	62.4	23.0	High
14	3.3	33.8	Intermediate
15	28.6	57.3	High
16	36.8	21.2	Intermediate
17	7.9	5.3	Normal
18	5.3	14.7	Normal
19	46.8	18.8	High
20	17.7	31.8	Intermediate
21	9.9	27.6	Intermediate
22	36.3	16.7	Intermediate
23	9.3	0.0	Normal
24	7.3	31.3	Intermediate
25	55.5	0.0	High
26	71.6	47.1	High
27	23.6	33.3	Intermediate
28	5.8	15.3	Normal
29	61.0	63.0	High
30	0.0	12.0	Normal

type of variation entails for each patient. This is why it is important to optimize the architecture of the fuzzy system to find the best architecture that provides the best classification rate and minimize the classification error in a meaningful way. As future work, we can consider other application as in the works of [29–35].

Acknowledgements We would like to express our gratitude to the CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Kenney, L., Humphrey, R., Mahler, D., Brayant, C.: *ACSM's Guidelines for Exercise Testing and Prescription*. Williams & Wilkins (1995)
2. Texas Heart Institute.: *High Blood Pressure (Hypertension)* (2017)
3. Mancia, G., Grassi, G., Kjeldsen, S.E.: *Manual of hypertension of the European society of hypertension*. Informa Healthcare, United Kingdom (2008)
4. Wizner, B., Gryglewska, B., Gasowski, J., Kocemba, J., Grodzicki, T.: Normal blood pressure values as perceived by normotensive and hypertensive subjects. *J. Hum. Hypertens.* **17**(2), 87–91 (2003)
5. Rosendorff, C.: *Essential Cardiology*, 3rd edn. Springer, Bronx, NY, USA (2013)
6. Battegay, E.J., Lip, G.Y.H., Bakris, G.L.: *Hypertension: Principles and Practices*. Taylor & Francis, Boca Raton, FL (2005)
7. Carretero, O.A., Oparil, S.: Essential hypertension. *Circulation* **101**(3), 329 LP-335, Jan 2000
8. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965)
9. Melin, P., Castillo, O.: *Hybrid Intelligent Systems for Pattern Recognition Using Soft Computing*. Springer-Verlag, Berlin, Heidelberg, Germany (2005)
10. Duodu, Q., Panford, J.K., Ben Hafronacquah, J.: *Designing Algorithm for Malaria Diagnosis using Fuzzy Logic for Treatment (AMDFLT) in Ghana*, vol. 91, no. 17 (2014)
11. Morsi, I., Abd El Gawad, Y.Z.: Fuzzy logic in heart rate and blood pressure measuring system. *IEEE Sensors Appl. Symp. Proc.*, pp. 113–117 (2013)
12. Nohria, R., Mann, P.S.: Diagnosis of hypertension using adaptive neuro-fuzzy inference system. *Int. J. Comput. Sci. Technol.* **8491**, 36–40 (2015)
13. Sikchi, S., Sikchi, S., Ali, M.: Design of fuzzy expert system for diagnosis of cardiac diseases. *Int. J. Med. Sci. Public Heal.* **2**(1), 56 (2013)
14. Oparaku, O., Udo, E.: Fuzzy logic system for fetal heart rate determination. *Int. J. Eng. Res.* **5013**(4), 60–63 (2015)
15. Sadat Asl, A.A., Zarandi, M.H.F.: A type-2 fuzzy expert system for diagnosis of Leukemia BT—fuzzy logic in intelligent system design, pp. 52–60 (2018)
16. Sotudian, S., Zarandi, M.H.F., Turksen, I.B.: From type-I to type-II fuzzy system modeling for diagnosis of Hepatitis **10**(7), 1280–1288 (2016)
17. Pabbi, V.: Fuzzy expert system for medical diagnosis. *Int. J. Sci. Res. Publ.* **5**(1), 1–7 (2015)
18. Mohamed, K.A., Hussein, E.M.: Malaria parasite diagnosis using fuzzy logic. *Int. J. Sci. Res.* **5**(6), 2015–2017 (2016)
19. Miramontes, I., Martínez, G., Melin, P., Prado-Arechiga, G.: A hybrid intelligent system model for hypertension risk diagnosis BT—Fuzzy Logic in Intelligent System Design, pp. 202–213 (2018)
20. Melin, P., Miramontes, I., Prado-Arechiga, G.: A hybrid model based on modular neural networks and fuzzy systems for classification of blood pressure and hypertension risk diagnosis. *Expert Syst. Appl.* **107**, 146–164 (2018)
21. Miramontes, I., Martínez, G., Melin, P., Prado-Arechiga, G.: A hybrid intelligent system model for hypertension diagnosis. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) *BT Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 541–550. Springer, Cham (2017)
22. Guzman, J.C., Melin, P., Prado-Arechiga, G.: Design of an optimized fuzzy classifier for the diagnosis of blood pressure with a new computational method for expert rule optimization. *Algorithms* **10**(3) (2017)
23. Guzmán, J.C., Melin, P., Prado-Arechiga, G.: Neuro-fuzzy hybrid model for the diagnosis of blood pressure. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 573–582. Springer, Cham (2017)
24. Guzmán, J.C., Melin, P., Prado-Arechiga, G.: Design of a fuzzy system for diagnosis of hypertension. In: *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization*, pp. 517–526. Springer, Cham (2015)
25. Melin, P., Prado-Arechiga, G., Miramontes, I., Guzman, J.C.: Classification of nocturnal blood pressure profile using fuzzy systems. *J. Hypertens.* vol. 36 (2018)

26. Guzmán, J.C., Miramontes, I., Melin, P., Prado-Arechiga, G.: Optimal genetic design of type-1 and interval type-2 fuzzy systems for blood pressure level classification. *Axioms* **8**, 8 (2019)
27. O'Brien, E., Parati, G., Stergiou, G.: Ambulatory blood pressure measurement. *Hypertension* **62**(6), 988 LP-994, Nov 2013
28. Sanchez, M.A., Castillo, O., Castro, J.R.: An overview of granular computing using fuzzy logic systems. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) *Nature-Inspired Design Of Hybrid Intelligent Systems*, pp. 19–38. Springer, Cham (2017)
29. Mendez, G.M., Castillo, O.: Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm. In: *The 14th IEEE International Conference on Fuzzy Systems, FUZZ'2005*, pp. 230–235 (2005)
30. Melin, P., Gonzalez, C.I., Castro, J.R., Mendoza, O., Castillo, O.: Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Trans. Fuzzy Syst.* **22**(6), 1515–1525 (2014)
31. Castillo, O., Melin, P., Ramírez, E., Soria, J.: Hybrid intelligent system for cardiac arrhythmia classification with Fuzzy K-Nearest Neighbors and neural networks combined with a fuzzy system. *Expert Syst. Appl.* **39**(3), 2947–2955 (2012)
32. González, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
33. González, C.I., Melin, P., Castro, J.R., Mendoza, O., Castillo, O.: An improved sobel edge detection method based on generalized type-2 fuzzy logic. *Soft. Comput.* **20**(2), 773–784 (2016)
34. Ontiveros, E., Melin, P., Castillo, O.: High order α -planes integration: a new approach to computational cost reduction of general type-2 fuzzy systems. *Eng. Appl. of AI* **74**, 186–197 (2018)
35. Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Appl. Soft Comput.* **53**, 74–87 (2017)

Intuitionistic Fuzzy Logic

Interval Valued Intuitionistic Fuzzy Evaluations for Analysis of Students' Knowledge



Evdokia Sotirova, Anthony Shannon, Vassia Atanassova,
Krassimir Atanassov and Veselina Bureva

Abstract In the paper is proposed a method for evaluation of student's knowledge obtained in the university e-learning courses. For the assessment of the student's solution of the respective assessment units the theory of intuitionistic fuzzy and interval valued intuitionistic fuzzy sets is used. The obtained intuitionistic fuzzy estimations reflect the degree of each student's good performances, or poor performances, for each assessment unit and for all units.

Keywords e-Learning · Interval valued intuitionistic fuzzy evaluations · Intuitionistic fuzzy evaluation · Model

1 Introduction

In the paper, the process of e-learning within university courses is described as a continuation of previous investigations of the authors into the modelling of a basic processes and functions of a typical university (see [6–16]). In their research, the authors used the apparatus of the Generalized Nets (GNs, see [1]). For example, GN-models describing the process of student assessment is discussed in [6, 11, 12].

E. Sotirova (✉) · K. Atanassov · V. Bureva
Intelligent Systems Laboratory, University “Prof. Dr. Assen Zlatarov”, Burgas, Bulgaria
e-mail: esotirova@btu.bg

K. Atanassov
e-mail: krat@bas.bg

V. Bureva
e-mail: vbureva@btu.bg

A. Shannon
Warrane College, The University of New South Wales, Kensington, NSW, Australia
e-mail: t.shannon@warrane.unsw.edu.au

V. Atanassova · K. Atanassov
Department of Bioinformatics and Mathematical Modelling, IBPhBME, Bulgarian Academy of Sciences, Sofia, Bulgaria
e-mail: vassia.atanassova@gmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_6

There, the assessments are represented in the form of intuitionistic fuzzy and interval valued intuitionistic fuzzy sets. The concept of Intuitionistic Fuzzy Set (IFS) was described in details in [2–4] and of Interval Valued IFS (IVIFS)—in [3]. The process of evaluation of the problems solved by students is described in [6]. In [10], the process of evaluation by lecturers of the tasks presented by students is described. In another paper [11], a model was constructed, that describes the process of evaluation by lecturers. A model that represents the standardization of the process of evaluation by lecturers was constructed [12]. In the next steps of the educational processes there is a modelling of the evaluation of the lecturers themselves [13], and the assessment of the course [14]. In [15, 16], the intuitionistic fuzzy assessments for the Analysis of a Student’s Knowledge in University e-Learning Courses are supposed. This paper continues the research in [7, 15, 16].

Within the context of e-learning, the information exchange between the education and training system and the student is performed electronically. The student obtains information on a given topic at his/her local electronic device. After this, the student’s knowledge can be rated by asking appropriate questions and giving appropriate problems, in order to pass on to the next topic of training.

During the process of e-learning the students have access to different training materials that can be classified as [5]:

- information units to acquire knowledge and skills;
- assessment units—task, problems, test;
- information resources—library, internet, and so on.

2 Proposed Assessment Model

Extending and correcting some misprints, and keeping the notation from [16], here we discuss the process of analysing the student’s knowledge, which is realized on two phases.

First we determine the evaluations of the assessment units for each discipline studied by the students. Then we evaluate the final mark for each student using weight coefficients of the different assessment units and the obtained evaluations for them.

Let us consider a group of m students, the students are labelled as follows $i = 1, 2, \dots, m$ and w disciplines $t = 1, 2, \dots, w$. The disciplines have to be evaluated via j assessment units, $j = 1, 2, \dots, n$. The assessments, which estimate a summative account of the students’ knowledge for the different problems, are formed on the basis of a set of intuitionistic fuzzy estimations $\langle \mu, \nu \rangle$ of real numbers from the set $[0, 1] \times [0, 1]$, related to the respective assessment units. These intuitionistic fuzzy estimations reflect the degree of each student’s good performances μ , or poor performances ν , for each assessment unit, and for them is valid that

$$\mu + \nu \leq 1.$$

The degree of uncertainty $\pi = 1 - \mu - \nu$ represents such cases wherein the student is currently unable to solve the problem and needs additional information. Within the paper the ordered pairs were defined in the sense of intuitionistic fuzzy sets.

2.1 Determination of the Students' Assessments of the Different Units for a Discipline

The way of evaluation of the different units for a discipline can vary, but for some groups of themes (e.g., mathematics, informatics, physics, chemistry, and etc.), the evaluations of the students' solutions of the different problems can be obtained, in general, by two cases:

Case 1. The assessment unit j for discipline t contains u^j in number subtasks (questions).

In this case the assessment unit can be in the form of a test with questions with attached possible answers "yes" and "no", or with questions with an attached list of optional answers.

Thus, the evaluation of the i th student for the j th assessment unit is obtained in two ways according to the following formula (1), for $i = 1, 2, \dots, m, j = 1, 2, \dots, n$:

$$\langle \mu^j(i), \nu^j(i) \rangle = \left\langle \frac{r^j(i)}{u^j}, \frac{s^j(i)}{u^j} \right\rangle, \tag{1}$$

where

- $r^j(i)$ is the number of right answers of the subtasks/questions in the assessment unit j ,
- $s^j(i)$ is the number of wrong answers of the subtasks/questions in the assessment unit j ,
- u^j is the total number of subtasks/questions in the assessment unit j .

Therefore, the degree of uncertainty in this case is determined by the number of the questions which the student had not worked over.

Case 2. The assessment unit j , for example one task, is evaluated independently for w^j levels.

Initially, when there has been no information obtained for the assessment unit, the estimation is given by the initial values $\langle 0, 0 \rangle$. For $k \geq 0$, the current (k)st estimation of the i th student for the j th assessment unit is obtained on the basis of the previous estimations according to the recurrence relation involved in the following formula (2), $i = 1, 2, \dots, m, j = 1, 2, \dots, n$.

$$\langle \mu_k^j(i), \nu_k^j(i) \rangle = \left\langle \frac{(k-1) \cdot \mu_{k-1}^j(i) + a_{p_i}^j(i)}{k}, \frac{(k-1) \cdot \nu_{k-1}^j(i) + b_{p_i}^j(i)}{k} \right\rangle, \tag{2}$$

where

- $\langle \mu_{k-1}^j(i), \nu_{k-1}^j(i) \rangle$ is the previous estimation of the j th assessment unit of the i th student on the basis of the solutions of the already solved subtasks in the completed levels,
- $\langle a_{p_l}^j(i), b_{p_l}^j(i) \rangle$ is the estimation of the level p_l of the j th assessment unit of the i th student, for $a_{p_l}^j(i), b_{p_l}^j(i) \in [0, 1]$, $a_{p_l}^j(i) + b_{p_l}^j(i) \leq 1$, and $l = 1, 2, \dots, w^j$.
- $a_{p_l}^j(i)$ and $b_{p_l}^j(i)$ are calculated according to (3) and (4) in the following way:

$$a_{p_l}^j(i) = \begin{cases} \frac{c_l^j(i) + d_l^j(i)}{p_l^j}, & \text{if the } i\text{th student had worked over level } p_l^j, \\ 0, & \text{if the } i\text{th student had not worked over level } p_l^j \end{cases}, \quad (3)$$

$$b_{p_l}^j(i) = \begin{cases} \frac{p_l^j - (c_l^j(i) + d_l^j(i))}{p_l^j}, & \text{if the } i\text{th student had worked over level } p_l^j, \\ 0, & \text{if the } i\text{th student had not worked over level } p_l^j \end{cases}, \quad (4)$$

where

- $c_l^j(i)$ are the points for the solution of the level p_l^j of the j th assessment unit of the i th student,
- $d_l^j(i)$ are the points for the description of the decision of the level p_l^j of the j th assessment unit of the i th student.

Therefore, the degree of uncertainty, in this case, is equal to 1, when the i th student did not work over the level p_l^j of the j th assessment unit.

2.2 Determine of the Final Mark for the i th Student for the t th Discipline

Here we introduce intuitionistic fuzzy coefficients $\langle \delta, \varepsilon \rangle$, setting weights of each assessment unit that contribute to the final mark for the i th student, $i = 1, 2, \dots, m$. Coefficient δ is based on the number of successive assessment units, and coefficient ε is based on the number of preceding assessment units. An example can clarify this. Suppose, for instance, that a trainee sits for eight assessment units, divided into three levels of difficulty (easy, average, difficult). Let there be three assessment units from the first level, three assessment units from the second level, and two assessment units from the third level. Then the weight coefficients will be distributed as follows:

- from the first level: $\langle \frac{5}{8}, 0 \rangle$,
- from the second level: $\langle \frac{2}{8}, \frac{3}{8} \rangle$, and
- from the third level: $\langle 0, \frac{6}{8} \rangle$.

In this way, the $(j + 1)$ st intuitionistic fuzzy estimation $\langle \mu^{j+1}(i), \nu^{j+1}(i) \rangle$, is calculated on the basis of the preceding estimations $\langle \mu^j(i), \nu^j(i) \rangle$ is obtained according

to the following formula (5), for $i = 1, 2, \dots, m, j = 1, 2, \dots, n$.

$$\langle \mu^{j+1}(i), v^{j+1}(i) \rangle = \left\langle \frac{\mu^j(i) \cdot j + \delta^j \cdot \varphi + \varepsilon^j \cdot \phi}{j + 1}, \frac{v^j(i) \cdot j + \delta^j \cdot \phi + \varepsilon^j \cdot \varphi}{j + 1} \right\rangle, \quad (5)$$

where $\langle \varphi, \phi \rangle$ is the estimation of the current assessment unit, $\varphi, \phi \in [0, 1]$ and $\varphi + \phi \leq 1$, and $\langle \delta^j, \varepsilon^j \rangle$ is the weight coefficients of the j th assessment unit, for $\delta^j, \varepsilon^j \in [0, 1], \delta^j + \varepsilon^j \leq 1$.

The calculated final mark based on all assessment units for the i th student has to satisfy the necessary “minimal threshold of knowledge”. To check this, we introduce the threshold values: $M_{\max}, M_{\min}, N_{\max}, N_{\min}$.

If

$$\mu(i) > M_{\max} \ \& \ v(i) < N_{\min},$$

then the i th student satisfies the “minimal threshold of knowledge” for the current e-learning course.

If

$$\mu(i) < M_{\min} \ \& \ v(i) > N_{\max},$$

then the i th student does not satisfy the “minimal threshold of knowledge” for the current e-learning course and he/she has to be evaluated for all assessment units again.

In the rest of the cases the “minimal threshold of knowledge” is undefined and the i th student has to be evaluated again for the assessment units for which:

$$\mu^j(i) \leq M_{\max} \ \& \ v^j(i) \geq N_{\min}$$

is valid.

2.3 Determine of Interval Valued Intuitionistic Fuzzy Evaluation of a Student for All Disciplines

Here the interval valued intuitionistic fuzzy evaluation for a student in the form $\langle P, Q \rangle$ is proposed, $P, Q \subset [0, 1]$ are intervals, i.e., $P = [\inf P, \sup P], Q = [\inf Q, \sup Q]$ and

$$\sup P + \sup Q \leq 1.$$

Using the intuitionistic fuzzy evaluations of the separate disciplines that students have to learn, we can obtain an aggregated evaluation for the i th student.

The final evaluation for the i th student ($i = 1, 2, \dots, m$) that has to learn d_i disciplines ($x_i = 1, 2, \dots, d_i$, and $d_i \leq w$) according to formulas (1), (2) and (5) has the form

$$\langle \mu_{x_i,i}, v_{x_i,i} \rangle = \left\langle \frac{\sum_{q=1}^{d_i} \mu_{x_i,i}}{d_i}, \frac{\sum_{q=1}^{d_i} v_{x_i,i}}{d_i} \right\rangle.$$

When we like to obtain interval valued intuitionistic fuzzy evaluation of the d_i th discipline, we construct numbers

$$\begin{aligned} P_{i,x,inf} &= \inf \{ \mu_{x_i,i} | x_i = 1, 2, \dots, d_i \}, \\ P_{i,x,sup} &= \sup \{ \mu_{x_i,i} | x_i = 1, 2, \dots, d_i \}, \\ Q_{i,x,inf} &= \inf \{ v_{x_i,i} | x_i = 1, 2, \dots, d_i \}, \\ Q_{i,x,sup} &= \sup \{ v_{x_i,i} | x_i = 1, 2, \dots, d_i \}. \end{aligned}$$

Now, we can construct the interval valued intuitionistic fuzzy evaluation for the i th student

$$\langle P_{i,x}, Q_{i,x} \rangle = \langle [P_{i,x,inf}, P_{i,x,sup}], [Q_{i,x,inf}, Q_{i,x,sup}] \rangle.$$

When we like to compare two disciplines (x th and y th) on the basis of the results, we can use the following formulas (see [3]):

$$\begin{aligned} \langle P_z, Q_z \rangle \leq \langle P_y, Q_y \rangle \text{ if and only if} \\ P_{z,inf} \leq P_{y,inf}, P_{z,sup} \leq P_{y,sup}, Q_{z,inf} \geq Q_{y,inf}, Q_{z,sup} \geq Q_{y,sup}. \end{aligned}$$

Now, to the student class we can juxtapose the IVIFS

$$\{ \langle i, P_{i,x}, Q_{i,x} \rangle | i = 1, 2, \dots, m \}.$$

3 Conclusion

In the current paper we have presented the procedure that gives the possibility for the algorithmization of the method of forming the student's evaluations by applying intuitionistic fuzzy estimations. The suggested evaluation methodology and procedures are intended to make the student's evaluations as objective as possible.

In practice, subjective estimation cannot be entirely avoided but it should be made as objective as possible. This can be achieved, to some extent, by approaches which use quantitative methods to utilize the instruments of subjective statistics.

In addition, the discussed procedure can be extended for arbitrary facts having intuitionistic fuzzy or interval valued intuitionistic fuzzy evaluations, using the same manner.

Acknowledgements This work was supported by the Bulgarian Ministry of Education and Science under the National Research Programme “Information and Communication Technologies for a Digital Single Market in Science, Education and Security” approved by DCM # 577/17.08.2018.

References

1. Atanassov, K.: Generalized Nets. World Scientific (1991)
2. Atanassov, K.: Intuitionistic Fuzzy Sets. Springer, Heidelberg (1999)
3. Atanassov, K.: On Intuitionistic Fuzzy Sets Theory. Springer, Berlin (2012)
4. Atanassov, K.: Intuitionistic fuzzy logics as tools for evaluation of data mining processes. *Knowl. Based Syst.* **80**, 122–130 (2015)
5. Kensington-Miller, J. Novak, T. Evans. Just do it: flipped lecture, determinants and debate. *Int. J. Math. Educ. Sci. Technol.* **47**(6), 853–862 (2016)
6. Melo-Pinto, P., Kim, T., Atanassov, K., Sotirova, E., Shannon, A., Krawczak, M.: Generalized net model of e-learning evaluation with intuitionistic fuzzy estimations. In: *Issues in the Representation and Processing of Uncertain and Imprecise Information*, Warszawa, pp. 241–249 (2005)
7. Kim, T., Sotirova, E., Shannon, A., Atanassova, V., Atanassov, Kr., Jang, L.-C.: Interval Valued intuitionistic fuzzy evaluations for analysis of a student’s knowledge in university e-learning courses. *Int. J. Fuzzy Logic Intell. Syst.* **18**(3), 190–195
8. Shannon, A., Langova-Orozova, D., Sotirova, E., Petrounias, I., Atanassov, K., Krawczak, M., Melo-Pinto, P., Kim, T.: Generalized Net Modelling of University Processes. KvB Visual Concepts Pty Ltd., Sydney (2005)
9. Shannon, A., Atanassov, K., Sotirova, E., Langova-Orozova, D., Krawczak, M., Melo-Pinto, P., Petrounias, I., Kim, T.: Generalized Nets and Information Flow Within a University, Warszawa (2007)
10. Shannon, A., Sotirova, E., Petrounias, I., Atanassov, K., Krawczak, M., Melo-Pinto, P., Kim, T.: Intuitionistic fuzzy estimations of lecturers’ evaluation of student work. In: *First International Workshop on Intuitionistic Fuzzy Sets, Generalized Nets & Knowledge Engineering*, University of Westminster, London, pp. 44–47, 6–7 Sept 2006
11. Shannon, A., Sotirova, E., Petrounias, I., Atanassov, K., Krawczak, M., Melo-Pinto, P., Kim, T.: Generalized net model of lecturers’ evaluation of student work with intuitionistic fuzzy estimations. In: *Second International Workshop on Intuitionistic Fuzzy Sets. Notes on IFS*, Banska Bystrica, Slovakia, vol. 12(4), pp. 22–28, 3 Dec 2006
12. Shannon, A., Sotirova, E., Atanassov, K., Krawczak, M., Melo-Pinto, P., Kim, T.: Generalized net model for the reliability and standardization of assessments of student problem solving with intuitionistic fuzzy estimations. In: *Developments in Fuzzy Sets, Generalized Nets and Related Topics. Applications*, vol. 2, pp. 249–256. System Research Institute, Polish Academy of Science (2008)
13. Shannon, A., Dimitrakiev, D., Sotirova, E., Krawczak, M., Kim, T.: Towards a model of the digital university: generalized net model of a lecturer’s evaluation with intuitionistic fuzzy estimations. *Cybern. Inform. Technol.* **9**(2), 69–78 (2009)
14. Shannon, A., Sotirova, E., Hristova, M., Kim, T.: Generalized net model of a student’s course evaluation with intuitionistic fuzzy estimations in a digital university, vol. 13(1), pp. 31–38. In: *Proceedings of the Jangjeon Mathematical Society* (2010)

15. Sotirova, E., Atanassov, K., Shannon, A., Kim, T., Krawczak, M., Melo-Pinto, P., Riečan, B.: Intuitionistic fuzzy evaluations for analysis of a student's knowledge of mathematics in the university e-learning courses. In: IEEE 8th International Conference on Intelligent Systems, pp. 535–537 (2016)
16. Sotirova, E., Shannon, A., Kim, T., Krawczak, M., Melo-Pinto, P., Riečan, B.: Intuitionistic fuzzy evaluations for the analysis of a student's knowledge in university e-learning courses. In: Studies in Computational Intelligence, vol. 757, pp. 95–100 (2018)

Generalized Net Model of the Network for Automatic Turning and Setting the Lighting in the Room with Intuitionistic Fuzzy Estimations



Tihomir Videv, Sotir Sotirov and Boris Bozveliev

Abstract Turning the lights on automatically helps people who enter a current room to turn on the lighting depending on the settings made by a person. In this paper we will take a look how by scanning the iris the system recognizes the person who is entering the room depending on if the the outside light exceeds the set in advance threshold value, the lighting turns on with the settings for lighting of a specific person and it also lets corrections to be made in the set in advance parameters.

Keywords Generalized net · Lighting systems · Tthreshold value · Scan the iris · Lighting settings

1 Introduction

The current paper takes a look of the generalized network (GN) [1, 2] model of the general lighting system in the room which identifies a person and turns on the lighting according to his requirements. Generalized nets and index matrix are tools with which we can describe the processes running through the system very detailed [3–7]. Here you can model processes that work parallel. With the help of the generalized nets are modeled many optimization algorithms [8–15], as well as processes taking place in health and education establishments [16–18,]. A part of the model is connected with intelligent systems as well as neuron nets [19] genetic and other algorithms [20–25]. As there are other lighting systems with identifications [26] we will use a system which scans the iris for identification of a person and we will show all parallel processes which go through it. Here we have to mention that all processes for identification [26, 27] are connected with ensuring the lighting settings in the room. The identification [26, 28] is an important factor for impeccable work, the GN model [29] helps us to easily analyze the lighting system and to recognize

T. Videv (✉) · S. Sotirov · B. Bozveliev
“Prof. Dr. Assen Zlatarov” University, “Prof. Yakimov” Blvd, Burgas 8010, Bulgaria
e-mail: tvidev@abv.bg

B. Bozveliev
e-mail: bozveli@gmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_7

possible omissions in the system. Now people are forced to use multiple lights for reading a book, watching television, work on a computer cleaning or just for relaxing [30]. The way of which the system [29] works is explained in a few simplification steps: A person has to create “user settings” with the preferable lighters as well as lighting and appearance so that the system can turn the lights on when it identifies the person with specific preferences. He will also need corrections or more changes in the lighting threshold. Identification data [26, 27] as well as the lighting settings are being stored in the database of the users account. When entering the room the system scans the iris of the person and if the momentary lighting in the room does on exceed the threshold value it turns on the lights with the made in advance settings of the person. Also you can correct the lighting if needed so that the next time it is turned on the corrections will be remembered. In this paper we are taking a look at a general model of a lighting system [29, 31] the GN model [1, 2] will help us in order to easily and simply understand the main way of usage and this way we can improve the efficiency as well as troubleshoot the system and make a better analysis. In the light of the thoughts above we need to evaluate the possibilities of failure of the lighting smart system. For this reason we are going to use fuzzy sets(IFS). The Intuitionistic Fuzzy Sets (IFSs, see [32–35]) represent an extension of the concept of fuzzy sets, as defined by Zadeh [35], exhibiting function $\mu_A(x)$ defining the membership of an element x to the set A , evaluated in the $[0; 1]$ -interval. The difference between fuzzy sets and intuitionistic fuzzy sets (IFSs) is in the presence of a second function $\nu_A(x)$ defining the non-membership of the element x to the set A , where $\mu_A(x) \in [0; 1]$, $\nu_A(x) \in [0; 1]$, under the condition

$$\mu_A(x) + \nu_A(x) \in [0; 1].$$

The IFS itself is formally denoted by:

$$A = \{(x, \mu_A(x), \nu_A(x)) | x \in E\}.$$

With IFS we are going to evaluate eventual failures in the smart system.

We need (IFS), in order to evaluate the possible failures of the system.

The estimations are presented by ordered pairs $\langle \mu, \nu \rangle$ of real numbers from set $[0, 1]$, where:

$$\mu = \frac{S_1}{S},$$

where

S all the possible info packets taking place in the smart system.

S_1 the info packets that successfully reach their destination.

$$v = \frac{S_2}{S},$$

where

S_2 the info packets that for unknown reason do not reach their destination.

The degree of uncertainty $\pi = 1 - \mu - v$ is all of the info packets in the smart system that enter place successfully and all the possible failure packets.

2 GN Model

The automatized lighting systems facilitate the users especially the disabled people.

Initially the following tokens enter in the generalized net:

For facilitation we separate the following types of tokens:

- α Person $i \in [1, n]$
- β Database
- γ Identifier
- μ Current condition.

The GN model of atomized lightening system (Fig. 1) is introduced by the set of transitions:

$$A = \{Z_1, Z_2, Z_3, Z_4, Z_5\},$$

where the transitions describe the following process:

- Z_1 = “Verification of the iris”;
- Z_2 = “Lighting the room”;
- Z_3 = “Corrections”;
- Z_4 = “Assignment of the parameters of the level of lightening”;
- Z_5 = “IFS evaluation of a possible failures of info packets in the system”;

$$Z_1 = \langle \{L_1, L_2\}, \{L_2, L_3, L_4, L_{exit}, L_{16}\}, R_1, \wedge(L_1, L_2) \rangle$$

$$R_1 = \begin{array}{c|ccccc} & L_2 & L_3 & L_4 & L_{exit} & L_{16} \\ L_1 & true & false & false & false & true \\ L_2 & true & W_{2,3} & W_{2,4} & W_{2,exit} & true \end{array}$$

where

$W_{2,3}$ “The Iris is been recognized”

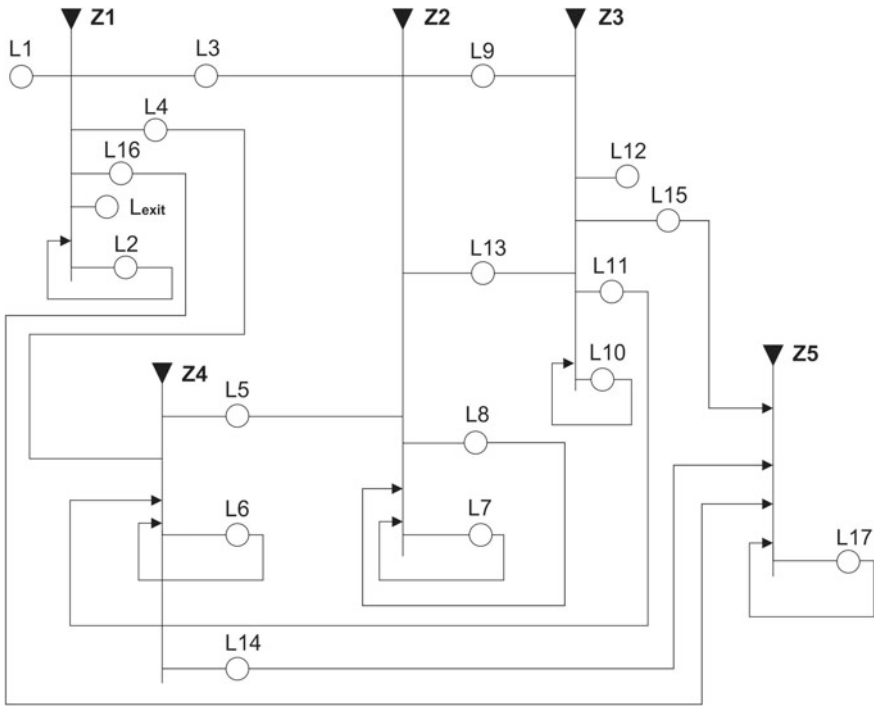


Fig. 1 GN model of the automated lighting system

$$\begin{aligned}
 W_{2,exit} &= \neg W_{2,3} \\
 W_{2,4} &= W_{2,3}.
 \end{aligned}$$

The β_1 token that enters place L_2 obtains the characteristic “Iris in Database”.

The α_1 token that enters place L_3 obtains the characteristic “Recognized person”.

The γ token that enters place L_4 obtains the characteristic “Identifier of a person”.

The γ_1 token that enters place L_{exit} obtains the characteristic “Unidentified person”.

$$Z_2 = \langle \{L_3, L_5, L_7, L_8\}, \{L_8, L_9, L_7, L_{13}\}, R_2, \wedge(L_3, L_5, L_7, L_8) \rangle$$

	L_9	L_8	L_7	L_{13}	
L_3	<i>true</i>	<i>false</i>	<i>false</i>	<i>false</i>	
L_5	<i>false</i>	<i>false</i>	<i>true</i>	<i>true</i>	,
L_7	<i>false</i>	<i>true</i>	<i>true</i>	<i>true</i>	
L_8	<i>false</i>	<i>true</i>	<i>true</i>	<i>false</i>	

The token from L_5 transfers to L_{13} and doesn't obtain new characteristics.

The token from L_3 transfers to L_9 and doesn't obtain new characteristics.

The α_1 token that enters place L_3 obtains the characteristic “Recognized person”.

The α_2 token that enters place L_5 obtains the characteristic “Personal settings of a person for lighting”.

The μ_1 token that enters place L_8 obtains the characteristic “Threshold value”.

$$Z_3 = (\{L_9, L_{13}, L_{10}\}, \{L_{12}, L_{11}, L_{10}, L_{15}\}, R_3, \vee(L_{10}, \wedge(L_9, L_{13})))$$

$$R_3 = \begin{array}{c|cccc} & L_{12} & L_{11} & L_{10} & L_{15} \\ \hline L_9 & true & false & false & true \\ L_{13} & false & true & true & true \\ L_{10} & false & W_{10,11} & true & true \end{array},$$

where

$W_{10,11}$ “There are corrections”.

The token of L_9 transfers to L_{12} and it doesn't obtain new characteristics.

The μ_2 token that enters place L_{10} obtains the characteristic “Current condition of the settings”.

The μ_3 token that enters place L_{11} obtains the characteristic “Corrections of the personal options of a person”.

$$Z_4 = (\{L_4, L_6, L_{11}\}, \{L_5, L_6, L_{14}\}, R_4, \vee(L_4, L_6, L_{11}))$$

$$R_4 = \begin{array}{c|ccc} & L_5 & L_6 & L_{14} \\ \hline L_4 & false & true & true \\ L_6 & W_{6,5} & true & true \\ L_{11} & false & true & true \end{array},$$

where

$W_{6,5}$ There are found settings for the lighting.

The β_2 token that enters place L_6 obtains the characteristic “Database with personal options for lighting”.

The γ token that enters place L_4 obtains the characteristic “Identifier of a person”.

$$Z_5 = (\{L_{14}, L_{15}, L_{16}, L_{17}\}, \{L_{17}\}, R_5, \vee(L_{14}, L_{15}, L_{16}, L_{17}))$$

$$R_5 = \begin{array}{c|c} & L_{17} \\ \hline L_{14} & true \\ L_{15} & true \\ L_{16} & true \\ L_{17} & true \end{array},$$

where

The token that enters place L_{16} obtains the characteristic “Unrecognized Iris - IFS estimate unknown”.

The token that enters place L_{14} obtains the characteristic “Unrecognized personal settings - IFS estimate unknown”.

The token that enters place L_{15} obtains the characteristic “Unrecognized corrections - IFS estimate unknown”.

The token that enters place L_{17} obtains the characteristic “IFS estimations $\langle \mu_k, \nu_k \rangle$ ”.

Initially when no information has been derived from places L_{16} , L_{14} , L_{15} L_{17} , all estimates take initial values of $\langle 0, 0 \rangle$.

When ≥ 0 , the current $(k + 1)$ st estimation is calculated on the basis of the previous estimations according to the recursive formula (as before):

$$\langle \mu_{k+1}, \nu_{k+1} \rangle = \frac{\mu_k k + \mu}{k + 1}, \frac{\nu_k k + \nu}{k + 1},$$

where $\langle \mu_k, \nu_k \rangle$ is the previous estimation, and $\langle \mu, \nu \rangle$ is the latest estimation of the smart system information packets flow, for $\mu, \nu \in [0, 1]$ and $\mu + \nu \leq 1$. Thus the token in place L_{17} forms the final estimation of the information packets in smart system on the basis of previous and the latest events.

3 Conclusion

The system for automatic starting of the lighting is used for facilitation of the user and especially the disabled ones. It is in the beginning of the process of the smart house and it provides the lighting of the room and helps for all other following processes. The model is presented with its generalized network and shows the processes of which the person goes through the system also possible mistakes which can be inflicted. The GN model helps us analyze the possible problems or to simulate other problems so that we can optimize the behavior of the system.

Acknowledgements This work was supported by the Bulgarian Ministry of Education and Science under the National Research Programme “Information and Communication Technologies for a Digital Single Market in Science, Education and Security” approved by DCM # 577/17.08.2018.

References

1. Atanassov, K., Aladjov, H.: Generalized Nets in Artificial Intelligence: Volume 2: Generalized Nets and Machine Learning. “Prof. Marin Drinov” Publishing House, Sofia (2000)
2. Atanassov, K.: Generalized nets as tools for modelling of intelligent systems. In: 9th IWGN, Sofia, 4 July 2008

3. Atanassov, K.: Index matrices with elements index matrices. In: Proc. Jangjeon Math. Soc. **21**(2), 221–228 (2018)
4. Atanassov, K.: Intuitionistic fuzzy sets and interval valued intuitionistic fuzzy sets. Adv. Stud. Contemp. Math. **28**(2), 167–176 (2018). Atanassov, K.: n-Dimensional extended index matrices. Adv. Stud. Contemp. Math. **28**(2), pp. 245–259 (2018)
5. Atanassov, K., Sotirova, E., Bureva, V.: On index matrices. Part 4: new operations over index matrices. Adv. Stud. Contemp. Math. **23**(3), 547–552 (2013)
6. Bureva, V., Sotirova, E., Bozov, H.: Generalized net model of biometric identification process. In: Proceedings of the 20th International Symposium on Electrical Apparatus and Technologies (SIELA), pp. 1–4 (2018)
7. Wikipedia, wikipedia.org—Threshold model, 30 May 2018
8. Atanassov, K., Sotirova, E., Andonov, V.: Generalized net model of multicriteria decision making procedure using InterCriteria analysis. In: Advances in Fuzzy Logic and Technology 2017, Advances in Intelligent Systems and Computing, vol. 641, pp. 99–111 (2018)
9. Bureva, V., Sotirova, E., Popov, S., Mavrov, D., Traneva, V.: Generalized net of cluster analysis process using STING: a statistical information grid approach to spatial data mining. In: International Conference on Flexible Query Answering Systems, Lecture Notes in Computer Science, vol. 10333, pp. 239–248 (2017)
10. Bureva, V., Yovcheva, Pl., Sotirov, S.: Generalized net model of fingerprint recognition with intuitionistic fuzzy evaluations. In: Advances in Fuzzy Logic and Technology 2017, Advances in Intelligent Systems and Computing, vol. 641, pp. 286–294 (2018)
11. Bureva, V., Popov, S., Sotirova, E., Miteva, B.: generalized net of the process of hierarchical cluster analysis. Ann. Assen Zlatarov Univ. Burgas Bulgaria **XLVI**(1), 107–111 (2017)
12. Bureva, V., Sotirova, E., Chountas, P.: Generalized net of the process of sequential pattern mining by generalized sequential pattern algorithm (GSP). Adv. Intell. Syst. Comput. **323**, 831–838 (2015)
13. Judd, D.B., MacAdam, D.L., Wyszecki, G., Budde, H.W., Condit, H.R., Henderson, S.T., Simonds, J.L.: Spectral distribution of typical daylight as a function of correlated color temperature. J. Opt. Soc. Am.
14. Petkov, T., Jovcheva, Pl., Tomov, Z., Simeonov, S., Sotirov, S.: A Generalized net model of the neocognitron neural network. In: International Conference on Flexible Query Answering Systems, Lecture Notes in Computer Science, vol. 10333, pp. 249–259 (2017)
15. Torsten, S., Stoica, P.: System Identification. Prentice Hall, New York. ISBN 978-0138812362 (1989)
16. Andreev, N., Vassilev, P., Atanassova, V., Roeva, O., Atanassov, K.: Generalized net model of the cooperation between the Departments of transfusion Haematology and the National Centre of transfusion Haematology. In: Advances in Neural Networks and Applications 2018 (ANNA'18), pp. 1–4, St. Konstantin and Elena Resort, Bulgaria, 15–17 Sept 2018
17. Roeva, O.: Application of artificial bee colony algorithm for model parameter identification. In: Zelinka, I., Vasant, P., Duy, V., Dao, T., et al. (eds.) Innovative Computing, Optimization and Its Applications, Studies in Computational Intelligence, vol. 741, pp. 285–303. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-66984-7_17. Book ISBN: 978-3-319-66983-0
18. Sotirova, E.N., Dimitrov, D.D., Atanassov, K.T.: On some applications of game method for modelling Part I: Forest dynamics. Proc. Jangjeon Math. Soc. **15**(2), 115–123 (2012)
19. Ribagin, S., Chountas, P., Pencheva, T.: Generalized Net Model of Muscle Pain Diagnosing. Lecture Notes in Artificial Intelligence, vol. 10333, pp. 269–275 (2017). ISSN 0302-9743, ISSN 1611-3349 (electronic), ISBN 978-3-319-59692-1
20. Angelova, M., Roeva, O., Pencheva, T.: Artificial bee colony algorithm for parameter identification of fermentation process model. In: Submitted to 3rd International Conference on: Applied Physics, System Science and Computers. Lecture Notes in Electrical Engineering, Dubrovnik, Croatia, 26–28 Sept 2018 (in press)
21. Fidanova, S., Roeva, O.: InterCriteria Analysis of Different Variants of ACO algorithm for Wireless Sensor Network Positioning, Lecture Notes in Computer Science (in press)

22. Gocheva, E., Sotirov, S.: Modelling of the verification by iris scanning by generalized nets. In: 9th IWGN, Sofia, 4 July 2008
23. Shannon, A., Sotirova, E., Hristova, M., Kim, T.: Generalized net model of a student's course evaluation with intuitionistic fuzzy estimations in a digital university. *Proc. Jangjeon Math. Soc.* **13**(1), 31–38 (2010)
24. Zoteva, D., Atanassova, V., Roeva, O., Szmidi, E.: Generalized net model of artificial bee colony optimization algorithm. In: *Advances in Neural Networks and Applications 2018 (ANNA'18)*, pp. 1–4, St. St. Konstantin and Elena Resort, Bulgaria, 15–17 Sept 2018
25. Kashtan, N., Noor, E., Alon, U.: Varying environments can speed up evolution. *Proc. Natl. Acad. Sci. USA* **104**(13), 13711–13716
26. Gioia, D.A., Patvardhan, S.: *Identity as Process and Flow*. Published to Oxford Scholarship (2012). <https://doi.org/10.1093/acprofoso/9780199640997.003.0003>
27. Zetter, K.: Reverse-engineered irises look so real, they fool eye-scanners. *Wired Magazine*. Retrieved 25 July 2012
28. Peneva, D., Tasseva, V., Kodogiannis, V., Sotirova, E., Atanassov, K.: Generalized nets as an instrument for description of the process of expert system construction. *IEEE Intelligent Systems*, Art. no. 4155522, pp. 755–759
29. Jaspal, R., Breakwell, G.: *Identity Process Theory Identity, Social Action and Social Change*. Cambridge University Press, Cambridge. ISBN 978–1-107-02270-6 (2014)
30. *Northeast Document Conservation Center Revised 2012. Protection from Light Damage*
31. Fidanova, S., Roeva, O.: Influence of Ant Colony Optimization Parameters on the Algorithm Performance, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10665, pp. 358–365 (2018)
32. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **20**(1)
33. Atanassov, K.: *Intuitionistic Fuzzy Sets: Theory and Applications*. Physica-Verlag, Heidelberg (1999)
34. Atanassov, K.: *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin (2012)
35. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**, 333–353 (1965)

Generalized Net Model of Common Internet Payment Gateway with Intuitionistic Fuzzy Estimations



Boris Bozveliev, Sotir Sotirov, Stanislav Simeonov and Tihomir Videv

Abstract A payment gateway is a service that authorizes and processes debit/credit card or other payments for online merchants it facilitates these transactions by encrypting sensitive data and transferring it between a payment portal (a website or a mobile device) and the bank/front end processor. In this paper we examine the transaction process, even though it takes only a few seconds, several steps are accomplished during that brief window of time. Here we present a generalized net model with intuitionistic fuzzy estimations of such an internet payment gateway.

Keywords Generalized nets · Payment gateway · Credit card · Merchant account · Merchandise · Authorization

Mathematics Subject Classification (2010) 68Q85 · 68M11

1 Introduction

The current paper examines Generalized Net (GN) [1, 2] model of a common payment gateway since there many payment systems [3], we are going to use a common one and will show all the parallel processes that go through such system.

Here we have to mention that all the processes (transactions) are encrypted [4, 5] for security reasons. The security [4, 6, 7] is the key factor for flawless work, the GN [8] model helps us easily analyze the payment gateway and solve possible gaps in the systems. Nowadays customers and merchants are forced to use multiple systems, mobiles or website for payments to pay for goods and the payment gateways are these that interconnect [9] these systems. The way that payment systems work [10],

B. Bozveliev (✉) · S. Sotirov · S. Simeonov · T. Videv
“Prof. Dr, Assen Zlatarov” University, “Prof. Yakimov” Blvd, Burgas 8010, Bulgaria

S. Simeonov
e-mail: st_sim@yahoo.com

T. Videv
e-mail: tvidev@abv.bg

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_8

is explained in a few steps in order to simplify the process these are: The customer must have a “user account” created with a personal data in order to be able to make purchases through the website or a mobile phone. He will also need a valid credit or debit card. Payment details are encrypted [4, 5] via the electronic payment system. Then there is a check of the user account database, if ok it continues to check the cards database. There is an initial merchant account validation too, if ok, the process continues to present the customer a choice of the merchandise or the service that the customer wants to buy. Here we can put a note that at all the different stages if there is problem with any of the described processes the system halts and exits, and the process can start from the begging. The next hop after success is the credit card balance check followed by the merchant account check, if a success the funds are withdrawn from the card. The goods then are send to the customer and a confirmation of successful transaction is being sent to the person responsible for the transaction. In this paper we are generally looking at a common payment model [10, 11] The GN model [1, 2] will help us easily and clearly understand the main mode of operation stages of the gateway so then we may be able to improve security, troubleshoot and analyze better.

In the above context we are going to evaluate the possibilities of unpredicted failure of the payment system. Here we are going to use fuzzy sets (IFS). The Intuitionistic Fuzzy Sets (IFSs, see [12–15]) represent an extension of the concept of fuzzy sets, as defined by Zadeh [16], exhibiting function $\mu_A(x)$ defining the membership of an element x to the set A , evaluated in the $[0; 1]$ - interval. The difference between fuzzy sets and Intuitionistic Fuzzy Sets (IFSs) is in the presence of a second function $\nu_A(x)$ defining the non-membership of the element x to the set A , where $\mu_A(x) \in [0; 1]$, $\nu_A(x) \in [0; 1]$, under the condition

$$\mu_A(x) + \nu_A(x) \in [0; 1].$$

The IFS itself is formally denoted by:

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in E \}$$

We need (IFS), in order to evaluate the possible failures of the system.

The estimations are presented by ordered pairs $\langle \mu, \nu \rangle$ of real numbers from set $[0, 1]$, where:

$$\mu = \frac{S_1}{S},$$

where:

- S All the possible packets of information taking place in the system.
- S_1 The packets of information that successfully reach their destination

$$v = \frac{S_2}{S},$$

where:

S_2 The packets of information that for unknown reason do not reach their destination.

The degree of uncertainty $\pi = 1 - \mu - v$ is all the packets of information in the payment system that enter place successfully and all the possible unsuccessful entries.

2 Common Internet Payment Gateway

Ultimately, payment gateways facilitate communication between a website or a brick and mortar stores, the payment processor and the bank that issued the credit card being used to complete the purchase.

Initially the following tokens enter in the generalized net:

- in place L_1-A —token with characteristic “Customer/visitor”;
- in place L_2-B —token with characteristic “Credit Cards Database Info”;
- in place L_3-B_1 —token with characteristic “User Account Database Info”;
- in place L_4-B_2 —token with characteristic “Merchant account Database Info”;
- in place L_6-A_1 —token with characteristic “Customer/visitor”;
- in place $L_7-\Theta$ —token with characteristic “Merchandise/services”;
- in place L_8-A_1 —token with characteristic “Customer/visitor”;
- in place L_9-B_3 —token with characteristic “Account balance”;
- in place $L_{10}-B_4$ —token with characteristic “Merchant account verify”;
- in place $L_{11}-P$ —token with characteristic “Answer to” “ L_4 ”.
- in place $L_{13}-A_2$ —token with characteristic “Customer/visitor”.
- in place $L_{14}-A$ —token with characteristic “Credit/Debit card funds withdrawal”.
- in place $L_{15}-A_3$ —token with characteristic “Customer/visitor”.
- in place $L_{17}-G_1$ —token with characteristic “Confirmation delivery”;
- in place $L_{19}-\Omega_1$ —token with characteristic “Possible unsuccessful authorization IFS estimate unknown”;
- in place $L_{20}-\Omega_2$ —token with characteristic “Possible unsuccessful payment IFS estimate unknown”;
- in place $L_{21}-\Omega_3$ —token with characteristic “Possible unsuccessful delivery IFS estimate unknown”;
- in place $L_{22}-\Omega_4$ —token with characteristic “IFS estimations $\langle \mu_k, \nu_k \rangle$ ”;

3 GN Model

The GN model of common payment gateway (Fig. 1) is introduced by the set of transitions:

$$A = \{Z_1, Z_2, Z_3, Z_4, Z_5, Z_6\},$$

where the transitions describe the following processes:

- Z_1 = “Initial authorization”;
- Z_2 = “Choice of merchandise/services”;
- Z_3 = “Payment processing”;
- Z_4 = “Holding funds”;
- Z_5 = “Delivering of goods & Confirmation Services”;
- Z_6 = “Evaluation of possible failure of information packets in the system”.

$$Z_1 = \{ \{L_1, L_{11}, L_2, L_3\} \{L_6, L_3, L_2, L_4, L_5, L_{19}\}, R_1, \vee(L_1, L_{11}, L_2, L_3) \}$$

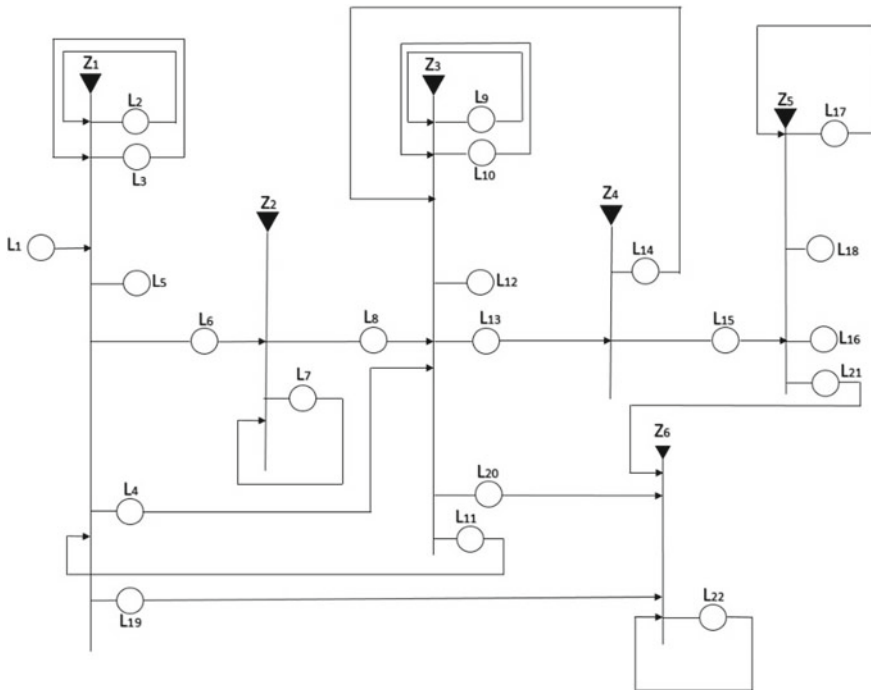


Fig. 1 GN model of common payment gateway

	L_2	L_3	L_4	L_5	L_6	L_{19}
$R_1 = L_1$	<i>true</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
L_2	<i>true</i>	$W_{2,3}$	<i>false</i>	$W_{2,5}$	$W_{2,6}$	<i>true</i>
L_3	<i>false</i>	<i>true</i>	<i>false</i>	$W_{3,5}$	$W_{3,6}$	<i>true</i>
L_{11}	<i>false</i>	<i>false</i>	$W_{11,4}$	<i>false</i>	<i>false</i>	<i>true</i>

where:

$W_{2,3} =$ “There is a customer account”;

$W_{2,5} = \neg W_{2,3}$;

$W_{2,6} =$ “ L_2, L_3, L_4 have been successfully completed”;

$W_{3,5} =$ “A problem with the debit/credit card”;

$W_{3,6} = W_{2,6}$;

$W_{11,4} =$ “Merchant account verified”.

The token that enters place L_2 obtains the characteristic “Credit Cards Database info”.

The token that enters place L_3 obtains the characteristic “User Accounts Database info”.

The token that enters place L_4 obtains the characteristic “Merchant account Database info”.

The token that enters place L_5 obtains the characteristic “Error Exit”.

The token that enters place L_6 obtains the characteristic “Customer/visitor”.

$$Z_2 = \langle \{L_6, L_7\} \{L_7, L_8\}, R_2, \wedge(L_6, L_7) \rangle$$

	L_7	L_8
$R_2 = L_6$	<i>true</i>	<i>false</i>
L_7	<i>true</i>	<i>true</i>

The token that enters place L_7 obtains the characteristic “merchandise/services”.

The token that enters place L_8 obtains the characteristic “customer/visitor”.

$$Z_3 = \langle \{L_8, L_9, L_{10}, L_4, L_{14}\} \{L_{12}, L_9, L_{10}, L_{13}, L_{20}\}, R_3, \vee(L_8, L_9, L_{10}, L_4, L_{14}) \rangle$$

	L_{12}	L_9	L_{10}	L_{13}	L_{20}
$R_3 = L_8$	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>
L_9	$W_{9,12}$	<i>true</i>	$W_{9,10}$	<i>false</i>	<i>true</i>
L_{10}	$W_{10,12}$	<i>false</i>	<i>true</i>	$W_{10,13}$	<i>true</i>
L_4	<i>false</i>	<i>false</i>	<i>true</i>	<i>false</i>	<i>true</i>
L_{14}	<i>false</i>	<i>true</i>	<i>false</i>	<i>false</i>	<i>true</i>

where:

$W_{9,12} =$ “Insufficient Funds”;

$W_{9,10} = \neg W_{9,12}$;

$W_{10,12} =$ “There is a problem with the merchant’s account”;

$W_{10,13} = \neg W_{10,12}$.

The token that enters place L_9 obtains the characteristic “Account balance”.

The token that enters place L_{10} obtains the characteristic “Merchant account database info”.

The token that enters place L_{11} obtains the characteristic “Answer to” “ L_4 ”.

The token that enters place L_{12} obtains the characteristic “Error Exit”.

The token that enters place L_{13} obtains the characteristic “Customer/visitor”.

$$Z_4 = (\{L_{13}\}\{L_{14}, L_{15}\}, R_4, \wedge(L_{13}))$$

$$R_4 = \frac{L_{14} \quad L_{15}}{L_{13} \mid \text{true true}}$$

The token that enters place L_{14} obtains the characteristic “Credit/Debit card funds withdrawal”.

The token that enters place L_{15} obtains the characteristic “Customer/visitor”.

$$Z_5 = (\{L_{15}, L_{17}\}\{L_{16}, L_{17}, L_{18}, L_{21}\}, R_5, \vee(L_{15}, L_{17}))$$

$$R_5 = \frac{L_{16} \quad L_{17} \quad L_{18} \quad L_{21}}{L_{13} \mid \text{true true false true}, \\ L_{17} \mid \text{false true true true}}$$

The token that enters place L_{16} obtains the characteristic “Merchandise/services delivery”.

The token that enters place L_{17} obtains the characteristic “Confirmation delivery”.

The token that enters place L_{18} obtains the characteristic “Exit”.

$$Z_6 = (\{L_{19}, L_{20}, L_{21}L_{22}\}\{L_{22}\}, R_6, \vee(L_{19}, L_{20}, L_{21}L_{22}))$$

$$R_5 = \begin{array}{c|c} & L_{22} \\ \hline L_{19} & true \\ L_{20} & true, \\ L_{21} & true \\ L_{22} & true \end{array}$$

where:

- The token that enters place L_{19} obtains the characteristic “IF estimation”.
- The token that enters place L_{20} obtains the characteristic “IF estimation”.
- The token that enters place L_{21} obtains the characteristic “IF estimation”.
- The token that enters place L_{22} obtains the characteristic “IF estimation $\langle \mu_k, \nu_k \rangle$ ”.

Initially when no information has been derived from places $L_{19}, L_{20}, L_{21}, L_{22}$, all estimates take initial values of $\langle 0, 0 \rangle$.

When ≥ 0 , the current $(k + 1) - st$ estimation is calculated on the basis of the previous estimations according to the recursive formula (as before):

$$\langle \mu_{k+1}, \nu_{k+1} \rangle = \left\langle \frac{\mu_k k + \mu}{k + 1}, \frac{\nu_k k + \nu}{k + 1} \right\rangle,$$

where $\langle \mu_k, \nu_k \rangle$ is the previous estimation, and $\langle \mu, \nu \rangle$ is the latest estimation of the payment procedure information packets, for $\mu, \nu \in [0, 1]$ and $\mu + \nu \leq 1$. This way the token in place L_{22} forms the final estimation of the info packets in the payment gateway on the basis of previous and the latest events.

4 Conclusion

The payment gateway is used for facilitating online transactions and lets them get approved. It is also the first place the transaction goes when a customer submits an order online. The model is presented with generalized net and shows the transaction flows through the payment gateway, and through different checks and approvals and possible declines. So the GN model helps us to look further in the gateway, and correct possible problems or simulate other problems or just use it for optimization of the behavior of the payment gateway.

Acknowledgements The authors acknowledged support (financial, computational, logistic, etc.) from the project UNITe BG05M2OP001-1.001-0004 /28.02.2018 (2018–2023).

References

1. Atanassov, K.: Generalized Nets. World Scientific, Singapore, New Jersey, London (1991)

2. Atanassov, K., Aladjov, H.: Generalized Nets in Artificial Intelligence. Prof. Marin Drinov Publishing House, Sofia (1998)
3. Best Payment Gateways | Formstack. Available from: Crossref. Date accessed 05 Mar 2017
4. Credit Card Tokenization 101—And Why it's Better than Encryption. www.3dsi.com. Retrieved 30 Mar 2016
5. P2PE: Point to Point Encryption for PCI Compliant Payments. Bluefin Payment Systems. Retrieved 30 Mar 2016
6. E-Commerce Security Systems. Available from: Crossref. Date accessed 20 February 2017
7. Security Features of Payment Gateway Information Technology, Essay. Available from: Crossref. Date accessed 20 February 2017
8. Nikolova, M., Szmids, E., Hadjitodorov, S.: Generalized nets with decision making components. In: Proceedings of International Workshop on Generalized Nets, Sofia, pp. 1–5, 9 July 2000
9. Quartel, D., Pras, A.: An Interconnection Architecture for Micropayment Systems. University of Twente, Enschede, The Netherlands (2005)
10. Hord, J.: How Electronic Payment Works. Available from: Crossref. Date accessed 23 Feb 2017
11. Choosing your payment methods. Digitalbusiness.gov.au. Retrieved 19 Nov 2012
12. Atanassov, K.: Intuitionistic fuzzy sets. In: Proceedings of VIIITKR's Session, Sofia, Bulgarian (1983)
13. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **20**(1), 87–96 (1986)
14. Atanassov, K.: Intuitionistic Fuzzy Sets: Theory and Applications. Physica, Heidelberg (1999)
15. Atanassov, K.: On Intuitionistic Fuzzy Sets Theory. Springer, Berlin (2012)
16. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**, 333–353 (1965)

Intuitionistic Fuzzy Neural Networks with Interval Valued Intuitionistic Fuzzy Conditions



Krassimir Atanassov, Sotir Sotirov and Nora Angelova

Abstract This paper describes a new approach for Intuitionistic fuzzy neural networks with interval valued intuitionistic fuzzy conditions. The theoretical basis for the new approach is presented, and well as examples to illustrate the proposed concepts and ideas. This paper can be viewed as the initial foundation of the area of intuitionistic fuzzy neural networks, and in future papers more results will be presented.

Keywords Intuitionistic fuzzy logic · Neural networks · Interval valued fuzzy systems

1 Introduction

During last 25 years some research over the concept of an intuitionistic fuzzy neural network were published – see [10, 13–15, 17] and some other tools, related to intuitionistic fuzziness, e.g., the intercriteria analysis were used for estimation of the results of neural network functioning (see [18, 19]).

In [10, 17] intuitionistic fuzzy feedforward network was constructed by combining feed forward neural networks and intuitionistic fuzzy logic. Some operations and two

K. Atanassov

Department of Bioinformatics and Mathematical Modelling Institute of Biophysics and Biomedical Engineering, Bulgarian Academy of Sciences, 105 Acad. G. Bonchev Str., 1113 Sofia, Bulgaria
e-mail: krat@bas.bg

K. Atanassov · S. Sotirov (✉)

Intelligent systems laboratory, Assen Zlatarov University, 8010 Burgas, Bulgaria
e-mail: ssotirov@btu.bg

N. Angelova

Department of Computer Informatics Faculty of Mathematics and Informatics, Sofia University, 5, James Bourchier Blvd., 1164 Sofia, Bulgaria
e-mail: nora.angelova@fmi.uni-sofia.bg

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_9

types of transferring functions involved in the working process of these nets were introduced.

There are a few papers [13, 16] that combine ideas from intuitionistic fuzziness and artificial neural networks. In [13] is introduced an Intuitionistic fuzzy RBF network. In [16] is produced Intuitionistic fuzzy model of a neural network.

Here, an extension of the concept of a neural network will be introduced. It will be based of the apparatus of the Intuitionistic Fuzzy Sets (IFSs; see [5, 6]), Interval Valued IFSs (IVIFSs; see [5]), Intuitionistic Fuzzy Pairs (IFPs; see [11]) and Interval Valued Intuitionistic Fuzzy Pairs (IVIFPs; see [12]). The previous and the present research are related to the idea from [8] that intuitionistic fuzziness can be used for estimation of neural network parameters.

In Sect. 2, we give some preliminary remarks and in Sect. 3 we introduced the idea for the new type of neural networks.

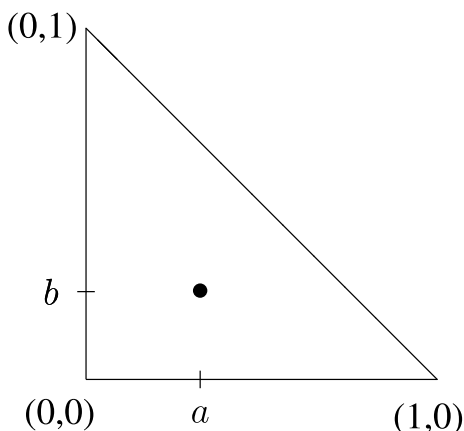
2 Preliminary Remarks

In [11, 12], formal definitions of IFP and IVIFP are given and definitions of some operations, relations and operators over them are defined.

The IFP and IVIFP are objects with the form $\langle \mu, \nu \rangle$ and $\langle M, N \rangle$, respectively, where $\mu, \nu \in [0, 1]$ and $\mu + \nu \leq 1$, $M, N \subseteq [0, 1]$ are closed sets, $M = [\inf M, \sup M]$, $N = [\inf N, \sup N]$ and $\sup M + \sup N \leq 1$. Their components (μ, ν and M, N , respectively) are interpreted as degrees of membership and non-membership, or degrees of validity and non-validity, or degree of correctness and non-correctness, etc. Geometrical interpretations of an IFP and of an IVIFP are shown on Figs. 1 and 2.

In some definitions below, we use functions sg and \overline{sg} defined by,

Fig. 1 Geometrical interpretation of an IFP



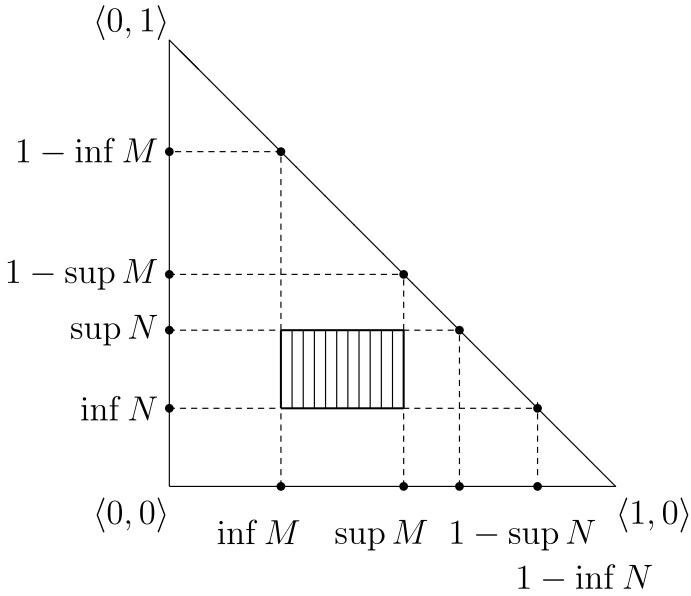


Fig. 2 Geometrical interpretation of an IVIFP

$$sg(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases},$$

$$\overline{sg}(x) = \begin{cases} 0 & \text{if } x > 0 \\ 1 & \text{if } x \leq 0 \end{cases}$$

Let us have two IFPs $x = \langle a, b \rangle$ and $y = \langle c, d \rangle$.

First, we define analogous of operations “conjunction” and “disjunction”:

$$\begin{aligned} x \&_1 y &= x \cap y = \langle \min(a, c), \max(b, d) \rangle, \\ x \vee_1 y &= x \cup y = \langle \max(a, c), \min(b, d) \rangle, \\ x \vee_2 y &= x + y = \langle a + c - a.c, b.d \rangle, \\ x \&_2 y &= x . y = \langle a.c, b + d - b.d \rangle, \\ x \vee_3 y &= x \&_3 y = x @ y = \langle \frac{a+c}{2}, \frac{b+d}{2} \rangle. \end{aligned}$$

While for the first four cases $(x \circ y) \circ z = x \circ (y \circ z)$, where \circ is one of the first four operations and therefore we can define

$$\begin{aligned} \circ(x, y) &= x \circ y, \\ \bigcirc_{i=1}^n x_i &= \circ(x_1, \dots, x_n, x_{n+1}) = \circ(\circ(x_1, \dots, x_n), x_{n+1}), \end{aligned}$$

for the fifth case this is not valid. So, we must define:

$$@ (x, y) = x @ y,$$

for $n = 2$, but for $n \geq 3$

$$\bigcirc_{i=1}^n x_i = @ (x_1, \dots, x_n) = \left\langle \frac{a_1 + \dots + a_n}{n}, \frac{b_1 + \dots + b_n}{n} \right\rangle.$$

In [9], 185 implications and 53 negations over IFPs are defined.

There, three types of operators over IFPs are defined. Below, we give only a part for them.

Let as above, $x = \langle a, b \rangle$ be an IFP and let $\alpha, \beta \in [0, 1]$. Then the modal type of operators defined over x have the forms:

$$\begin{aligned} \square x &= \langle a, 1 - a \rangle \\ \diamond x &= \langle 1 - b, b \rangle \\ D_\alpha(x) &= \langle a + \alpha.(1 - a - b), b + (1 - \alpha).(1 - a - b) \rangle \\ F_{\alpha, 1-\alpha}(x) &= \langle a + \alpha.(1 - a - b), b + \beta.(1 - a - b) \rangle, \text{ where } \alpha + \beta \leq 1 \\ G_{\alpha, \beta}(x) &= \langle \alpha.a, \beta.b \rangle \\ H_{\alpha, \beta}(x) &= \langle \alpha.a, b + \beta.(1 - a - b) \rangle \\ H_{\alpha, \beta}^*(x) &= \langle \alpha.a, b + \beta.(1 - \alpha.a - b) \rangle \\ J_{\alpha, \beta}(x) &= \langle a + \alpha.(1 - a - b), \beta.b \rangle \\ J_{\alpha, \beta}^*(x) &= \langle a + \alpha.(1 - a - \beta.b), \beta.b \rangle \end{aligned}$$

$$X_{\alpha,\beta,\gamma,\delta,\varepsilon,\eta}(x) = \langle \alpha.a + \beta.(1 - a - \gamma.b), \delta.b + \varepsilon.(1 - \eta.a - b) \rangle$$

$$\boxplus x = \langle \frac{a}{2}, \frac{b+1}{2} \rangle$$

$$\boxtimes x = \langle \frac{a+1}{2}, \frac{b}{2} \rangle$$

$$\boxplus_{\alpha} x = \langle \alpha.a, \alpha.b + 1 - \alpha \rangle$$

$$\boxtimes_{\alpha} x = \langle \alpha.a + 1 - \alpha, \alpha.b \rangle$$

$$\boxplus_{\alpha,\beta} x = \langle \alpha.a, \alpha.b + \beta \rangle, \text{ where } \alpha + \beta \leq 1$$

$$\boxtimes_{\alpha,\beta} x = \langle \alpha.a + \beta, \alpha.b \rangle, \text{ where } \alpha + \beta \leq 1$$

$$\boxplus_{\alpha,\beta,\gamma} x = \langle \alpha.a, \beta.b + \gamma \rangle, \text{ where } \max(\alpha, \beta) + \gamma \leq 1$$

$$\boxtimes_{\alpha,\beta,\gamma} x = \langle \alpha.a + \gamma, \beta.b \rangle, \text{ where } \max(\alpha, \beta) + \gamma \leq 1$$

$$\bullet_{\alpha,\beta,\gamma,\delta} x = \langle \alpha.a + \gamma, \beta.b + \delta \rangle, \text{ where } \max(\alpha, \beta) + \gamma + \delta \leq 1$$

$$E_{\alpha,\beta} x = \langle \beta(\alpha.a + 1 - \alpha), \alpha(\beta.b + 1 - \beta) \rangle \text{ (see [11])}$$

$$\boxplus_{\alpha,\beta,\gamma,\delta,\varepsilon,\zeta} x = \langle \alpha.a - \varepsilon.b + \gamma, \beta.b - \zeta.a + \delta \rangle, \text{ where } \max(\alpha - \zeta, \beta - \varepsilon) + \gamma + \delta \leq 1$$

$$\text{and } \min(\alpha - \zeta, \beta - \varepsilon) + \gamma + \delta \geq 0$$

3 Definition of an Intuitionistic Fuzzy Neural Networks with Interval Valued Intuitionistic Fuzzy Conditions

Let us have a Neural Network with s layers. Let the neurons in i -th layer be denoted by $V_{i,j}$, where $0 \leq i \leq s$, $1 \leq j \leq w_i$, where w_i is the number of neurons in i -th layer. Let to each ark connecting neurons $V_{i,j}$ and $V_{i+1,k}$, where $1 \leq k \leq w_{i+1}$, the pair of pairs $\langle \langle M_{i,j,k}, N_{i,j,k} \rangle, \langle \varphi_{i,j}, \psi_{i,j} \rangle \rangle$ be juxtaposed, where $M_{i,j,k}, N_{i,j,k} \subseteq [0, 1]$ be intervals, so that $M_{i,j,k} = [\inf M_{i,j,k}, \sup M_{i,j,k}]$, $N_{i,j,k} = [\inf N_{i,j,k}, \sup N_{i,j,k}]$ and $\sup M_{i,j,k} + \sup N_{i,j,k} \leq 1$, and $\varphi_{i,j,k}, \psi_{i,j,k}, \varphi_{i,j,k} + \psi_{i,j,k} \in [0, 1]$. Pair $\langle \varphi_{i,j,k}, \psi_{i,j,k} \rangle$ are the intuitionistic fuzzy values for neuron $V_{i,j}$, that are calculated by the formula (1), given below. The initial values of the neurons $V_{0,j}$ for $1 \leq j \leq w_0$ are $\langle \varphi_{0,j}, \psi_{0,j} \rangle$ and they are given in the beginning. Pair $\langle M_{i,j,k}, N_{i,j,k} \rangle$ corresponds to the condition that determines whether the neuron $V_{i,j}$ can send signal to neuron $V_{i+1,k}$ or not. This condition is given by predicate

$$P(V_{i,j}, V_{i+1,k}) = \begin{cases} 1, & \text{if } \varphi_{i,j,k} \in M_{i,j,k} \text{ and } \psi_{i,j,k} \in N_{i,j,k} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

The existing of the condition P is the reason that we called the described neuron network an Intuitionistic Fuzzy Neural Networks with Interval Valued Intuitionistic Fuzzy Conditions (IFNN-IVIFC).

Formally, it is defined by

$$\langle \langle w_0, w_1, \dots, w_s \rangle, \langle \langle \varphi_{0,1}, \psi_{0,1} \rangle, \dots, \langle \varphi_{0,w_0}, \psi_{0,w_0} \rangle \rangle, \{V_{i,j} | 0 \leq i \leq s; 1 \leq j \leq w_i\},$$

$$\{\{V_{i,1}, \dots, V_{i,w_i}\}, \{V_{i+1,1}, \dots, V_{i+1,w_{i+1}}\}, \{\langle M_{i,j,k}, N_{i,j,k} \rangle\}\},$$

where w_0, w_1, \dots, w_s are the numbers of neurons in the different layers;
 $\langle \varphi_{0,1}, \psi_{0,1} \rangle, \dots, \langle \varphi_{0,w_0}, \psi_{0,w_0} \rangle$ are the intuitionistic fuzzy values that the input neurons obtain;

$\{V_{i,j} | 0 \leq i \leq s; 1 \leq j \leq w_i\}$ is the set of all neurons;

$$\begin{aligned} & \{\{V_{i,1}, \dots, V_{i,w_i}\}, \{V_{i+1,1}, \dots, V_{i+1,w_{i+1}}\}, \{\langle M_{i,j,k}, N_{i,j,k} \rangle\}\} \\ = & \begin{array}{c|cccc} & V_{i+1,1} & \dots & V_{i+1,k} & \dots & V_{i+1,w_{i+1}} \\ V_{i,1} & \langle M_{i,1,1}, N_{i,1,1} \rangle & \dots & \langle M_{i,1,k}, N_{i,1,k} \rangle & \dots & \langle M_{i,1,w_{i+1}}, N_{i,1,w_{i+1}} \rangle \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ V_{i,j} & \langle M_{i,j,1}, N_{i,j,1} \rangle & \dots & \langle M_{i,j,k}, N_{i,j,k} \rangle & \dots & \langle M_{i,j,w_{i+1}}, N_{i,j,w_{i+1}} \rangle \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ V_{i,w_i} & \langle M_{i,w_i,1}, N_{i,w_i,1} \rangle & \dots & \langle M_{i,w_i,k}, N_{i,w_i,k} \rangle & \dots & \langle M_{i,w_i,w_{i+1}}, N_{i,w_i,w_{i+1}} \rangle \end{array} \end{aligned}$$

is an Index Matrix (IM; see [4, 7]) which elements – the pairs $\langle M_{i,j,k}, N_{i,j,k} \rangle$ that are defined above.

Before to start the IFNN-IVIFC functioning, the user must determine two the most suitable intuitionistic fuzzy conjunction and disjunction among operations $\&_1, \&_2, \& + 3, \vee_1, \vee_2, \vee_3$ (given above) or others, introduced in [1–3]. For example, $\circ, * \in \{\vee_1, \&_1, \vee_2, \&_2, \vee_3, \&_3\}$.

The results of the five logical operations from Sect. 2 can be interpreted by:

- \vee_2 – strong optimistic result,
- \vee_1 – optimistic result,
- \vee_3 ($\&_3$) – average result,
- $\&_1$ – pessimistic result,
- $\&_2$ – strong pessimistic result,

Let us denote the fixed operations by pair $\circ, *$ and let us assume that

$$P(V_{i,j}, V_{i+1,k}) \langle \varphi_{i,j}, \psi_{i,j} \rangle = \begin{cases} \langle \varphi_{i,j}, \psi_{i,j} \rangle, & \text{if } P(V_{i,j}, V_{i+1,k}) = 1 \\ \perp, & \text{otherwise} \end{cases}, \quad (2)$$

where symbol \perp denotes a lack of an expression. Let use define for operations \circ and $*$ for each IFP $\langle a, b \rangle$:

$$\langle a, b \rangle * \perp = \langle a, b \rangle = \perp * \langle a, b \rangle,$$

$$\langle a, b \rangle * \perp = \perp = \perp * \langle a, b \rangle,$$

and especially for operation $@$, the denominator must be decreased with the number of arguments that are exactly \perp . For example,

$$\textcircled{1, \perp, 2, 3, \perp} = \frac{1 + 2 + 3}{3} = 2.$$

On the first moment of the IFNN-IVIFC functioning, the signals enter input neurons with the respective intuitionistic fuzzy values.

After i time-steps (for $i \geq 1$), these signals from i -th layer (as in the standard case of the neuron networks) go to the neurons from the $(i + 1)$ -st layer, but now, having in mind formula (2), only these values that satisfy predicate (1) participate for the determination of the new values of the neurons from the $(i + 1)$ -st layer. These values are calculated by formula

$$\bigcirc_{j=1}^{w_i} ((P(V_{i,j}, V_{i+1,k}) \langle \varphi_{i,j}, \psi_{i,j} \rangle) * \langle M_{i,j,k}, N_{i,j,k} \rangle). \quad (3)$$

In practice, formula (3) is new one for neural networks theory. In the theory, a function f is defined that changes the values that are similar to (3). Here, function f is changed with one of the operators over IFPs, defined in Sect. 2. Let $O_{\alpha,\beta}$ be a fixed operator and $\alpha, \beta \in [0, 1]$ are its fixed arguments. Then

$$\langle \varphi_{i+1,k}, \psi_{i+1,k} \rangle = O_{\alpha,\beta} \left(\bigcirc_{j=1}^{w_i} (P(V_{i,j}, V_{i+1,k}) \langle \varphi_{i,j}, \psi_{i,j} \rangle) * \langle M_{i,j,k}, N_{i,j,k} \rangle \right).$$

4 Conclusion

In the present paper, we introduced a new type of neural networks, but in near future, it will be extended. For example, for each neuron $V_{i,j}$ we can define specific operations $\langle \circ, * \rangle$ and a specific operator $O_{\alpha_i,j,\beta_i,j}$ with specific arguments.

We must to show that the IFNN-IVIFC is a suitable Data Mining-tool in the sense of [8].

Acknowledgements The first two authors are thankful for the support provided by the Bulgarian National Science Fund under Grant Ref. No. DN-02-10/2016 “New Instruments for Knowledge Discovery from Data, and their Modelling”.

References

1. Angelova, N., Stoenchev, M.: Intuitionistic fuzzy conjunctions and disjunctions from first type, Annual of “Informatics” Section, Union of Scientists in Bulgaria, vol. 8, pp. 1–17 (2015/2016)
2. Angelova, N., Stoenchev, M.: Intuitionistic fuzzy conjunctions and disjunctions from second type. Issues Intuit.Istic Fuzzy Sets Gen. Nets **13**, 143–170 (2017)

3. Angelova, N., Stoenchev, M.: Intuitionistic fuzzy conjunctions and disjunctions from third type. *Notes Intuitionistic Fuzzy Sets* **23**(5), 29–41 (2017)
4. Atanassov K.: Generalized index matrices. *Comptes rendus de l'Academie Bulgare des Sciences*, **40**(11), 15–18 (1987)
5. Atanassov, K.: *Intuitionistic Fuzzy Sets*. Springer, Heidelberg (1999)
6. Atanassov, K.: *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin (2012)
7. Atanassov, K.: *Index Matrices: Towards an Augmented Matrix Calculus*. Springer, Cham (2014)
8. Atanassov, K.: Intuitionistic fuzzy logics as tools for evaluation of data mining processes. *Knowl.-Based Syst.* **80**, 122–130 (2015)
9. Atanassov, K.: *Intuitionistic Fuzzy Logics*. Springer, Cham (2017)
10. Atanassov, K., Sotirov, S., Krawczak, M.: Generalized net model of the intuitionistic fuzzy feed forward neural network. *Notes on Intuitionistic Fuzzy Sets* **15**(2), 18–23 (2009)
11. Atanassov, K., Szmidi, E., Kacprzyk, J.: On intuitionistic fuzzy pairs, *Notes on Intuitionistic Fuzzy Sets*, **19**(3), 1–13 (2013)
12. Atanassov, K., Vassilev, P., Kacprzyk, J., Szmidi, E.: On interval valued intuitionistic fuzzy pairs. *J. Univers. Math.* **1**(3), 261–268 (2018)
13. Hadjyisky, L., Atanassov, K.: Intuitionistic fuzzy model of a neural network. *BUSEFAL* **54**, 36–39 (1993)
14. Hadjyisky, L., Atanassov, K.: Generalized net model of the intuitionistic fuzzy neural networks. *Adv. Model. Anal.* **23**(2), 59–64 (1995)
15. Krawczak, M., El-Darzi, E., Atanassov, K., Tasseva, V.: Generalized net for control and optimization of real processes through neural networks using intuitionistic fuzzy estimations. *Notes on Intuitionistic Fuzzy Sets*, **13**(2), 54–60 (2007)
16. Kuncheva L., Atanassov, A.: An Intuitionistic fuzzy RBF network. In: *Proceedings of EUFIT96*, Aachen, Sept. 2–5, 777–781 (1996)
17. Sotirov, S., Atanassov, K.: Intuitionistic fuzzy feed forward neural network. *Cybern. Inf. Technol.* **9**(2), 62–68 (2009)
18. Sotirov, S., Sotirova, E., Atanassova, V., Atanassov, K., Castillo, O., Melin, P., Petkov, T., Surchev, S.: A Hybrid Approach for modular neural network design using intercriteria analysis and intuitionistic fuzzy logic. *Complexity* **2018**, Article ID 3927951, 11p. <https://doi.org/10.1155/2018/3927951>
19. Sotirov, S., Sotirova, E., Melin, P., Castillo, O., Atanassov, K.: Modular neural network pre-processing procedure with intuitionistic fuzzy intercriteria analysis method. In: T. Andreasen, et al. (Eds.) *Flexible Query Answering Systems 2015*, pp. 175–186. Springer, Cham (2016)

Generalised Atanassov Intuitionistic Fuzzy Sets Are Actually Intuitionistic Fuzzy Sets



Peter Vassilev and Krassimir Atanassov

Abstract The purpose of this paper is to investigate the relationship between the recently introduced Generalised Atanassov Intuitionistic Fuzzy Sets and the Intuitionistic Fuzzy Sets (also sometimes called Atanassov sets). As a result it will be established that the two notions are completely equivalent.

Keywords Intuitionistic fuzzy pairs · Intuitionistic fuzzy sets · Atanassov sets · Generalised Atanassov intuitionistic fuzzy sets

1 Introduction

In [9] the notion of Generalised Atanassov Intuitionistic Fuzzy Sets (GAIFS) is introduced. In fact there are six proposed generalizations. In the words of the authors: “*Atanassov’s definition assumes that the membership and non-membership functions must have their sum smaller than or equal to one for every element of the universe of discourse. While it is a good hypothesis in many practical situations, there are cases when this constraint does not work and it must be replaced by other relations.*” Further, we will consider the proposed generalizations and will show that they are in fact intuitionistic fuzzy sets in disguise. The situation is much alike that of the Vague Sets which were shown by Burillo and Bustince [7] to coincide completely with the intuitionistic fuzzy sets.

2 Preliminaries

Here we remind and introduce some basic definitions which will be used further.

Definition 1 (cf. [6]) *An intuitionistic fuzzy set (IFS) A over a universe X is an object of the form:*

P. Vassilev (✉) · K. Atanassov
Institute of Biophysics and Biomedical Engineering,
Bulgarian Academy of Sciences, Acad. G. Bonchev Str. Bl. 105, Sofia 1113, Bulgaria
e-mail: peter.vassilev@gmail.com

© Springer Nature Switzerland AG 2020
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_10

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in X \} \quad (1)$$

where $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ are mappings such that for all $x \in X$

$$\mu_A(x) + \nu_A(x) \leq 1 \quad (2)$$

The most common visual geometrical interpretation for the IFS is shown on Fig. 1.

Definition 2 (cf. [5]) An intuitionistic fuzzy pair (IFP) is an ordered couple of real non-negative numbers $\langle a, b \rangle$, with the additional constraint:

$$a + b \leq 1. \quad (3)$$

Further we will use exclusively IFPs, as IFSs may be viewed as elements of the universe of discourse labelled with IFPs.

Lemma 1 Let $a \in [0, 1]$ and $b \in [0, 1]$ and

$$a + b \geq 1. \quad (4)$$

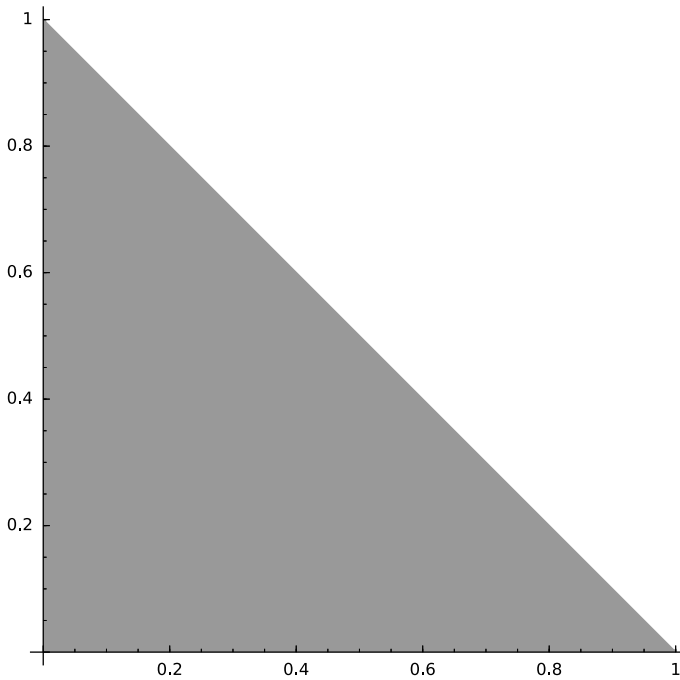


Fig. 1 Visual representation of IFS

Then,

$$\langle 1 - a, 1 - b \rangle \quad (5)$$

is an IFP.

Proof By (4) we have

$$0 \geq 1 - b - a$$

adding 1 to both sides of the inequality we obtain:

$$1 - b + 1 - a \leq 1.$$

Since, $a \in [0, 1]$ it follows that $1 - a \in [0, 1]$. The same argument holds for $1 - b$.

Hence,

$$1 - b + 1 - a \geq 0.$$

and thus $\langle 1 - a, 1 - b \rangle$ is an IFP.

Similar construction, but with another sense is discussed in P. Dworniczak's papers [10] and [11].

Lemma 2 Let $a \in [0, 1]$ and $b \in [0, 1]$ and

$$a \geq b \quad (6)$$

Then,

$$\langle 1 - a, b \rangle \quad (7)$$

is an IFP.

Proof By (6) we have

$$-a + b \leq 0$$

adding 1 to both sides of the inequality we obtain:

$$1 - a + b \leq 1.$$

Since, $a \in [0, 1]$ it follows that $1 - a \in [0, 1]$. Also $b \in [0, 1]$.

Hence,

$$1 - a + b \geq 0.$$

and thus $\langle 1 - a, b \rangle$ is an IFP.

Lemma 3 Let $\langle a, b \rangle$ be an IFP. Then,

$$1 - a + 1 - b \geq 1. \quad (8)$$

Proof By (3) we have

$$1 - a - b \geq 0$$

adding 1 to both sides of the inequality we obtain:

$$1 - a + 1 - b \geq 1.$$

In what follows we will briefly remind the definition of the different GAIFSs and will show how they can be represented by IFSs.

Definition 3 ([9, Definition 5]). Let X be a non-empty universe of discourse. Then a generalised Atanassov intuitionistic fuzzy set (GAIFS1) is described by

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in X \},$$

where the membership/non-membership functions $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ satisfy the condition (see Fig. 2)

$$\mu_A(x) + \nu_A(x) \geq 1 \tag{9}$$

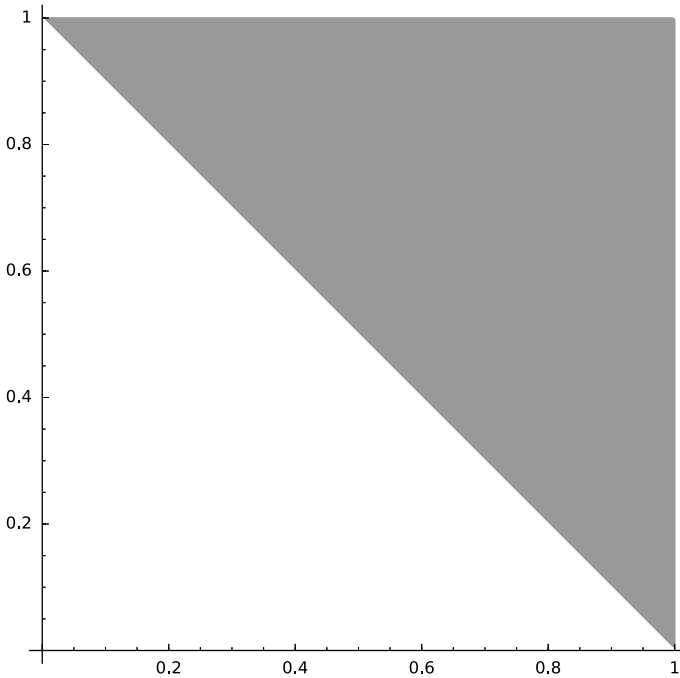


Fig. 2 Visual representation of GAIFS1. Note that it is a mirror image of an IFS (see Fig. 1)

The degree of indeterminacy of x to A is defined as:

$$\pi_A(x) = \mu_A(x) + \nu_A(x) - 1 \quad (10)$$

and once again, clearly:

$$0 \leq \pi_A(x) \leq 1 \quad \forall x \in X$$

In [4], bijective functions transforming the unit square to the intuitionistic fuzzy interpretation triangle (Fig. 1) are given.

Definition 4 ([9, Definition 6]). Let X be a non-empty universe of discourse. Then a generalised Atanassov intuitionistic fuzzy set (GAIFS2) is described by

$$A = \{(x, \mu_A(x), \nu_A(x)) | x \in X\},$$

where the membership/non-membership functions $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ satisfy the condition

$$\mu_A(x) \leq \nu_A(x) \quad (11)$$

The degree of indeterminacy of x to A is defined as:

$$\pi_A(x) = \nu_A(x) - \mu_A(x) \quad (12)$$

Definition 5 ([9, Definition 7]). Let X be a non-empty universe of discourse. Then a generalised Atanassov intuitionistic fuzzy set (GAIFS3) is described by

$$A = \{(x, \mu_A(x), \nu_A(x)) | x \in X\},$$

where the membership/non-membership functions $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ satisfy the condition

$$\nu_A(x) \leq \mu_A(x) \quad (13)$$

The degree of indeterminacy of x to A is defined as:

$$\pi_A(x) = \mu_A(x) - \nu_A(x) \quad (14)$$

The geometrical interpretation of GAIFS2 and GAIFS3 is shown on Fig. 3

Definition 6 ([9, Definition 8]). Let X be a non-empty universe of discourse. Then a generalised Atanassov intuitionistic fuzzy set (GAIFS4) is described by

$$A = \{(x, \mu_A(x), \nu_A(x)) | x \in X\},$$

where the membership/non-membership functions $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ satisfy the condition

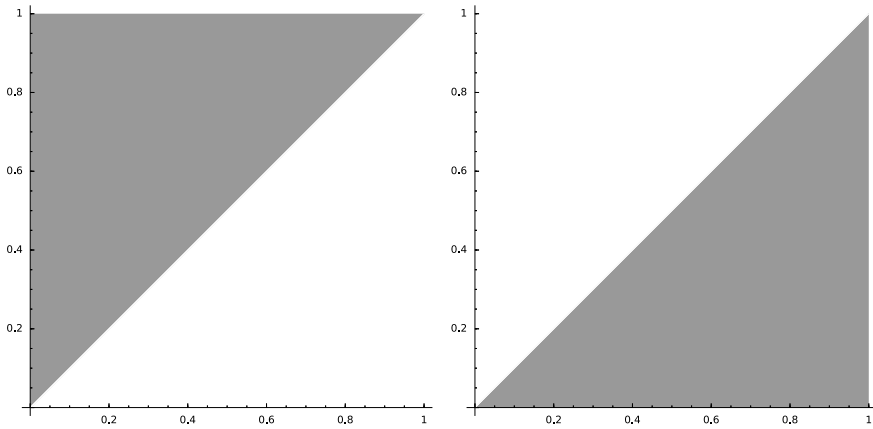


Fig. 3 Visual representation of GAIFS2 and GAIFS3, respectively

$$\begin{aligned}
 \mu_A(x) &\geq \nu_A(x) \ \& \ \mu_A(x) + \nu_A(x) \geq 1 \\
 \mu_A(x) &\leq \nu_A(x) \ \& \ \mu_A(x) + \nu_A(x) \leq 1 \quad \forall x \in X.
 \end{aligned}
 \tag{15}$$

Definition 7 ([9, Definition 9]). *Let X be a non-empty universe of discourse. Then a generalised Atanassov intuitionistic fuzzy set (GAIFS5) is described by*

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle \mid x \in X \},$$

where the membership/non-membership functions $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ satisfy the condition

$$\begin{aligned}
 \mu_A(x) &\leq \nu_A(x) \ \& \ \mu_A(x) + \nu_A(x) \leq 1 \\
 \mu_A(x) &\geq \nu_A(x) \ \& \ \mu_A(x) + \nu_A(x) \geq 1 \quad \forall x \in X.
 \end{aligned}
 \tag{16}$$

The geometrical interpretation of GAIFS4 and GAIFS5 is given in Fig. 4

The definition of GAIFS6 as quoted below coincides completely with the definitions for IFS of second type introduced for the first time in 1989 in [1] and subsequently investigated in [2]. Their geometrical interpretation is presented in the right hand-side in Fig. 5.

Definition 8 ([9, Definition 10], cf. [1–3, 13]). *Let X be a non-empty universe of discourse. Then a generalised Atanassov intuitionistic fuzzy set (GAIFS6) is described by*

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle \mid x \in X \},$$

where the membership/non-membership functions $\mu_A : X \rightarrow [0, 1]$ and $\nu_A : X \rightarrow [0, 1]$ satisfy the condition

$$\mu_A^2(x) + \nu_A^2(x) \leq 1 \tag{17}$$

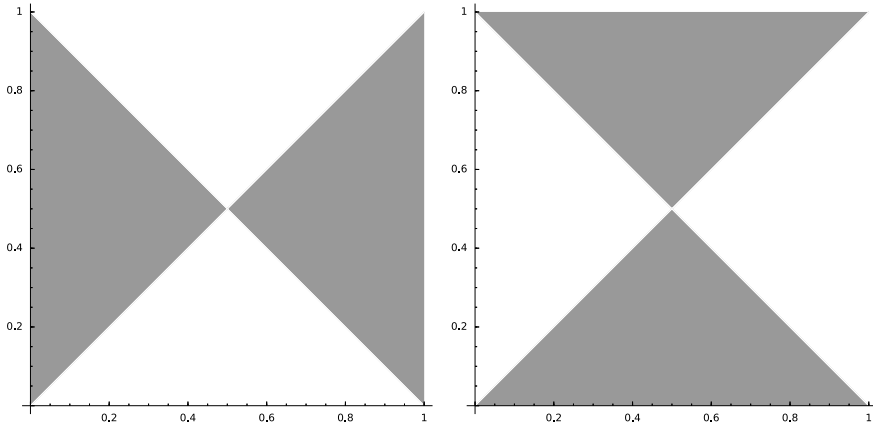


Fig. 4 Visual representation of GAIFS4 and GAIFS5, respectively

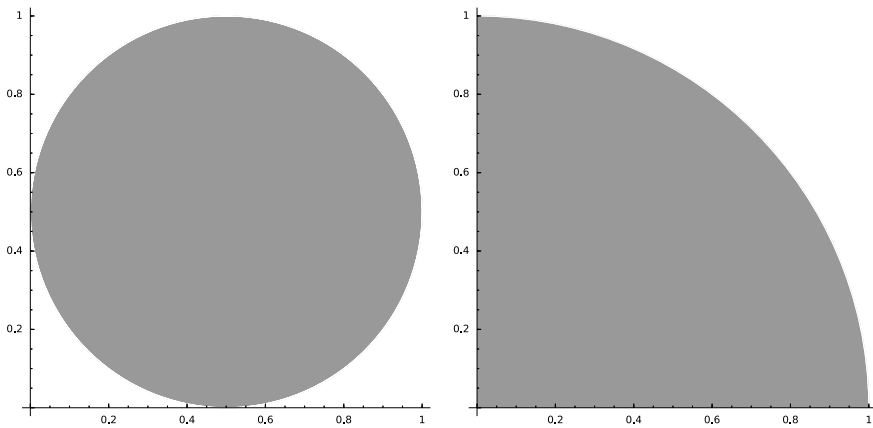


Fig. 5 Figure given in [9, Definition 10] corresponding to Definition 8 but really fulfilling (18) (left), actual figure corresponding to (17) (right)

The figure accompanying [9, Definition 10] (see Fig. 5), however, does not match the above definition. In fact the satisfied inequality for this figure is:

$$\left(\mu_A(x) - \frac{1}{2}\right)^2 + \left(\nu_A(x) - \frac{1}{2}\right)^2 \leq \frac{1}{4} \tag{18}$$

3 Main Result

Having reminded all the relevant definitions we are now ready to prove our main result, namely:

Theorem 1 *GAIIFS1 – GAIIFS5 coincide with the standard Intuitionistic Fuzzy Sets, after appropriate change of variables.*

Proof Let a GAIIFS1 set be given by:

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle | x \in X\},$$

then by Lemma 1 we can construct the IFS

$$A_{GAIIFS1} = \{\langle x, 1 - \mu_A(x), 1 - \nu_A(x) \rangle | x \in X\}.$$

In both cases $\pi_A(x)$ coincides and the ordering between sets are recovered.

Let a GAIIFS2 be given by:

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle | x \in X\},$$

then by Lemma 2 we can construct the IFS

$$A_{GAIIFS2} = \{\langle x, \mu_A(x), 1 - \nu_A(x) \rangle | x \in X\},$$

which also recovers the desired orderings and defines the same value for $\pi_A(x)$.

Let a GAIIFS3 be given by:

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle | x \in X\},$$

then by Lemma 2 we can construct the IFS

$$A_{GAIIFS3} = \{\langle x, 1 - \mu_A(x), \nu_A(x) \rangle | x \in X\},$$

which also recovers the desired orderings and defines the same value for $\pi_A(x)$.

Let GAIIFS4 be given by:

$$A = \{\langle x, \mu_A(x), \nu_A(x) \rangle | x \in X\},$$

Then we can construct the IFS:

$$A_{GAIIFS4} = \{\langle x, \mu_A^*(x), \nu_A^*(x) \rangle | x \in X\},$$

where

$$\mu_A^* = \begin{cases} \mu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ 1 - \nu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ \frac{\mu_A(x)}{2} & \text{if } \mu_A(x) = \nu_A(x) \end{cases}, \quad \nu_A^* = \begin{cases} \nu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ 1 - \mu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ \frac{\nu_A(x)}{2} & \text{if } \mu_A(x) = \nu_A(x) \end{cases}.$$

Conversely, let an IFS

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in X \},$$

be given, then we can immediately construct the set

$$\mu_A^G = \begin{cases} \mu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ 1 - \nu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ 2\mu_A(x) & \text{if } \mu_A(x) = \nu_A(x) \end{cases} \quad \nu_A^G = \begin{cases} \nu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ 1 - \mu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ 2\nu_A(x) & \text{if } \mu_A(x) = \nu_A(x) \end{cases}$$

which would be a GAIFS4 set. We illustrate the construction by Fig. 6.

Completely analogous is the construction in the case for GAIFS5.

Let GAIFS5 be given by:

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in X \},$$

Then we can construct the IFS:

$$A_{GAIFS5} = \{ \langle x, \mu_A^*(x), \nu_A^*(x) \rangle | x \in X \},$$

where

$$\mu_A^* = \begin{cases} \mu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ 1 - \nu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ \frac{\mu_A(x)}{2} & \text{if } \mu_A(x) = \nu_A(x) \end{cases} \quad \nu_A^* = \begin{cases} \nu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ 1 - \mu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ \frac{\nu_A(x)}{2} & \text{if } \mu_A(x) = \nu_A(x) \end{cases}$$

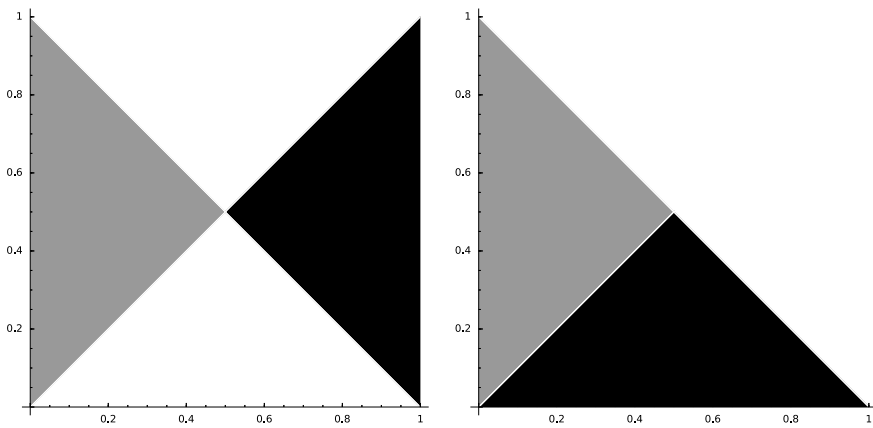


Fig. 6 Visual representation of GAIFS4 and IFS correspondence

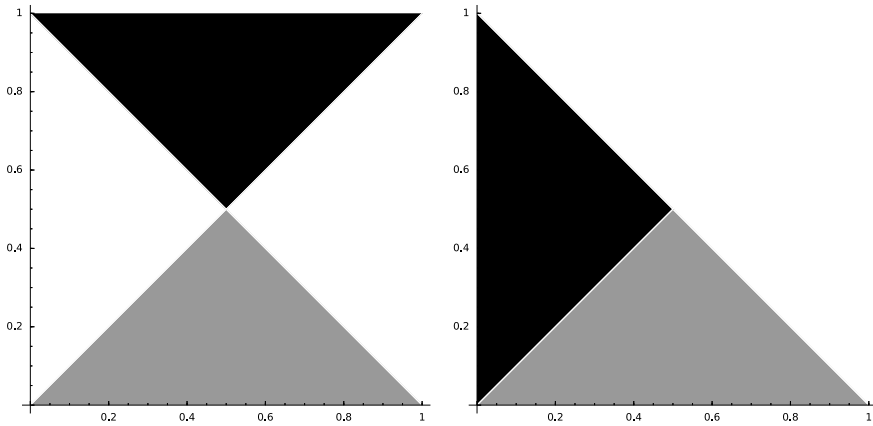


Fig. 7 Visual representation of GAIFS5 and IFS correspondence

Conversely, let an IFS

$$A = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in X \},$$

be given, then we can immediately construct the set

$$\mu_A^G = \begin{cases} \mu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ 1 - \nu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ 2\mu_A(x) & \text{if } \mu_A(x) = \nu_A(x) \end{cases} \quad \nu_A^G = \begin{cases} \nu_A(x) & \text{if } \mu_A(x) > \nu_A(x); \\ 1 - \mu_A(x) & \text{if } \mu_A(x) < \nu_A(x); \\ 2\nu_A(x) & \text{if } \mu_A(x) = \nu_A(x) \end{cases}$$

which would be a GAIFS5 set.

We illustrate the construction by Fig. 7.

Thus, we have shown that all the considered GAIFSs are completely representable by IFSs.

4 Conclusion

In the presented work we have clearly demonstrated that in essence the proposed Generalised Atanassov intuitionistic fuzzy sets can be completely described by the Intuitionistic Fuzzy Sets. The situation here is quite similar to the case of Vague sets. Unlike the case of Interval-Valued Fuzzy Sets, Interval Type-2 Fuzzy Sets and Intuitionistic Fuzzy Sets which may be viewed to describe one and the same mathematical object but on semantical level allow for different treatment of the uncertainty (see [8, 12]), no new advantage is evident in the proposed generalizations of IFS in [9]. Therefore, we conclude that the proposed GAIFS are superficial and unnecessary alternative forms of IFS.

Acknowledgements Work presented here is supported by Program for career development of young scientists, Bulgarian Academy of Sciences, Grant number DFNP-17-139 “Investigation of different extensions of fuzzy sets and assessment of their correctness”.

References

1. Atanassov, K.: Geometrical interpretation of the elements of the intuitionistic fuzzy objects, Preprint IM-MFAIS-1-89, Sofia, 1989. Reprinted: *Int. J. Bioautomation*, 2016, **20**(S1), S27–S42 (1989)
2. Atanassov, K.: A second type of intuitionistic fuzzy sets. *BUSEFAL* **56**, 66–70 (1993)
3. Atanassov, K.: *Intuitionistic Fuzzy Sets*. Springer, Heidelberg (1999)
4. Atanassov, K.: Remark on a property of the intuitionistic fuzzy interpretation triangle. *Notes on Intuitionistic Fuzzy Sets* **8**(1), 34–36 (2002)
5. Atanassov, K., Szmidt, E., Kacprzyk, J.: On Intuitionistic Fuzzy Pairs. *Notes on Intuitionistic Fuzzy Sets* **19**(3), 1–13 (2013)
6. Atanassov, K.T.: *On Intuitionistic Fuzzy Sets Theory*. Springer, Berlin (2012)
7. Bustince, H., Burillo, P.: Vague sets are intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **79**(3), 403–405 (1996)
8. Cornelis, C., Atanassov, K., Kerre, E.: Intuitionistic fuzzy sets and interval-valued fuzzy sets: a critical comparison. In: Wagenknecht, M., Hampel, R. (eds.) *Proceedings of Third International Conference on Fuzzy Logic and Technology*, pp. 159–163 (2003)
9. Despi, I., Opris, D., Yalcin, E.: Generalised Atanassov intuitionistic fuzzy sets. In: *Proceedings of the Fifth International Conference on Information, Process, and Knowledge Management 2013, (eKNOW 2013)*, pp. 51–56 (2013)
10. Dworniczak, P.: A note on the unconscious experts’ evaluations in the intuitionistic fuzzy environment. *Notes on Intuitionistic Fuzzy Sets* **18**(3), 23–29 (2012)
11. Dworniczak, P.: Further remarks about the unconscious experts’ evaluations in the intuitionistic fuzzy environment. *Notes on Intuitionistic Fuzzy Sets* **19**(1), 27–31 (2013)
12. Niewiadomski, A.: Interval-valued and interval type-2 fuzzy sets: a subjective comparison. In: *Proceedings of FUZZ-IEEE’07, London*, pp. 1198–1203 (2007)
13. Vassilev, P.: *Intuitionistic fuzzy sets with membership and non-membership functions in metric relation*, Ph.D. Thesis (in Bulgarian) (2013)

The Numerical Solution of Intuitionistic Fuzzy Differential Equations by the Third Order Runge-Kutta Nyström Method



Bouchra Ben Amma, Said Melliani and S. Chadli

Abstract In this work we introduce numerical algorithm for solving intuitionistic fuzzy differential equations. We discuss in detail a numerical method based on a Runge-Kutta Nyström method. Sufficiently conditions for the convergence of the proposed algorithms are given and their applicability is illustrated via an example.

Keywords Numerical solution · Intuitionistic fuzzy differential equations · Runge-Kutta Nyström method

1 Introduction

Generalizations of fuzzy sets theory [37] is considered to be one of intuitionistic fuzzy set (IFS). Later on Atanassov generalized the concept of fuzzy set and introduced the idea of intuitionistic fuzzy set [2, 6]. Atanassov [3] explored the concept of fuzzy set theory by intuitionistic fuzzy set (IFS) theory. Now-a-days, IFSs are being studied extensively and being used in different disciplines of Science and Technology. Amongst the all research works mainly on IFS we can include Atanassov [4, 5, 8, 9], Atanassov and Gargov [7], Szmidt and Kacprzyk [33], Buhaescu [10, 11], Ban [12], Deschrijver and Kerre [21], Cornelis et al. [19], Gerstenkorn and Manko [22], Mahapatra and Roy [27], Adak et al. [1], Melliani et al. [28]. Further improvement of IFS theory, together with intuitionistic fuzzy geometry, intuitionistic fuzzy logic, intuitionistic fuzzy topology, an intuitionistic fuzzy approach to artificial intelligence, and intuitionistic fuzzy generalized nets can be found in [32], then they

B. Ben Amma (✉) · S. Melliani · S. Chadli

Laboratory of Applied Mathematics and Scientific Computing, Faculty of Sciences and Technologies, Sultan Moulay Slimane University, BP 523, 23000 Beni Mellal, Morocco
e-mail: bouchrabenamma@gmail.com

S. Melliani

e-mail: s.melliani@yahoo.fr

S. Chadli

e-mail: sa.chadli@yahoo.fr

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_11

are very necessary and powerful tool in modeling imprecision, valuable applications of IFSs have been flourished in many different fields, medical diagnosis [20], drug selection [23], along pattern recognition [24], microelectronic fault analysis [34], weight assessment [35], and decision-making problems [25, 36].

The concept of intuitionistic fuzzy differential equations was first introduced by Melliani and Chadli [26]. The first step which included applicable definitions of intuitionistic fuzzy derivative and the intuitionistic fuzzy integral was followed by introducing intuitionistic fuzzy differential equations and establishing sufficient conditions for the existence of unique solutions to these equations [16, 17, 29, 30]. The topics of numerical methods for solving intuitionistic fuzzy differential equations have been rapidly growing in recent years. It is difficult to obtain exact solution for intuitionistic fuzzy DEs and hence some applications of numerical methods such as the intuitionistic fuzzy Euler and Taylor methods [13], Runge-Kutta of order four [15], Runge-Kutta Gill [18], Variational iteration method [31], Adams-Bashforth, Adams-Moulton and Predictor-Corrector methods [14].

The structure of this paper organizes as follows: In Sect. 2, some basic definitions and results are brought. In Sect. 3 the intuitionistic fuzzy initial value problem is being discussed. The numerically solving intuitionistic fuzzy differential equation by the Runge-Kutta Nyström method is discussed in Sect. 4. The proposed algorithm is illustrated by solving an example in Sect. 5 and conclusion is in Sect. 6.

2 Preliminaries

Consider the initial value problem:

$$\begin{cases} y'(t) = g(t, y(t)), & t \in [t_0, T] \\ y(t_0) = y_0 \end{cases} \tag{1}$$

The basis of all Runge-Kutta method is to express the difference between the value of y at t_{n+1} and t_n as

$$y_{n+1} - y_n = \sum_{i=1}^{i=m} w_i k_i, \tag{2}$$

where for $i = 1, 2, 3, \dots, m$ the w_i 's are constants and

$$k_i = hg(t_n + c_i h, y_n + \sum_{j=1}^{i-1} \beta_{ij} k_j) \tag{3}$$

Equation 2 is to be exact for powers of h through h^m , because it is to be coincident with Taylor series of order m . Therefore, the truncation error T_m can be written as

$$T_m = \gamma_m h^{m+1} + O(h^{m+2}) \tag{4}$$

The true value of γ_m will generally be much less than the bound of Theorem 1. Thus, if the $O(h^{m+2})$ term is small compared to $\gamma_m h^{m+1}$ for small h , then the bound on $\gamma_m h^{m+1}$ will usually be a bound on the error as a whole. The famous nonzero constants c_i, β_{ij} in the Runge-Kutta Nyström method of order three are:

$$c_1 = 0, c_2 = \frac{2}{3}, c_3 = \frac{2}{3}, \beta_{21} = \frac{2}{3}, \beta_{32} = \frac{2}{3}$$

where $m = 3$. Hence we have

$$\begin{aligned} k_1 &= hg(t_i, y_i) \\ k_2 &= hg(t_i + \frac{2}{3}h, y_i + \frac{2}{3}k_1) \\ k_3 &= hg(t_i + \frac{2}{3}h, y_i + \frac{2}{3}k_2) \\ y_{i+1} &= y_i + \frac{1}{8}(2k_1 + 3k_2 + 3k_3) \end{aligned}$$

where

$$t_0 < t_1 < t_2 < \dots < t_N = T, \quad h = \frac{T - t_0}{N}, \quad t_i = t_0 + ih, \quad i = 0, 1, \dots, N \tag{5}$$

Theorem 1 *Let $g(t, y)$ belong to $C^3[a, b]$ and let it's partial derivatives are bounded and assume there exists, L, M positive numbers, such that*

$$|g(t, y)| < M, \quad \left| \frac{\partial^{i+j} f}{\partial t^i \partial y^j} \right| < \frac{L^{i+j}}{M^{j-i}}, \quad i + j \leq m \tag{6}$$

then in the Runge-Kutta Nyström method we have

$$y(t_{i+1}) - y_{i+1} \approx \frac{25}{108} h^4 ML^3 + O(h^5)$$

Throughout this paper, $(\mathbb{R}, B(\mathbb{R}), \mu)$ denotes a complete finite measure space.

Let us $P_k(\mathbb{R})$ the set of all nonempty compact convex subsets of \mathbb{R} .

we denote by

$$IF_1 = \text{IF}(\mathbb{R}) = \{ \langle u, v \rangle : \mathbb{R} \rightarrow [0, 1]^2, |\forall x \in \mathbb{R} \ 0 \leq u(x) + v(x) \leq 1 \}$$

An element $\langle u, v \rangle$ of IF_1 is said an intuitionistic fuzzy number if it satisfies the following conditions

- (i) $\langle u, v \rangle$ is normal i.e there exists $x_0, x_1 \in \mathbb{R}$ such that $u(x_0) = 1$ and $v(x_1) = 1$.
- (ii) u is fuzzy convex and v is fuzzy concave.
- (iii) u is upper semi-continuous and v is lower semi-continuous
- (iv) $\text{supp}\langle u, v \rangle = \text{cl}\{x \in \mathbb{R} : |v(x) < 1\}$ is bounded.

so we denote the collection of all intuitionistic fuzzy number by IF_1 .

Definition 1 Let $\langle u, v \rangle$ an element of IF_1 and $\alpha \in [0, 1]$, we define the following sets:

$$\begin{aligned} [\langle u, v \rangle]_l^+(\alpha) &= \inf\{x \in \mathbb{R} \mid u(x) \geq \alpha\}, & [\langle u, v \rangle]_r^+(\alpha) &= \sup\{x \in \mathbb{R} \mid u(x) \geq \alpha\} \\ [\langle u, v \rangle]_l^-(\alpha) &= \inf\{x \in \mathbb{R} \mid v(x) \leq 1 - \alpha\}, & [\langle u, v \rangle]_r^-(\alpha) &= \sup\{x \in \mathbb{R} \mid v(x) \leq 1 - \alpha\} \end{aligned}$$

Remark 1

$$\begin{aligned} [\langle u, v \rangle]_\alpha &= \left[[\langle u, v \rangle]_l^+(\alpha), [\langle u, v \rangle]_r^+(\alpha) \right] \\ [\langle u, v \rangle]^\alpha &= \left[[\langle u, v \rangle]_l^-(\alpha), [\langle u, v \rangle]_r^-(\alpha) \right] \end{aligned}$$

A Triangular Intuitionistic Fuzzy Number (TIFN) $\langle u, v \rangle$ is an intuitionistic fuzzy set in \mathbb{R} with the following membership function u and non-membership function v :

$$\begin{aligned} u(x) &= \begin{cases} \frac{x-a_1}{a_2-a_1} & \text{if } a_1 \leq x \leq a_2 \\ \frac{a_3-x}{a_3-a_2} & \text{if } a_2 \leq x \leq a_3 \\ 0 & \text{otherwise} \end{cases} \\ v(x) &= \begin{cases} \frac{a_2-x}{a_2-a'_1} & \text{if } a'_1 \leq x \leq a_2 \\ \frac{x-a_2}{a'_3-a_2} & \text{if } a_2 \leq x \leq a'_3 \\ 1 & \text{otherwise} \end{cases} \end{aligned}$$

where $a'_1 \leq a_1 \leq a_2 \leq a_3 \leq a'_3$ and $u(x), v(x) \leq 0.5$ for $u(x) = v(x), \forall x \in \mathbb{R}$
 This TIFN is denoted by $\langle u, v \rangle = \langle a_1, a_2, a_3; a'_1, a_2, a'_3 \rangle$ where,

$$[\langle u, v \rangle]_\alpha = [a_1 + \alpha(a_2 - a_1), a_3 - \alpha(a_3 - a_2)] \tag{7}$$

$$[\langle u, v \rangle]^\alpha = [a'_1 + \alpha(a_2 - a'_1), a'_3 - \alpha(a'_3 - a_2)] \tag{8}$$

Theorem 2 ([28])

d_∞ define a metric on IF_1 .

Theorem 3 ([28])

The metric space (IF_1, d_∞) is complete.

Remark 2 If $F: [a, b] \rightarrow IF_1$ is Hukuhara differentiable and its Hukuhara derivative F' is integrable over $[0, 1]$ then

$$F(t) = F(t_0) \oplus \int_{t_0}^t F'(s)ds$$

Definition 2 Let $\langle u, v \rangle$ and $\langle u', v' \rangle \in IF_1$, the H-difference is the IFN $\langle z, w \rangle \in IF_1$, if it exists, such that

$$\langle u, v \rangle \ominus \langle u', v' \rangle = \langle z, w \rangle \iff \langle u, v \rangle = \langle u', v' \rangle \oplus \langle z, w \rangle$$

Definition 3 A mapping $F: [a, b] \rightarrow IF_1$ is said to be Hukuhara derivable at t_0 if there exist $F'(t_0) \in IF_1$ such that both limits:

$$\lim_{\Delta t \rightarrow 0^+} \frac{F(t_0 + \Delta t) \ominus F(t_0)}{\Delta t}$$

and

$$\lim_{\Delta t \rightarrow 0^+} \frac{F(t_0) \ominus F(t_0 - \Delta t)}{\Delta t}$$

exist and they are equal to $F'(t_0) = \langle u'(t_0), v'(t_0) \rangle$, which is called the Hukuhara derivative of F at t_0 .

3 Intuitionistic Fuzzy Cauchy Problem

In this section we consider the initial value problem for the intuitionistic fuzzy differential equation

$$\begin{cases} \langle u, v \rangle'(t) = F(t, \langle u, v \rangle(t)), & t \in I \\ \langle u, v \rangle(t_0) = \langle u_{t_0}, v_{t_0} \rangle \in IF_1 \end{cases} \tag{9}$$

where $\langle u, v \rangle \in IF_1$ is unknown $I = [t_0, T]$ and $F: I \times IF_1 \rightarrow IF_1$.

$\langle u, v \rangle(t_0)$ is intuitionistic fuzzy number.

Denote the α - level set

$$[\langle u, v \rangle(t)]_\alpha = \left[[\langle u, v \rangle(t)]_l^+(\alpha), [\langle u, v \rangle(t)]_r^+(\alpha) \right]$$

$$[\langle u, v \rangle(t)]^\alpha = \left[[\langle u, v \rangle(t)]_l^-(\alpha), [\langle u, v \rangle(t)]_r^-(\alpha) \right]$$

and

$$\begin{aligned}
 [\langle u, v \rangle(t_0)]_\alpha &= \left[[\langle u, v \rangle(t_0)]_l^+(\alpha), [\langle u, v \rangle(t_0)]_r^+(\alpha) \right] \\
 [\langle u, v \rangle(t_0)]^\alpha &= \left[[\langle u, v \rangle(t_0)]_l^-(\alpha), [\langle u, v \rangle(t_0)]_r^-(\alpha) \right] \\
 [F(t, \langle u, v \rangle(t))]_\alpha &= \left[F_l^+(t, \langle u, v \rangle(t); \alpha), F_r^+(t, \langle u, v \rangle(t); \alpha) \right] \\
 [F(t, \langle u, v \rangle(t))]^\alpha &= \left[F_l^-(t, \langle u, v \rangle(t); \alpha), F_r^-(t, \langle u, v \rangle(t); \alpha) \right]
 \end{aligned}$$

where

$$\begin{aligned}
 F_1^+(t, \langle u, v \rangle(t); \alpha) &= \min \left\{ F(t, z) \mid z \in \left[[\langle u, v \rangle(t)]_l^+(\alpha), [\langle u, v \rangle(t)]_r^+(\alpha) \right] \right\} \\
 F_2^+(t, \langle u, v \rangle(t); \alpha) &= \max \left\{ F(t, z) \mid z \in \left[[\langle u, v \rangle(t)]_l^+(\alpha), [\langle u, v \rangle(t)]_r^+(\alpha) \right] \right\} \\
 F_3^-(t, \langle u, v \rangle(t); \alpha) &= \min \left\{ F(t, z) \mid z \in \left[[\langle u, v \rangle(t)]_l^-(\alpha), [\langle u, v \rangle(t)]_r^-(\alpha) \right] \right\} \\
 F_4^-(t, \langle u, v \rangle(t); \alpha) &= \max \left\{ F(t, z) \mid z \in \left[[\langle u, v \rangle(t)]_l^-(\alpha), [\langle u, v \rangle(t)]_r^-(\alpha) \right] \right\}
 \end{aligned} \tag{10}$$

Denote

$$\begin{aligned}
 F_1^+(t, \langle u, v \rangle(t); \alpha) &= P(t, [\langle u, v \rangle(t)]_l^+(\alpha), [\langle u, v \rangle(t)]_r^+(\alpha)) \\
 F_2^+(t, \langle u, v \rangle(t); \alpha) &= Q(t, [\langle u, v \rangle(t)]_l^+(\alpha), [\langle u, v \rangle(t)]_r^+(\alpha)) \\
 F_3^-(t, \langle u, v \rangle(t); \alpha) &= R(t, [\langle u, v \rangle(t)]_l^-(\alpha), [\langle u, v \rangle(t)]_r^-(\alpha)) \\
 F_4^-(t, \langle u, v \rangle(t); \alpha) &= S(t, [\langle u, v \rangle(t)]_l^-(\alpha), [\langle u, v \rangle(t)]_r^-(\alpha))
 \end{aligned} \tag{11}$$

Sufficient conditions for the existence of an unique solution to Eq. (9) are:

1. Continuity of F
2. Lipschitz condition: for any pair $(t, \langle u, v \rangle), (t, \langle u', v' \rangle) \in I \times IF_1$, we have

$$d_\infty(F(t, \langle u, v \rangle), F(t, \langle u', v' \rangle)) \leq K d_\infty(\langle u, v \rangle, \langle u', v' \rangle) \tag{12}$$

where $K > 0$ is a given constant.

Theorem 4 [16] *Let us suppose that the following conditions hold:*

- (a) *Let $R_0 = [t_0, t_0 + p] \times \bar{B}(\langle u, v \rangle_{t_0}, q)$, $p, q \geq 0, \langle u, v \rangle_{t_0} \in IF_1$ where $\bar{B}(\langle u, v \rangle_{t_0}, q) = \{\langle u, v \rangle \in IF_1 : d_\infty(\langle u, v \rangle, \langle u, v \rangle_{t_0}) \leq q\}$ denote a closed ball in IF_1 and let $F: R_0 \rightarrow IF_1$ be a continuous function such that $d_\infty(F(t, \langle u, v \rangle), 0_{(1,0)}) \leq M$ for all $(t, \langle u, v \rangle) \in R_0$.*

(b) Let $g : [t_0, t_0 + p] \times [0, q] \longrightarrow \mathbb{R}$ such that $g(t, 0) \equiv 0$ and $0 \leq g(t, x) \leq M_1$, $\forall t \in [t_0, t_0 + p], 0 \leq x \leq q$ such that $g(t, x)$ is non-decreasing in u and g is such that the initial value problem

$$x'(t) = g(t, x(t)), x(t_0) = x_0. \tag{13}$$

has only the solution $x(t) \equiv 0$ on $[t_0, t_0 + p]$

(c) We have $d_\infty(F(t, \langle u, v \rangle), F(t, \langle z, w \rangle)) \leq g(t, d_\infty(\langle u, v \rangle, \langle z, w \rangle)), \forall (t, \langle u, v \rangle), (t, \langle z, w \rangle) \in R_0$ and $d_\infty(\langle u, v \rangle, \langle z, w \rangle) \leq q$. Then the intuitionistic fuzzy initial value problem

$$\begin{cases} \langle u, v \rangle' &= F(t, \langle u, v \rangle), \\ \langle u, v \rangle(t_0) &= \langle u, v \rangle_{t_0} \end{cases} \tag{14}$$

has an unique solution $\langle u, v \rangle \in C^1[[t_0, t_0 + r], B(x_0, q)]$ on $[t_0, t_0 + r]$ where $r = \min\{p, \frac{q}{M}, \frac{q}{M_1}, d\}$ and the successive iterations

$$\langle u, v \rangle_0(t) = \langle u, v \rangle_{t_0}, \quad \langle u, v \rangle_{n+1}(t) = \langle u, v \rangle_{t_0} + \int_{t_0}^t F(s, \langle u, v \rangle_n(s)) ds \tag{15}$$

converge to $\langle u, v \rangle(t)$ on $[t_0, t_0 + r]$.

4 Third-Order Runge-Kutta Nyström Method

Let the exact solutions

$$\begin{aligned} [\langle U, V \rangle(t_n)]_\alpha &= \left[[\langle U, V \rangle(t_n)]_l^+(\alpha), [\langle U, V \rangle(t_n)]_r^+(\alpha) \right], \\ [\langle U, V \rangle(t_n)]^\alpha &= \left[[\langle U, V \rangle(t_n)]_l^-(\alpha), [\langle U, V \rangle(t_n)]_r^-(\alpha) \right] \end{aligned}$$

be approximated by

$$\begin{aligned} [\langle u, v \rangle(t_n)]_\alpha &= \left[[\langle u, v \rangle(t_n)]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) \right], \\ [\langle u, v \rangle(t_n)]^\alpha &= \left[[\langle u, v \rangle(t_n)]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^-(\alpha) \right] \end{aligned}$$

at $t_n, 0 \leq n \leq N$.

The solutions are calculated by grid points at

$$t_0 < t_1 < t_2 < \dots < t_N = T, \quad h = \frac{T - t_0}{N}, \quad t_n = t_0 + nh, \quad n = 0, 1, \dots, N \tag{16}$$

From (2) and (3) we define

$$[\langle u, v \rangle(t_{n+1})]_l^+(\alpha) - [\langle u, v \rangle(t_n)]_l^+(\alpha) = \sum_{i=1}^{i=3} w_i [k_i]_l^+(\alpha) \tag{17}$$

$$[\langle u, v \rangle(t_{n+1})]_r^+(\alpha) - [\langle u, v \rangle(t_n)]_r^+(\alpha) = \sum_{i=1}^{i=3} w_i [k_i]_r^+(\alpha) \tag{18}$$

$$[\langle u, v \rangle(t_{n+1})]_l^-(\alpha) - [\langle u, v \rangle(t_n)]_l^-(\alpha) = \sum_{i=1}^{i=3} w_i [k_i]_l^-(\alpha) \tag{19}$$

$$[\langle u, v \rangle(t_{n+1})]_r^-(\alpha) - [\langle u, v \rangle(t_n)]_r^-(\alpha) = \sum_{i=1}^{i=3} w_i [k_i]_r^-(\alpha) \tag{20}$$

where the w_i 's are constants and

$$\left\{ \begin{aligned} [k_i]_\alpha &= [k_i]_l^+(\alpha), [k_i]_r^+(\alpha), \quad i = 1, 2, 3 \\ [k_i]^\alpha &= [k_i]_l^-(\alpha), [k_i]_r^-(\alpha) \\ [k_i]_l^+(\alpha) &= hP(t_n + c_i h, [\langle u, v \rangle(t_n)]_l^+ + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+ + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_r^+(\alpha)) \\ [k_i]_r^+(\alpha) &= hQ(t_n + c_i h, [\langle u, v \rangle(t_n)]_l^+ + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+ + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_r^+(\alpha)) \\ [k_i]_l^-(\alpha) &= hR(t_n + c_i h, [\langle u, v \rangle(t_n)]_l^- + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^- + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_r^-(\alpha)) \\ [k_i]_r^-(\alpha) &= hS(t_n + c_i h, [\langle u, v \rangle(t_n)]_l^- + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^- + \sum_{j=1}^{i-1} \beta_{ij} [k_j]_r^-(\alpha)) \end{aligned} \right.$$

Assuming the following Runge-Kutta Nyström method with three slopes:

$$[\langle u, v \rangle(t_{n+1})]_l^+(\alpha) = [\langle u, v \rangle(t_n)]_l^+(\alpha) + \frac{1}{8}(2[k_1]_l^+(\alpha) + 3[k_2]_l^+(\alpha) + 3[k_3]_l^+(\alpha)) \tag{21}$$

where

$$\left\{ \begin{aligned} [k_1]_l^+(\alpha) &= hP(t_n, [\langle u, v \rangle(t_n)]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha)) \\ [k_2]_l^+(\alpha) &= hP(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^+(\alpha) + \frac{2}{3}[k_1]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_1]_r^+(\alpha)) \\ [k_3]_l^+(\alpha) &= hP(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^+(\alpha) + \frac{2}{3}[k_2]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_2]_r^+(\alpha)) \end{aligned} \right.$$

and

$$[\langle u, v \rangle(t_{n+1})]_r^+(\alpha) = [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{1}{8}(2[k_1]_r^+(\alpha) + 3[k_2]_r^+(\alpha) + 3[k_3]_r^+(\alpha)) \tag{22}$$

where

$$\begin{cases} [k_1]_r^+(\alpha) = hQ(t_n, [\langle u, v \rangle(t_n)]_l^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha)) \\ [k_2]_r^+(\alpha) = hQ(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^+(\alpha) + \frac{2}{3}[k_1]_r^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_1]_r^+(\alpha)) \\ [k_3]_r^+(\alpha) = hQ(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^+(\alpha) + \frac{2}{3}[k_2]_r^+(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_2]_r^+(\alpha)) \end{cases}$$

and

$$[\langle u, v \rangle(t_{n+1})]_l^-(\alpha) = [\langle u, v \rangle(t_n)]_l^-(\alpha) + \frac{1}{8}(2[k_1]_l^-(\alpha) + 3[k_2]_l^-(\alpha) + 3[k_3]_l^-(\alpha)) \tag{23}$$

where

$$\begin{cases} [k_1]_l^-(\alpha) = hR(t_n, [\langle u, v \rangle(t_n)]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^-(\alpha)) \\ [k_2]_l^-(\alpha) = hR(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^-(\alpha) + \frac{2}{3}[k_1]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_1]_r^-(\alpha)) \\ [k_3]_l^-(\alpha) = hR(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^-(\alpha) + \frac{2}{3}[k_2]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_2]_r^-(\alpha)) \end{cases}$$

and

$$[\langle u, v \rangle(t_{n+1})]_r^-(\alpha) = [\langle u, v \rangle(t_n)]_r^-(\alpha) + \frac{1}{8}(2[k_1]_r^-(\alpha) + 3[k_2]_r^-(\alpha) + 3[k_3]_r^-(\alpha)) \tag{24}$$

where

$$\begin{cases} [k_1]_r^-(\alpha) = hS(t_n, [\langle u, v \rangle(t_n)]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^-(\alpha)) \\ [k_2]_r^-(\alpha) = hS(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^-(\alpha) + \frac{2}{3}[k_1]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_1]_r^-(\alpha)) \\ [k_3]_r^-(\alpha) = hS(t_n + \frac{2}{3}h, [\langle u, v \rangle(t_n)]_l^-(\alpha) + \frac{2}{3}[k_2]_l^-(\alpha), [\langle u, v \rangle(t_n)]_r^+(\alpha) + \frac{2}{3}[k_2]_r^-(\alpha)) \end{cases}$$

Let $P(t, z^+, w^+)$, $Q(t, z^+, w^+)$, $R(t, z^-, w^-)$ and $S(t, z^-, w^-)$ be the functions of (11), where z^+ , w^+ , z^- and w^- are the constants and $z^+ \leq w^+$ and $z^- \leq w^-$.

The domain of P and Q is

$$M_1 = \{(t, z^+, w^+) \mid t_0 \leq t \leq T, \quad \infty < z^+ \leq w^+, \quad -\infty < w^+ < +\infty\}$$

and the domain of R and S is

$$M_2 = \{(t, z^-, w^-) \mid t_0 \leq t \leq T, \quad \infty < z^- \leq w^-, \quad -\infty < w^- < +\infty\}$$

where $M_1 \subseteq M_2$

Theorem 5 Let $P(t, z^+, w^+)$, $Q(t, z^+, w^+)$ belong to $C^3(M_1)$ and $R(t, z^-, w^-)$, $S(t, z^-, w^-)$ belong to $C^3(M_2)$ and the partial derivatives of P, Q and R, S be bounded over M_1 and M_2 respectively. Then, for arbitrarily fixed $0 \leq \alpha \leq 1$, the numerical solutions of (21), (22), (23) and (24) converge to the exact solutions $[\langle U, V \rangle(t)]_l^+(\alpha)$, $[\langle U, V \rangle(t)]_r^+(\alpha)$, $[\langle U, V \rangle(t)]_l^-(\alpha)$ and $[\langle U, V \rangle(t)]_r^-(\alpha)$ uniformly in t .

Proof see [13].

5 Example

Consider the intuitionistic fuzzy initial value problem:

$$\begin{cases} \langle u, v \rangle'(t) = \langle u, v \rangle(t) \text{ for all } t \in [0, T] \\ \langle u, v \rangle_0 = ((\alpha - 1.0013, 1.0013 - \alpha), (-1.58\alpha, 1.58\alpha)) \end{cases} \quad (25)$$

Then, we have the following parametrized differential system:

$$\begin{cases} [\langle u, v \rangle(t)]_l^+(\alpha) = (\alpha - 1.0013) \exp(t) \\ [\langle u, v \rangle(t)]_r^+(\alpha) = (1.0013 - \alpha) \exp(t) \\ [\langle u, v \rangle(t)]_l^-(\alpha) = 1.58(\alpha - 1.0013) \exp(t) \\ [\langle u, v \rangle(t)]_r^-(\alpha) = 1.58(1.0013 - \alpha) \exp(t) \end{cases}$$

Therefore the exact solutions is given by

$$\begin{aligned} [\langle U, V \rangle(t)]_\alpha &= [(\alpha - 1.0013) \exp(t), (1.0013 - \alpha) \exp(t)] \\ [\langle U, V \rangle(t)]^\alpha &= [1.58(\alpha - 1.0013) \exp(t), 1.58(1.0013 - \alpha) \exp(t)] \end{aligned}$$

which at $t = 1$ are

$$\begin{aligned} [\langle U, V \rangle(1)]_\alpha &= [(\alpha - 1.0013) \exp(1), (1.0013 - \alpha) \exp(1)] \\ [\langle U, V \rangle(1)]^\alpha &= [1.58(\alpha - 1.0013) \exp(1), 1.58(1.0013 - \alpha) \exp(1)] \end{aligned}$$

Comparison of results of the third order Runge-Kutta Nyström method and the Euler and 2nd-order Taylor method in [13] for $N = 20$ and $t = 1$:

α	Exact		RK-Nyström	
	$(\{(u, v)\}_I^+, \{(u, v)\}_I^+)$	$(\{(u, v)\}_I^-, \{(u, v)\}_I^-)$	$(\{(U, V)\}_I^+, \{(U, V)\}_I^+)$	$(\{(U, V)\}_I^-, \{(U, V)\}_I^-)$
0	(-2.72181559, 2.72181559)	(0,0)	(-2.72180197, 2.72180197)	(0,0)
0.2	(-2.17815922, 2.17815922)	(-0.85897705, 0.85897705)	(-2.17814832, 2.17814832)	(-0.85897275, 0.85897275)
0.4	(-1.63450286, 1.63450286)	(-1.71795411, 1.71795411)	(-1.63449468, 1.63449468)	(-1.71794551, 1.71794551)
0.6	(-1.09084649, 1.09084649)	(-2.57693117, 2.57693117)	(-1.09084103, 1.09084103)	(-2.57691827, 2.57691827)
0.8	(-0.54719013, 0.54719013)	(-3.43590823, 3.43590823)	(-0.54718739, 0.54718739)	(-3.43589103, 3.43589103)
1	(-0.00353376, 0.00353376)	(-4.29488528, 4.29488528)	(-0.00353374, 0.00353374)	(-4.29486379, 4.29486379)

α	Euler		2nd-order taylor	
	$(\{(u, v)\}_I^+, \{(u, v)\}_I^+)$	$(\{(u, v)\}_I^-, \{(u, v)\}_I^-)$	$(\{(u, v)\}_I^+, \{(u, v)\}_I^+)$	$(\{(u, v)\}_I^-, \{(u, v)\}_I^-)$
0	(-2.65674699, 2.65674699)	(0,0)	(-2.72072340, 2.72072340)	(0,0)
0.2	(-2.12608745, 2.12608745)	(-0.83844207, 0.83844207)	(-2.17728519, 2.17728519)	(-0.85863237, 0.85863237)
0.4	(-1.59542791, 1.59542791)	(-1.67688414, 1.67688414)	(-1.63384698, 1.63384698)	(-1.71726474, 1.71726474)
0.6	(-1.06476836, 1.06476836)	(-2.51532622, 2.51532622)	(-1.09040877, 1.09040877)	(-2.57589711, 2.57589711)
0.8	(-0.53410882, 0.53410882)	(-3.35376829, 3.35376829)	(-0.54697055, -0.54697055)	(-3.43452949, 3.43452949)
1	(-0.00344928, 0.00344928)	(-4.19221037, 4.19221037)	(-0.00353234, 0.00353234)	(-4.29316186, 4.29316186)

α	Error in euler	Error in 2nd-order taylor	Error in RK-Nyström
0	3.2534×10^{-2}	5.4609×10^{-4}	6.8103×10^{-6}
0.2	3.6303×10^{-2}	6.0936×10^{-4}	7.5993×10^{-6}
0.4	4.0072×10^{-2}	6.7262×10^{-4}	8.3882×10^{-6}
0.6	4.3841×10^{-2}	7.3589×10^{-4}	9.1772×10^{-6}
0.8	4.7610×10^{-2}	7.9915×10^{-4}	9.9662×10^{-6}
1	5.1379×10^{-2}	8.6242×10^{-4}	1.0755×10^{-5}

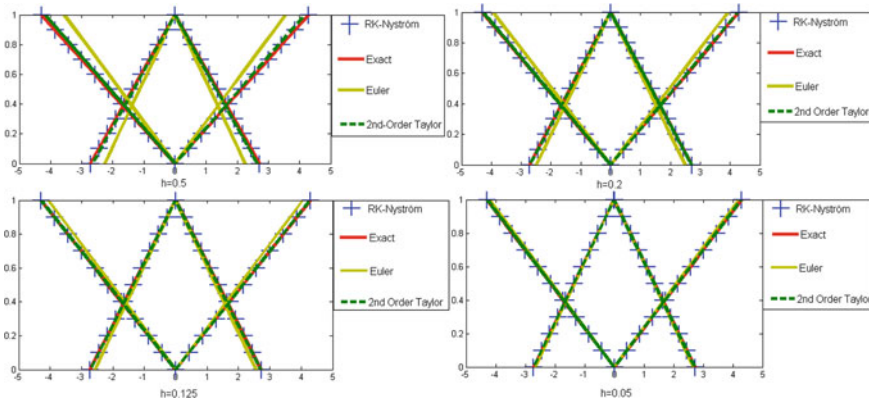


Fig. 1 The exact and approximate solutions

The exact and approximate solutions obtained by the Euler method and 2nd-order Taylor method and the third order Runge-Kutta Nyström method are compared and plotted at $t = 1$ in Fig. 1.

The error between the Euler and 2nd-order Taylor method and 3rd-order Runge-Kutta Nyström method is plotted in Fig. 2:

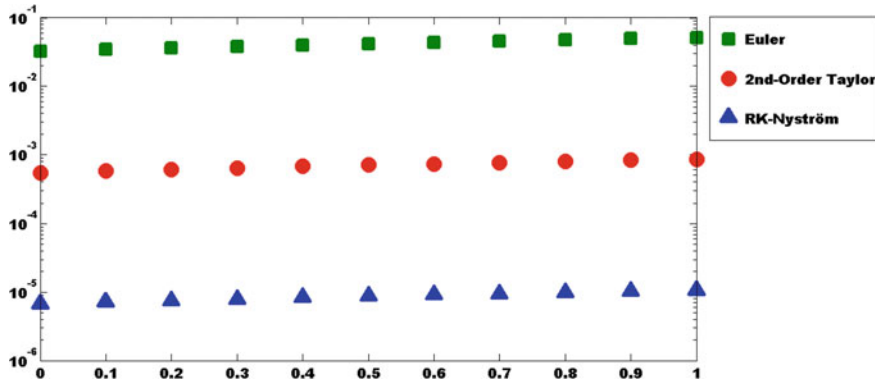


Fig. 2 $h = 0.05$

6 Conclusion

In this paper we have applied iterative solution of Runge-Kutta Nyström method for numerical solution of intuitionistic fuzzy differential equations. It is shown that the convergence order of the method proposed is $O(h^3)$, while the Euler method and 2nd-order Taylor method from [13] converges with the rate $O(h)$ and $O(h^2)$ respectively. In comparison with [13] the solutions of the example have the higher accuracy in this paper.

References

1. Adak, A.K., Bhowmik, M., Pal, M.: Intuitionistic fuzzy block matrix and its some properties. *Ann. Pure Appl. Math.* **1**(1), 13–31 (2012)
2. Atanassov, K.T.: Intuitionistic fuzzy sets. VII ITKR's session, Sofia (deposited in Central Science and Technical Library of the Bulgarian Academy of Sciences 1697/84) (1983)
3. Atanassov, K.T.: Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **20**, 87–96 (1986)
4. Atanassov, K.T., Gargov, G.: Interval-valued intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **31**(3), 43–49 (1989)
5. Atanassov, K.T.: More on intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **33**(1), 37–45 (1989)
6. Atanassov, K.T.: Operators over interval valued intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **64**(2), 159–174 (1994)
7. Atanassov, K.T., Gargov, G.: Elements of intuitionistic fuzzy logic. Part I, *Fuzzy Sets Syst.* **95**(1), 39–52 (1998)
8. Atanassov, K.T.: *Intuitionistic Fuzzy Sets*. Physica-Verlag, Heidelberg, New York (1999)
9. Atanassov, K.T.: Two theorems for Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **110**, 267–269 (2000)
10. Buhaesku, T.: On the convexity of intuitionistic fuzzy sets. In: *Itinerant Seminar on Functional Equations, Approximation and Convexity*, pp. 137–144. Cluj-Napoca (1988)
11. Buhaesku, T.: Some observations on intuitionistic fuzzy relations. In: *Itinerant Seminar of Functional Equations, Approximation and Convexity*, pp. 111–118 (1989)

12. Ban, A.I.: Nearest interval approximation of an intuitionistic fuzzy number. In: Computational Intelligence, Theory and Applications, pp. 229–240. Springer-Verlag, Berlin, Heidelberg (2006)
13. Ben Amma, B., Melliani, S., Chadli, L.S.: Numerical solution of intuitionistic fuzzy differential equations by Euler and Taylor methods. *Notes Intuit.Istic Fuzzy Sets*, **22**(2), 71–86 (2016)
14. Ben Amma, B., Melliani, S., Chadli, L.S.: Numerical solution of intuitionistic fuzzy differential equations by Adams three order predictor-corrector method. *Notes on Intuitionistic Fuzzy Sets* **22**(3), 47–69 (2016)
15. Ben Amma, B., Chadli, L.S.: Numerical solution of intuitionistic fuzzy differential equations by Runge-Kutta Method of order four. *Notes on Intuitionistic Fuzzy Sets* **22**(4), 42–52 (2016)
16. Ben Amma, B., Melliani, S., Chadli, L.S.: The Cauchy Problem Of Intuitionistic Fuzzy Differential Equations. *Notes on Intuitionistic Fuzzy Sets*, **24**(1), 37–47 (2018)
17. Ben Amma, B., Melliani, S., Chadli, L.S.: Intuitionistic Fuzzy Functional Differential Equations, Fuzzy Logic in Intelligent System Design: Theory and Applications, Ed. pp. 335–357. Springer International Publishing, Cham (2018)
18. Ben Amma B., Melliani S., Chadli L.S. (2019) A Fourth order runge-kutta gill method for the numerical solution of intuitionistic fuzzy differential equations. In: Melliani S., Castillo O. (eds.) Recent Advances in Intuitionistic Fuzzy Logic Systems. Studies in Fuzziness and Soft Computing, vol. 372. Springer, Cham
19. Cornelis, C., Deschrijver, G., Kerre, E.E.: Implication in intuitionistic fuzzy and interval-valued fuzzy set theory: construction, application. *Int. J. Approx. Reason.* **35**, 55–95 (2004)
20. De, S.K., Biswas, R., Roy, A.R.: An application of intuitionistic fuzzy sets in medical diagnosis. *Fuzzy Sets Syst.* **117**, 209–213 (2001)
21. Deschrijver, G., Kerre, E.E.: On the relationship between intuitionistic fuzzy sets and some other extensions of fuzzy set theory. *J. Fuzzy Math.* **10**(3), 711–724 (2002)
22. Gerstenkorn, T., Manko, J.: Correlation of intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **44**, 39–43 (1991)
23. Kharal, A.: Homeopathic drug selection using intuitionistic fuzzy sets. *Homeopathy* **98**, 35–39 (2009)
24. Li, D.F., Cheng, C.T.: New similarity measures of intuitionistic fuzzy sets and application to pattern recognitions. *Pattern Recognit. Lett.* **23**, 221–225 (2002)
25. Li, D.F.: Multiattribute decision making models and methods using intuitionistic fuzzy sets. *J. Comput. Syst. Sci.* **70**, 73–85 (2005)
26. Melliani, S., Chadli, L.S.: Intuitionistic fuzzy differential equation. *Notes on Intuitionistic Fuzzy Sets* **6**, 37–41 (2000)
27. Mahapatra, G.S., Roy, T.K.: Reliability evaluation using triangular intuitionistic fuzzy numbers arithmetic operations. *Proc. World Acad. Sci., Eng. Technol.* **38**, 587–595 (2009)
28. Melliani, S., Elomari, M., Chadli, L.S., Ettoussi, R.: Intuitionistic Fuzzy Metric Space. *Notes on Intuitionistic Fuzzy sets* **21**(1), 43–53 (2015)
29. Melliani, S., Elomari, M., Chadli, L.S., Ettoussi, R.: Intuitionistic fuzzy fractional equation. *Notes on Intuitionistic Fuzzy sets* **21**(4), 76–89 (2015)
30. Melliani, S., Elomari, M., Atraoui, M., Chadli, L.S.: Intuitionistic fuzzy differential equation with nonlocal condition. *Notes on Intuitionistic Fuzzy sets* **21**(4), 58–68 (2015)
31. Melliani, S., Atti, H., Ben Amma, B., Chadli, L.S.: Solution of n-th order intuitionistic fuzzy differential equation by variational iteration method, *Notes on Intuitionistic Fuzzy sets*, **24**(3), 92–105 (2018)
32. Nikolova, M., Nikolov, N., Cornelis, C., Deschrijver, G.: Survey of the research on intuitionistic fuzzy sets. *Adv. Stud. Contempor. Math.* **4**(2), 127–157 (2002)
33. Szmjdt, E., Kacprzyk, J.: Distances between intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **114**(3), 505–518 (2000)
34. Shu, M.H., Cheng, C.H., Chang, J.R.: Using intuitionistic fuzzy sets for fault-tree analysis on printed circuit board assembly. *Microelectron. Reliab.* **46**(12), 2139–2148 (2006)
35. Wang, Z., Li, K.W., Wang, W.: An approach to multiattribute decision making with interval-valued intuitionistic fuzzy assessments and incomplete weights. *Inf. Sci.* **179**(17), 3026–3040 (2009)

36. Ye, J.: Multicriteria fuzzy decision-making method based on a novel accuracy function under interval valued intuitionistic fuzzy environment. *Expert Syst. Applicat.* **36**, 6899–6902 (2009)
37. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965)

Intuitionistic Fuzzy Linear Systems



Hafida Atti, Bouchra Ben Amma, Said Melliani and S. Chadli

Abstract In this work we present a method for solving intuitionistic fuzzy linear systems by four crisp linear systems. Also necessary and sufficient conditions for existence of intuitionistic fuzzy solution are given. A numerical examples are illustrated the efficiency of the proposed method.

Keywords Intuitionistic fuzzy linear systems · Intuitionistic fuzzy numbers · Nonnegative matrix

1 Introduction

Fuzzy sets (FS) introduced by Zadeh [11] has showed meaningful applications in many fields of study. The idea of fuzzy set is welcome because it handles uncertainty and vagueness which Cantorian set could not address. In fuzzy set theory, the membership of an element to a fuzzy set is a single value between zero and one. However in reality, it may not always be true that the degree of non-membership of an element in a fuzzy set is equal to 1 minus the membership degree because there may be some hesitation degree. Therefore, a generalization of fuzzy sets was proposed by Atanassov [5, 6] as intuitionistic fuzzy sets (IFS) which incorporate

H. Atti · B. Ben Amma (✉) · S. Melliani · S. Chadli
Laboratory of Applied Mathematics and Scientific Computing, Faculty of Sciences and Technologies, Sultan Moulay Slimane University, BP 523, 23000 Beni Mellal, Morocco
e-mail: bouchrabenamma@gmail.com

H. Atti
e-mail: hafida.atti@gmail.com

S. Melliani
e-mail: s.melliani@yahoo.fr

S. Chadli
e-mail: sa.chadli@yahoo.fr

© Springer Nature Switzerland AG 2020
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_12

the degree of hesitation called hesitation margin (and is defined as 1 minus the sum of membership and non-membership degrees respectively). The notion of defining intuitionistic fuzzy set as generalized fuzzy set is quite interesting and useful in many application areas.

Systems of linear equations have so many applications in various fields of mathematical, physical and engineering sciences such as traffic flow, circuit analysis, heat transport, structural mechanics, fluid flow etc. In most of the problems, we generally work with given approximate datas. To avoid these type of errors we can represent the given data as fuzzy and more generally intuitionistic fuzzy numbers rather than crisp numbers. Fuzzy linear systems are the linear systems whose parameters are all or partially represented by fuzzy numbers. A general model for solving a fuzzy linear system whose coefficient matrix is crisp and the right hand side column is an arbitrary fuzzy number was first proposed by Friedman et al. [7]. They have used the parametric form of fuzzy numbers and replace the original $n \times n$ fuzzy system by a $2n \times 2n$ crisp system. Fuzzy linear system has been studied by several authors [1, 2, 4, 8], in this paper we present a method for solving $n \times n$ intuitionistic fuzzy linear system whose coefficients matrix is crisp and the right-hand side column is an arbitrary intuitionistic fuzzy number vector. For solving $n \times n$ intuitionistic fuzzy linear system we solve four $n \times n$ crisp function linear systems witch we inspired by previous definitions of [3].

The remainder of the paper is arranged as follows : In Sect. 2, we give some basic preliminaries which will be used throughout this paper. In Sect. 3 we have discussed the detailed solution procedure of system of intuitionistic fuzzy linear equations. An numerical examples are provided to illustrate the efficiency of the method in Sect. 4 and finally some conclusions in Sect. 5.

2 Preliminaries

We denote by

$$\text{IF}_1 = \text{IF}(\mathbb{R}) = \{ \langle u, v \rangle : \mathbb{R} \rightarrow [0, 1]^2, |\forall x \in \mathbb{R} 0 \leq u(x) + v(x) \leq 1 \}$$

An element $\langle u, v \rangle$ of IF_1 is said an intuitionistic fuzzy number if it satisfies the following conditions

- (i) $\langle u, v \rangle$ is normal i.e there exists $x_0, x_1 \in \mathbb{R}$ such that $u(x_0) = 1$ and $v(x_1) = 1$.
- (ii) u is fuzzy convex and v is fuzzy concave.
- (iii) u is upper semi-continuous and v is lower semi-continuous
- (iv) $\text{supp} \langle u, v \rangle = \text{cl}\{x \in \mathbb{R} : |v(x) < 1\}$ is bounded.

So we denote the collection of all intuitionistic fuzzy number by IF_1 .

A triangular intuitionistic fuzzy number (TIFN) $\langle u, v \rangle$ is an intuitionistic fuzzy set in \mathbb{R} with the following membership function u and non-membership function v :

$$u(y) = \begin{cases} \frac{y - a_1}{a_2 - a_1} & \text{if } a_1 \leq y \leq a_2 \\ \frac{a_3 - y}{a_3 - a_2} & \text{if } a_2 \leq y \leq a_3, \\ 0 & \text{otherwise} \end{cases}$$

$$v(y) = \begin{cases} \frac{a_2 - y}{a_2 - a'_1} & \text{if } a'_1 \leq y \leq a_2 \\ \frac{y - a_2}{a'_3 - a_2} & \text{if } a_2 \leq y \leq a'_3, \\ 1 & \text{otherwise.} \end{cases}$$

where $a'_1 \leq a_1 \leq a_2 \leq a_3 \leq a'_3$

For $\alpha \in [0, 1]$ and $\langle u, v \rangle \in \text{IF}_1$, the upper and lower α -cuts of $\langle u, v \rangle$ are defined by

$$[\langle u, v \rangle]^\alpha = \{y \in \mathbb{R} : v(y) \leq 1 - \alpha\}$$

and

$$[\langle u, v \rangle]_\alpha = \{y \in \mathbb{R} : u(y) \geq \alpha\}$$

We define $0_{(1,0)} \in \text{IF}_1$ as

$$0_{(1,0)}(t) = \begin{cases} (1, 0) & t = 0 \\ (0, 1) & t \neq 0 \end{cases}$$

For $\langle u, v \rangle, \langle z, w \rangle \in \text{IF}_1$ and $\lambda \in \mathbb{R}$, the addition and scaler-multiplication are defined as follows

$$\begin{aligned} [\langle u, v \rangle \oplus \langle z, w \rangle]^\alpha &= [\langle u, v \rangle]^\alpha + [\langle z, w \rangle]^\alpha, & [\lambda \langle z, w \rangle]^\alpha &= \lambda [\langle z, w \rangle]^\alpha \\ [\langle u, v \rangle \oplus \langle z, w \rangle]_\alpha &= [\langle u, v \rangle]_\alpha + [\langle z, w \rangle]_\alpha, & [\lambda \langle z, w \rangle]_\alpha &= \lambda [\langle z, w \rangle]_\alpha \end{aligned}$$

Definition 1 [9] An intuitionistic fuzzy number x in parametric form is a pair $x = ((\underline{x}^+, \overline{x}^+), (\underline{x}^-, \overline{x}^-))$ of functions $\underline{x}^-(\alpha), \overline{x}^-(\alpha), \underline{x}^+(\alpha)$ and $\overline{x}^+(\alpha)$, which satisfies the following requirements:

1. $\underline{x}^+(\alpha)$ is a bounded monotonic increasing left continuous function,
2. $\overline{x}^+(\alpha)$ is a bounded monotonic decreasing left continuous function,
3. $\underline{x}^-(\alpha)$ is a bounded monotonic increasing left continuous function,
4. $\overline{x}^-(\alpha)$ is a bounded monotonic decreasing left continuous function,
5. $\underline{x}^-(\alpha) \leq \overline{x}^-(\alpha)$ and $\underline{x}^+(\alpha) \leq \overline{x}^+(\alpha)$, for all $0 \leq r \leq 1$.

This TIFN is denoted by $x = \langle a_1, a_2, a_3; a'_1, a_2, a'_3 \rangle$.
 Its parametric form is

$$\begin{aligned} \underline{x}^+(\alpha) &= a_1 + \alpha(a_2 - a_1), & \bar{x}^+(\alpha) &= a_3 - \alpha(a_3 - a_2) \\ \underline{x}^-(\alpha) &= a'_1 + \alpha(a_2 - a'_1), & \bar{x}^-(\alpha) &= a'_3 - \alpha(a'_3 - a_2) \end{aligned}$$

3 Intuitionistic Fuzzy Linear System

Definition 2 The $n \times n$ linear system of equations

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = y_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = y_2, \\ \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = y_n, \end{cases} \tag{1}$$

where the coefficients matrix $A = (a_{ij}), 1 \leq i \leq n, 1 \leq j \leq n$ is a crisp $n \times n$ matrix and $y_i \in \text{IF}_1, 1 \leq i \leq n$ and the unknown $x_j \in \text{IF}_1, 1 \leq j \leq n$, is called an intuitionistic fuzzy linear system (IFLS).

Definition 3 An intuitionistic fuzzy number vector $(\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n)^t$ given by $\tilde{x}_j = (\underline{x}_j^+(\alpha), \bar{x}_j^+(\alpha), \underline{x}_j^-(r), \bar{x}_j^-(\alpha)), 1 \leq j \leq n, 0 \leq \alpha \leq 1$, is called solution of (1) if :

$$\begin{aligned} \underline{\sum_{j=1}^n a_{ij}x_j^+} &= \underline{\sum_{j=1}^n a_{ij}\underline{x}_j^+} = \underline{y_i^+}, \quad i = 1, 2, \dots, n \\ \overline{\sum_{j=1}^n a_{ij}x_j^+} &= \overline{\sum_{j=1}^n a_{ij}\overline{x}_j^+} = \overline{y_i^+}, \quad i = 1, 2, \dots, n \\ \underline{\sum_{j=1}^n a_{ij}x_j^-} &= \underline{\sum_{j=1}^n a_{ij}\underline{x}_j^-} = \underline{y_i^-}, \quad i = 1, 2, \dots, n \\ \overline{\sum_{j=1}^n a_{ij}x_j^-} &= \overline{\sum_{j=1}^n a_{ij}\overline{x}_j^-} = \overline{y_i^-}, \quad i = 1, 2, \dots, n \end{aligned}$$

if, for a particular i , $a_{ij} > 0$, $1 \leq j \leq n$, we get

$$\begin{aligned} \sum_{j=1}^n a_{ij}x_j^+ &= \sum_{j=1}^n \overline{a_{ij}x_j^+} = \sum_{j=1}^n a_{ij}\overline{x_j^+} = \underline{y_i^+}, \\ \overline{\sum_{j=1}^n a_{ij}x_j^+} &= \sum_{j=1}^n \overline{a_{ij}x_j^+} = \sum_{j=1}^n a_{ij}\overline{x_j^+} = \overline{y_i^+}, \\ \sum_{j=1}^n a_{ij}x_j^- &= \sum_{j=1}^n \overline{a_{ij}x_j^-} = \sum_{j=1}^n a_{ij}\overline{x_j^-} = \underline{y_i^-}, \\ \overline{\sum_{j=1}^n a_{ij}x_j^-} &= \sum_{j=1}^n \overline{a_{ij}x_j^-} = \sum_{j=1}^n a_{ij}\overline{x_j^-} = \overline{y_i^-}. \end{aligned}$$

$$(b_{i1} - c_{i1})(\underline{x_1^+}, \overline{x_1^+}, \underline{x_1^-}, \overline{x_1^-}) + \dots + (b_{in} - c_{in})(\underline{x_n^+}, \overline{x_n^+}, \underline{x_n^-}, \overline{x_n^-}) = (\underline{y_i^+}, \overline{y_i^+}, \underline{y_i^-}, \overline{y_i^-})$$

hence

$$b_{i1}\underline{x_1^+} - c_{i1}\overline{x_1^+} + b_{i2}\underline{x_2^+} - c_{i2}\overline{x_2^+} + \dots + b_{in}\underline{x_n^+} - c_{in}\overline{x_n^+} = \underline{y_i^+} \tag{2}$$

$$b_{i1}\overline{x_1^+} - c_{i1}\underline{x_1^+} + b_{i2}\overline{x_2^+} - c_{i2}\underline{x_2^+} + \dots + b_{in}\overline{x_n^+} - c_{in}\underline{x_n^+} = \overline{y_i^+} \tag{3}$$

$$b_{i1}\underline{x_1^-} - c_{i1}\overline{x_1^-} + b_{i2}\underline{x_2^-} - c_{i2}\overline{x_2^-} + \dots + b_{in}\underline{x_n^-} - c_{in}\overline{x_n^-} = \underline{y_i^-} \tag{4}$$

$$b_{i1}\overline{x_1^-} - c_{i1}\underline{x_1^-} + b_{i2}\overline{x_2^-} - c_{i2}\underline{x_2^-} + \dots + b_{in}\overline{x_n^-} - c_{in}\underline{x_n^-} = \overline{y_i^-} \tag{5}$$

Let matrix B contains the positive entries of A and matrix C contains the absolute value of the negative entries of A , hence $A = B - C$.

Now using matrix notation for (1) we get $AX = Y$ or $(B - C)X = Y$.

Theorem 1 *Let X be an intuitionistic fuzzy solution of system (1) where A is non-singular matrix and Y is an intuitionistic fuzzy number vector. Then $AE^+ = F^+$ and $AE^- = F^-$.*

Proof By addition of (2) and (3) we have:

$$(b_{i1} - c_{i1})(\underline{x_1^+} + \overline{x_1^+}) + \dots + (b_{in} - c_{in})(\underline{x_n^+} + \overline{x_n^+}) = (\underline{y_i^+} + \overline{y_i^+})$$

and same for (4) and (5) we have

$$(b_{i1} - c_{i1})(\underline{x_1^-} + \overline{x_1^-}) + \dots + (b_{in} - c_{in})(\underline{x_n^-} + \overline{x_n^-}) = (\underline{y_i^-} + \overline{y_i^-})$$

and hence

$$\sum_{j=1}^n (b_{ij}(\overline{x_j^+} + \underline{x_j^+})) - \sum_{j=1}^n (c_{ij}(\underline{x_j^+} + \overline{x_j^+})) = (\overline{y_i^+} + \underline{y_i^+})$$

and

$$\sum_{j=1}^n (b_{ij} \overline{x_j} + \underline{x_j}) - \sum_{j=1}^n (c_{ij} \overline{x_j} + \underline{x_j}) = \overline{y_i} + \underline{y_i}$$

Let

$$E^+ = (e_1^+, e_2^+, \dots, e_n^+)^t, \quad F^+ = (f_1^+, f_2^+, \dots, f_n^+)^t$$

and

$$E^- = (e_1^-, e_2^-, \dots, e_n^-)^t, \quad F^- = (f_1^-, f_2^-, \dots, f_n^-)^t$$

where $e_i^+ = (\overline{x_i} + \underline{x_i}), f_i^+ = (\overline{y_i} + \underline{y_i})$ and $e_i^- = (\overline{x_i} - \underline{x_i}), f_i^- = (\overline{y_i} - \underline{y_i})$.
So

$$AE^+ = F^+$$

$$AE^- = F^-$$

Remark 1 Let X be an intuitionistic fuzzy solution of system (1) where A is nonsingular matrix and Y is an intuitionistic fuzzy number. If Y is symmetric intuitionistic fuzzy number vector then X is symmetric intuitionistic fuzzy number vector.

From (2) and (3) we get

$$b_{i1}(\overline{x_1} - \underline{x_1}) + \dots + b_{in}(\overline{x_n} - \underline{x_n}) - c_{i1}(\overline{x_1} - \underline{x_1}) - \dots - c_{in}(\overline{x_n} - \underline{x_n}) = \overline{y_i} - \underline{y_i}$$

and from (4) and (5) we get

$$b_{i1}(\overline{x_1} - \underline{x_1}) + \dots + b_{in}(\overline{x_n} - \underline{x_n}) - c_{i1}(\overline{x_1} - \underline{x_1}) - \dots - c_{in}(\overline{x_n} - \underline{x_n}) = \overline{y_i} - \underline{y_i}$$

and hence

$$\sum_{j=1}^n b_{ij}(\overline{x_j} - \underline{x_j}) + \sum_{j=1}^n c_{ij}(\overline{x_j} - \underline{x_j}) = \overline{y_i} - \underline{y_i}$$

and

$$\sum_{j=1}^n b_{ij}(\overline{x_j} - \underline{x_j}) + \sum_{j=1}^n c_{ij}(\overline{x_j} - \underline{x_j}) = \overline{y_i} - \underline{y_i}$$

if we pose $w_j^+ = \overline{x_j} - \underline{x_j}, v_j^+ = \overline{y_i} - \underline{y_i}, w_j^- = \overline{x_j} - \underline{x_j}$ and $v_j^- = \overline{y_i} - \underline{y_i}$
we get

$$(B + C)W^+ = V^+$$

$$(B + C)W^- = V^-$$

where $W^+ = (w_1^+, w_2^+, \dots, w_n^+)^t$, $V^+ = (v_1^+, v_2^+, \dots, v_n^+)^t$, $W^- = (w_1^-, w_2^-, \dots, w_n^-)^t$, $V^- = (v_1^-, v_2^-, \dots, v_n^-)^t$ and $A = B - C$.

For solving system (1), we must solve the following systems:

$$\begin{cases} (B + C)W^+ = V^+, \\ (B - C)E^+ = F^+. \end{cases}$$

and

$$\begin{cases} (B + C)W^- = V^-, \\ (B - C)E^- = F^-. \end{cases}$$

Then after solving, for each $1 \leq i \leq n$ we take,

$$\begin{cases} \underline{x}_i^+ = \frac{1}{2}e_i^+ - \frac{1}{2}w_i^+, \\ \overline{x}_i^+ = \frac{1}{2}e_i^+ + \frac{1}{2}w_i^+, \\ \underline{x}_i^- = \frac{1}{2}e_i^- - \frac{1}{2}w_i^-, \\ \overline{x}_i^- = \frac{1}{2}e_i^- + \frac{1}{2}w_i^-. \end{cases}$$

Equation (1) has a unique solution if and only if $(B + C)$ and $(B - C)$ are both nonsingular.

Theorem 2 *If $(B + C)$ and $(B - C)$ are both nonsingular and y is an arbitrary intuitionistic fuzzy number vector. Then the system (1) has a unique intuitionistic fuzzy solution if $(B + C)^{-1}$ is a nonnegative matrix.*

Proof The Eq. (1) has a unique solution if and only if $(B + C)$ and $(B - C)$ are both nonsingular. So the system (1) has a unique solution but may still not be an intuitionistic fuzzy number vector.

we have

$$\begin{aligned} (B + C)W^+ &= V^+ \\ (B + C)W^- &= V^- \end{aligned}$$

Since $(B + C)$ is nonsingular

$$\begin{aligned} W^+ &= (B + C)^{-1}V^+ \\ W^- &= (B + C)^{-1}V^- \end{aligned}$$

Since $(\overline{y}_i^+ - \underline{y}_i^+) \geq 0$, $(\overline{y}_i^- - \underline{y}_i^-) \geq 0$ for $1 \leq i \leq n$ and $(B + C)^{-1} \geq 0$ therefore $\underline{x}_j^+ - \overline{x}_j^+ \geq 0$ and $\underline{x}_j^- - \overline{x}_j^- \geq 0$ for $1 \leq j \leq n$.

Remark 2 *If $(B + C)$ and $(B - C)$ are both nonsingular and y is an arbitrary intuitionistic fuzzy number vector. Then the system has a unique intuitionistic fuzzy solution if $(B + C)^{-1}V^+ \geq 0$ et $(B + C)^{-1}V^- \geq 0$.*

4 Examples

Example 1 The authors in [10] considered the electrical circuit shown in Fig. 1, where \tilde{v}_1 and \tilde{v}_2 are the input voltages, and \tilde{v}_3 and \tilde{v}_4 are the output voltages. The circuit is a kind of summing amplifier with two inputs and two outputs. The relationship between input and output voltages is as follows:

$$\begin{bmatrix} 3 & 0.5 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} \tilde{v}_1 \\ \tilde{v}_2 \end{bmatrix} = \begin{bmatrix} \tilde{v}_3 \\ \tilde{v}_4 \end{bmatrix} \tag{6}$$

They considered the output voltages as type-2 fuzzy numbers. In this paper we treat the same example but we consider the output voltages as intuitionistic fuzzy numbers : $\tilde{v}_3 = (14 + 2\alpha; 18 - 2\alpha; 16 - 3\alpha; 16 + 3\alpha)$ and $\tilde{v}_4 = (-18 + 2\alpha; -14 - 2\alpha; -16 - 3\alpha; -16 + 3\alpha)$.

$$\begin{cases} 3\tilde{v}_1 + 0.5\tilde{v}_2 = (14 + 2\alpha; 18 - 2\alpha; 16 - 3\alpha; 16 + 3\alpha) \\ -2\tilde{v}_1 - 3\tilde{v}_2 = (-18 + 2\alpha; -14 - 2\alpha; -16 - 3\alpha; -16 + 3\alpha) \end{cases} \tag{7}$$

Hence

$$\begin{aligned} 3\underline{v}_1^+ + 0.5\underline{v}_2^+ &= 14 + 2\alpha \\ 3\underline{v}_1^+ + 0.5\underline{v}_2^+ &= 18 - 2\alpha \\ -2\underline{v}_1^+ - 3\underline{v}_2^+ &= -18 + 2\alpha \\ -2\underline{v}_1^+ - 3\underline{v}_2^+ &= -14 - 2\alpha \end{aligned}$$

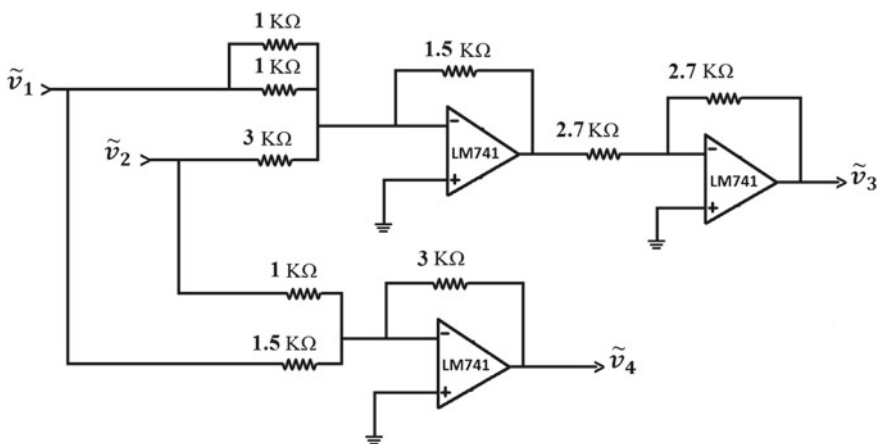


Fig. 1 Electrical circuit

and

$$\begin{aligned} 3\underline{v}_1^- + 0.5\underline{v}_2^- &= 16 - 3\alpha \\ 3\overline{v}_1^- + 0.5\overline{v}_2^- &= 16 + 3\alpha \\ -2\underline{v}_1^- - 3\underline{v}_2^- &= -16 - 3\alpha \\ -2\overline{v}_1^- - 3\overline{v}_2^- &= -16 + 3\alpha \end{aligned}$$

then we get 4 crisp systems

$$\begin{cases} 3z_1^+ + 0.5z_2^+ = 4 - 4\alpha \\ 2z_1^+ + 3z_2^+ = 4 - 4\alpha \end{cases}$$

where $z_1^+ = \overline{v}_1^+ - \underline{v}_1^+$, $z_2^+ = \overline{v}_2^+ - \underline{v}_2^+$.
and

$$\begin{cases} 3e_1^+ + 0.5e_2^+ = 32 \\ -2e_1^+ - 3e_2^+ = -32 \end{cases}$$

where $e_1^+ = \overline{v}_1^+ + \underline{v}_1^+$, $e_2^+ = \overline{v}_2^+ + \underline{v}_2^+$.

$$\begin{cases} 3z_1^- + 0.5z_2^- = 6\alpha \\ 2z_1^- + 3z_2^- = 6\alpha \end{cases}$$

where $z_1^- = \overline{v}_1^- - \underline{v}_1^-$, $z_2^- = \overline{v}_2^- - \underline{v}_2^-$.
and

$$\begin{cases} 3e_1^- + 0.5e_2^- = 32 \\ -2e_1^- - 3e_2^- = -32 \end{cases}$$

where $e_1^- = \overline{v}_1^- + \underline{v}_1^-$, $e_2^- = \overline{v}_2^- + \underline{v}_2^-$.

by solving systems, we have $z_1^+ = 1.27 - 1.27\alpha$, $z_2^+ = 0.5 - 0.5\alpha$, $e_1^+ = 10$ and $e_2^+ = 4$, and $z_1^- = 1.875\alpha$, $z_2^- = 1.875\alpha$, $e_1^- = 10$ and $e_2^- = 4$
hence

$$\begin{aligned} \underline{v}_1^+ &= 4.365 + 0.635\alpha \\ \overline{v}_1^+ &= 5.635 - 0.635\alpha \\ \underline{v}_1^- &= 5 - 0.9375\alpha \\ \overline{v}_1^- &= 5 + 0.9375\alpha \end{aligned}$$

and

$$\begin{aligned} \underline{v_2^+} &= 1.75 + 0.25\alpha \\ \overline{v_2^+} &= 2.25 - 0.25\alpha \\ \underline{v_2^-} &= 2 - 0.9375\alpha \\ \overline{v_2^-} &= 2 + 0.9375\alpha \end{aligned}$$

Since $\underline{v_1^+} \leq \overline{v_1^+}$, $\underline{v_1^-} \leq \overline{v_1^-}$, $\underline{v_2^+} \leq \overline{v_2^+}$, $\underline{v_2^-} \leq \overline{v_2^-}$ and $\underline{v_1^+}$, $\underline{v_2^+}$ and $\underline{v_1^-}$ and $\underline{v_2^-}$ are monotonic non-increasing and $\overline{v_1^+}$, $\overline{v_2^+}$, $\overline{v_1^-}$ and $\overline{v_2^-}$ are monotonic non-decreasing, then $\tilde{v}_1 = (\underline{v_1^+}, \overline{v_1^+}, \underline{v_1^-}, \overline{v_1^-})$ and $\tilde{v}_2 = (\underline{v_2^+}, \overline{v_2^+}, \underline{v_2^-}, \overline{v_2^-})$ is an intuitionistic fuzzy solution.

Example 2 We consider the following 2×2 intuitionistic fuzzy linear system:

$$\begin{cases} x_1 - x_2 = (\alpha; 2 - \alpha; 1 - 1.75\alpha; 1 + 1.75\alpha) \\ x_1 + 3x_2 = (4 + 2\alpha; 8 - 2\alpha; 6 - 3\alpha; 6 + 3\alpha) \end{cases} \tag{8}$$

Hence

$$\begin{aligned} \underline{x_1^+} - \overline{x_2^+} &= \alpha \\ \overline{x_1^+} - \underline{x_2^+} &= 2 - \alpha \\ \underline{x_1^+} + 3\underline{x_2^+} &= 4 + 2\alpha \\ \overline{x_1^+} + 3\overline{x_2^+} &= 8 - 2\alpha \end{aligned}$$

and

$$\begin{aligned} \underline{x_1^-} - \overline{x_2^-} &= 1 - 1.75\alpha \\ \overline{x_1^-} - \underline{x_2^-} &= 1 + 1.75\alpha \\ \underline{x_1^-} + 3\underline{x_2^-} &= 6 - 3\alpha \\ \overline{x_1^-} + 3\overline{x_2^-} &= 6 + 3\alpha \end{aligned}$$

then we get 4 crisp systems

$$\begin{cases} z_1^+ + z_2^+ = 2 - 2\alpha \\ z_1^+ + 3z_2^+ = 4 - 4\alpha \end{cases}$$

where $z_1^+ = \underline{x_1^+} - \overline{x_1^+}$, $z_2^+ = \underline{x_2^+} - \overline{x_2^+}$.

and

$$\begin{cases} e_1^+ - e_2^+ = 2 \\ e_1^+ + 3e_2^+ = 12 \end{cases}$$

where $e_1^+ = \overline{x_1^+} + \underline{x_1^+}$, $e_2^+ = \overline{x_2^+} + \underline{x_2^+}$.

$$\begin{cases} z_1^- + z_2^- = 3.5\alpha \\ z_1^- + 3z_2^- = 6\alpha \end{cases}$$

where $z_1^+ = \overline{x_1^+} - \underline{x_1^+}$, $z_2^+ = \overline{x_2^+} - \underline{x_2^+}$.

and

$$\begin{cases} e_1^- - e_2^- = 2 \\ e_1^- + 3e_2^- = 12 \end{cases}$$

where $e_1^- = \overline{x_1^-} + \underline{x_1^-}$, $e_2^- = \overline{x_2^-} + \underline{x_2^-}$.

by solving systems, we have $z_1^+ = 1 - \alpha$, $z_2^+ = 1 - \alpha$, $e_1^+ = 4.5$ and $e_2^+ = 2.5$, and $z_1^- = 2.25\alpha$, $z_2^- = 1.25\alpha$, $e_1^- = 4.5$ and $e_2^- = 2.5$ hence

$$\begin{aligned} \underline{x_1^+} &= 1.75 + 0.5\alpha \\ \overline{x_1^+} &= 2.75 - 0.5\alpha \\ \underline{x_1^-} &= 2.25 - 1.125\alpha \\ \overline{x_1^-} &= 2.25 + 1.125\alpha \end{aligned}$$

and

$$\begin{aligned} \underline{x_2^+} &= 0.75 + 0.5\alpha \\ \overline{x_2^+} &= 1.75 - 0.5\alpha \\ \underline{x_2^-} &= 1.25 - 0.625\alpha \\ \overline{x_2^-} &= 1.25 + 0.625\alpha \end{aligned}$$

Since $\underline{x_1^+} \leq \overline{x_1^+}$, $\underline{x_1^-} \leq \overline{x_1^-}$, $\underline{x_2^+} \leq \overline{x_2^+}$, $\underline{x_2^-} \leq \overline{x_2^-}$ and $\underline{x_1^+}$, $\underline{x_2^+}$ and $\underline{x_1^-}$ and $\underline{x_2^-}$ are monotonic non-increasing and $\overline{x_1^+}$, $\overline{x_2^+}$, $\overline{x_1^-}$ and $\overline{x_2^-}$ are monotonic non-decreasing, then $\tilde{x}_1 = (\underline{x_1^+}, \overline{x_1^+}, \underline{x_1^-}, \overline{x_1^-})$ and $\tilde{x}_2 = (\underline{x_2^+}, \overline{x_2^+}, \underline{x_2^-}, \overline{x_2^-})$ is an intuitionistic fuzzy solution.

5 Conclusion

In this paper we studied intuitionistic fuzzy linear system, we propose an efficient method for solving a system of n intuitionistic fuzzy linear equations with n variables. The original system is replaced by four $n \times n$ crisp systems. We have also considered a numerical examples and solved by using this approach.

References

1. Allahviranloo, T.: Numerical method for fuzzy system of linear equations. *Appl Math Comput.* **153**, 493–502 (2004)
2. Asady, B., Abbasbandy, S., Alavi, M.: Fuzzy general linear systems. *Appl. Math. Comput.* **169**(1), 34–40 (2005)
3. Abbasbandy, S., Alavi, M.: A method for solving fuzzy linear systems. *Iran. J. Fuzzy Syst.* **2**(2), 37–43 (2005)
4. Abbasbandy, S., Allahviranloo, T., Ezzati, R.: A method for solving fuzzy linear general systems. *J. Fuzzy Math.* **15**(4), 881–889 (2007)
5. Atanassov, K.T.: Intuitionistic fuzzy sets. VII ITKRs session, Sofia (deposited in Central Science and Technical Library of the Bulgarian Academy of Sciences (1697/84) (1983)
6. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **20**, 87–96 (1986)
7. Friedman, M., Min, Ma., Kandel, A.: Fuzzy linear systems, *Fuzzy Sets Syst.* **96**, 201–209 (1998)
8. Ezzati, R.: Solving linear systems. *Soft Comput.* **15**(1), 193–197 (2011)
9. Keyanpour1, M., Akbarian, T.: Solving intuitionistic fuzzy nonlinear equations. *J. Fuzzy Set Valued Anal.* 1–6 (2014)
10. Najariyan1, M., Mazandarani, M., John, R.: Type-2 fuzzy linear systems. *Granul. Comput.* **2**:175–186 (2017)
11. Zadeh, L.A.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965)

Nonlocal Intuitionistic Fuzzy Differential Equation



R. Ettoussi, Said Melliani and S. Chadli

Abstract In this paper, we shall prove the existence and uniqueness theorem of solution of intuitionistic fuzzy differential equation with nonlocal condition by using the contraction mapping principle and we establish some property of the solution like the continuous dependence.

Keywords Intuitionistic fuzzy differential equations · Intuitionistic fuzzy solution · Nonlocal initial condition

Mathematics Subject Classification (2010) 34A07 · 03E72

1 Introduction

In 1965, Zadeh [10] first introduced the fuzzy set theory. Later many researchers have applied this theory to the well known results in the classical set theory. A large class of physically important problems is described by fuzzy differential equations. Kaleva [5] discussed the properties of differentiable fuzzy set valued mappings and gave the existence and uniqueness theorem for a solution of the fuzzy differential equations $x'(t) = f(t, x(t))$, $x(t_0) = x_0$ when f satisfies the Lipschitz condition.

R. Ettoussi (✉) · S. Melliani · S. Chadli
Department of mathematics, Laboratory of Applied Mathematics and Scientific Computing,
Sultan Moulay Slimane University, PO Box 523, 23000 Beni Mellal, Morocco
e-mail: razika.imi@gmail.com

S. Melliani
e-mail: saidmelliani@gmail.com

S. Chadli
e-mail: sa.chadli@yahoo.fr

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_13

The idea of intuitionistic fuzzy set was first published by Atanassov [1, 2] as a generalization of the notion of fuzzy set. The notions of differential and integral calculus for intuitionistic fuzzy-set-valued are given using Hukuhara difference in intuitionistic Fuzzy theory [4]. The authors of papers [7, 9] are discussed differential and partial differential equations under intuitionistic fuzzy environment respectively. The existence and uniqueness of the solution of intuitionistic fuzzy differential equations by using successive approximations method have been discussed in [4], while in [3] the theorem of the existence and uniqueness of the solution for differential equations with intuitionistic fuzzy data are proved by using the theorem of fixed point in the complete metric space, also the explicit formula of the solution are given by using the α -cuts method.

This paper is organized as follows: in Sect. 2 we give preliminaries which we will use throughout this work. In Sect. 3, we prove the existence and uniqueness of solutions for Intuitionistic Fuzzy differential equations with nonlocal conditions of the following form:

$$\begin{cases} x'(t) = f(t, x(t)) & , t \in I = [0, a] \\ x(0) \oplus g(x) & = x_0 \end{cases} \quad (1.1)$$

where $x_0 \in \text{IF}_1$ f and g are given functions to be specified.

In the last section, we study the continuous dependence of mild solution of intuitionistic fuzzy differential equations.

2 Preliminaries

In this section, we introduce notations, definitions, and preliminary facts which are used throughout this paper.

Let us denote by $P_k(\mathbb{R})$ the set of all nonempty compact convex subsets of \mathbb{R} .

We denote by

$$\text{IF}_1 = \text{IF}(\mathbb{R}) = \{ \langle u, v \rangle : \mathbb{R} \rightarrow [0, 1]^2 / \forall x \in \mathbb{R}, 0 \leq u(x) + v(x) \leq 1 \}$$

An element $\langle u, v \rangle$ of IF_1 is said an intuitionistic fuzzy number if it satisfies the following conditions

- (i) $\langle u, v \rangle$ is normal i.e there exists $x_0, x_1 \in \mathbb{R}$ such that $u(x_0) = 1$ and $v(x_1) = 1$.
- (ii) u is fuzzy convex and v is fuzzy concave.
- (iii) u is upper semi-continuous and v is lower semi-continuous.
- (iv) $\text{supp } \langle u, v \rangle = \text{cl}\{x \in \mathbb{R} : |v(x) < 1\}$ is bounded.

So we denote the collection of all intuitionistic fuzzy number by IF_1

For $\alpha \in [0, 1]$ and $\langle u, v \rangle \in IF_1$, the upper and lower α -cuts of $\langle u, v \rangle$ are defined by

$$[\langle u, v \rangle]^\alpha = \{x \in \mathbb{R} : v(x) \leq 1 - \alpha\}$$

and

$$[\langle u, v \rangle]_\alpha = \{x \in \mathbb{R} : u(x) \geq \alpha\}$$

Remark 2.1 If $\langle u, v \rangle \in IF_1$, so we can see $[\langle u, v \rangle]_\alpha$ as $[u]^\alpha$ and $[\langle u, v \rangle]^\alpha$ as $[1 - v]^\alpha$ in the fuzzy case.

We define $0_{(1,0)} \in IF_1$ as

$$0_{(1,0)}(t) = \begin{cases} \langle 1, 0 \rangle & t = 0 \\ \langle 0, 1 \rangle & t \neq 0 \end{cases}$$

Let $\langle u, v \rangle, \langle u', v' \rangle \in IF_1$ and $\lambda \in \mathbb{R}$, we define the following operations by :

$$\left(\langle u, v \rangle \oplus \langle u', v' \rangle \right)(z) = \left(\sup_{z=x+y} \min(u(x), u'(y)), \inf_{z=x+y} \max(v(x), v'(y)) \right)$$

$$\lambda \langle u, v \rangle = \begin{cases} \langle \lambda u, \lambda v \rangle & \text{if } \lambda \neq 0 \\ 0_{(1,0)} & \text{if } \lambda = 0 \end{cases}$$

For $\langle u, v \rangle, \langle z, w \rangle \in IF_1$ and $\lambda \in \mathbb{R}$, the addition and scaler-multiplication are defined as follows

$$\begin{aligned} [\langle u, v \rangle \oplus \langle z, w \rangle]^\alpha &= [\langle u, v \rangle]^\alpha + [\langle z, w \rangle]^\alpha, \\ [\lambda \langle z, w \rangle]^\alpha &= \lambda [\langle z, w \rangle]^\alpha \end{aligned} \tag{2.1}$$

$$\begin{aligned} [\langle u, v \rangle \oplus \langle z, w \rangle]_\alpha &= [\langle u, v \rangle]_\alpha + [\langle z, w \rangle]_\alpha, \\ [\lambda \langle z, w \rangle]_\alpha &= \lambda [\langle z, w \rangle]_\alpha \end{aligned} \tag{2.2}$$

Definition 2.1 Let $\langle u, v \rangle$ an element of IF_1 and $\alpha \in [0, 1]$, we define the following sets:

$$\begin{aligned} \left[\langle u, v \rangle \right]_l^+(\alpha) &= \inf\{x \in \mathbb{R} \mid u(x) \geq \alpha\}, & \left[\langle u, v \rangle \right]_r^+(\alpha) &= \sup\{x \in \mathbb{R} \mid u(x) \geq \alpha\} \\ \left[\langle u, v \rangle \right]_l^-(\alpha) &= \inf\{x \in \mathbb{R} \mid v(x) \leq 1 - \alpha\}, & \left[\langle u, v \rangle \right]_r^-(\alpha) &= \sup\{x \in \mathbb{R} \mid v(x) \leq 1 - \alpha\} \end{aligned}$$

Remark 2.2

$$\left[\langle u, v \rangle \right]_\alpha = \left[\left[\langle u, v \rangle \right]_l^+(\alpha), \left[\langle u, v \rangle \right]_r^+(\alpha) \right], \quad \left[\langle u, v \rangle \right]^\alpha = \left[\left[\langle u, v \rangle \right]_l^-(\alpha), \left[\langle u, v \rangle \right]_r^-(\alpha) \right]$$

Theorem 2.1 [8] (IF_1, d_∞) is a complete metric space.

Definition 2.2 [3] Let $F : [a, b] \rightarrow IF_1$ be an intuitionistic fuzzy valued mapping and $t_0 \in [a, b]$. Then F is called intuitionistic fuzzy continuous in t_0 iff:

$$(\forall \varepsilon > 0)(\exists \delta > 0) \left(\forall t \in [a, b] \text{ such as } |t - t_0| < \delta \right) \Rightarrow d_\infty(F(t), F(t_0)) < \varepsilon$$

Definition 2.3 [6]

Let $\langle u, v \rangle, \langle u', v' \rangle \in IF_1$ the H -difference is the IFN $\langle z, w \rangle \in IF_1$, if it exists, such that

$$\langle u, v \rangle \ominus \langle u', v' \rangle = \langle z, w \rangle \iff \langle u, v \rangle = \langle u', v' \rangle \oplus \langle z, w \rangle$$

Definition 2.4 [3] A mapping $F : [a, b] \rightarrow IF_1$ is said to be differentiable at $t_0 \in (a, b)$ if there exist $F'(t_0) \in IF_1$ such that both limits:

$$\lim_{\Delta t \rightarrow 0^+} \frac{F(t_0 + \Delta t) \ominus F(t_0)}{\Delta t} \quad \text{and} \quad \lim_{\Delta t \rightarrow 0^+} \frac{F(t_0) \ominus F(t_0 - \Delta t)}{\Delta t} \quad (2.3)$$

exist and they are equal to $F'(t_0)$, which is called intuitionistic fuzzy derivative of F at t_0 . Here the limit is taken in the metric space (IF_1, d_∞) . At the end points of $[a, b]$ we consider only the one-sided derivatives.

Definition 2.5 [3] $F : [a, b] \rightarrow IF_1$ is called integrably bounded if there exists an integrable function h such that $|y| \leq h(t)$ holds for any $y \in \text{supp}(F(t)), t \in [a, b]$.

Definition 2.6 [3] we say that a mapping $F : [a, b] \rightarrow IF_1$ is strongly measurable if for all $\alpha \in [0, 1]$ the set-valued mapping $F_\alpha : [a, b] \rightarrow P_k(\mathbb{R})$ defined by $F_\alpha(t) = [F(t)]_\alpha$ and $F^\alpha : [a, b] \rightarrow P_k(\mathbb{R})$ defined by $F^\alpha(t) = [F(t)]^\alpha$ are (Lebesgue) measurable, when $P_k(\mathbb{R})$ is endowed with the topology generated by the Hausdorff metric d_H .

Definition 2.7 Suppose $A = [a, b], F : A \rightarrow IF_1$ is integrably bounded and strongly measurable for each $\alpha \in (0, 1]$ write

$$\left[\int_A F(t)dt \right]_\alpha = \int_A [F(t)]_\alpha dt = \left\{ \int_A f dt \mid f : A \rightarrow \mathbb{R} \text{ is a measurable selection for } F_\alpha \right\}.$$

$$\left[\int_A F(t)dt \right]^\alpha = \int_A [F(t)]^\alpha dt = \left\{ \int_A f dt \mid f : A \rightarrow \mathbb{R} \text{ is a measurable selection for } F^\alpha \right\}.$$

if there exists $\langle u, v \rangle \in IF_1$ such that $\left[\langle u, v \rangle \right]^\alpha = \left[\int_A F(t)dt \right]^\alpha$ and $\left[\langle u, v \rangle \right]_\alpha = \left[\int_A F(t)dt \right]_\alpha \forall \alpha \in (0, 1]$. Then F is called integrable on A , write $\langle u, v \rangle = \int_A F(t)dt$.

3 Nonlocal Intuitionistic Fuzzy Differential Equation

Throughout this work, we suppose that

- (H₁) $f : I \times IF_1 \rightarrow IF_1$ is continuous intuitionistic fuzzy mapping with respect to t which satisfies a Lipschitz condition i.e there exists constant L such that $d_\infty(f(t, x), f(t, y)) \leq Ld_\infty(x, y)$.
- (H₂) $g : I \times IF_1 \rightarrow IF_1$ satisfies a Lipschitz condition i.e there exists constant K such that $d_\infty(g(x), g(y)) \leq Kd_\infty(x, y) \forall t \in I$ and $x, y \in IF_1$.

We consider the nonlocal intuitionistic fuzzy differential equation:

$$\begin{cases} x'(t) = f(t, x(t)), & t \in I = [0, a] \\ x(0) \oplus g(x) = x_0 \end{cases} \tag{3.1}$$

Definition 3.1 A function $x : I \rightarrow IF_1$ is said to be a mild solution of Eq. (3.1), if

$$x(t) = x_0 \ominus g(x) \oplus \int_0^t f(s, x(s))ds$$

for $0 \leq t \leq a$.

Theorem 3.1 Assume that assumptions (H₁) – (H₂) hold. Then Eq. (3.1) has a unique mild solution on the interval $[0, \xi]$ where

$$\xi = \min \left\{ a, \frac{b - N}{M}, \frac{1 - K}{L} \right\}$$

and

$$d_\infty(f(t, x), 0_{(1,0)}) \leq M, \quad d_\infty(g(x), 0_{(1,0)}) \leq N$$

Proof Let $B = \{x \in \mathbb{IF}_1 | H(x, x_0) \leq b\}$ be the space of continuous intuitionistic fuzzy mappings with

$$H(x, y) = \sup_{0 \leq t \leq \xi} d_\infty(x(t), y(t))$$

and b a positive number.

Define a mapping $P : B \rightarrow B$ by

$$Px(t) = x_0 \ominus g(x) \oplus \int_0^t f(s, x(s)) ds$$

First of all, we show that P is continuous and $H(Px, x_0) \leq b$.

Since f is continuous, we have

$$\begin{aligned} d_\infty(Px(t+h), Px(t)) &= d_\infty\left(x_0 \ominus g(x) \oplus \int_0^{t+h} f(s, x(s)) ds, \right. \\ &\quad \left. x_0 \ominus g(x) \oplus \int_0^t f(s, x(s)) ds\right) \\ &\leq d_\infty\left(\int_0^{t+h} f(s, x(s)) ds, \int_0^t f(s, x(s)) ds\right) \\ &\leq \int_t^{t+h} d_\infty\left(f(s, x(s)), 0_{(1,0)}\right) ds \\ &= hM \rightarrow 0 \quad (h \rightarrow 0) \end{aligned}$$

That is, the map P is continuous on I .

Furthermore,

$$\begin{aligned} d_\infty(Px(t), x_0) &= d_\infty\left(x_0 \ominus g(x) \oplus \int_0^t f(s, x(s)) ds, x_0\right) \\ &\leq d_\infty(g(x), 0_{(1,0)}) + d_\infty\left(\int_0^t f(s, x(s)) ds, 0_{(1,0)}\right) \\ &\leq d_\infty(g(x), 0_{(1,0)}) + \int_0^t d_\infty\left(f(s, x(s)), 0_{(1,0)}\right) ds \\ &\leq N + Mt \end{aligned}$$

and so

$$\begin{aligned} H(Px, x_0) &= \sup_{0 \leq t \leq \xi} \varphi(Px(t), x_0) \\ &\leq N + M\xi \\ &\leq b. \end{aligned}$$

Since (IF_1, d_∞) see [8] is a complete metric space, a standard proof applies to show that

$$C([0, \xi], IF_1) = \{x : [0, \xi] \rightarrow IF_1 \mid x(t) \text{ is continuous} \}$$

is complete.

Now we show that B is a closed subset of $C([0, \xi], IF_1)$.

Let $\{x_n\}$ be a sequence in B such that $x_n \rightarrow x \in C([0, \xi], IF_1)$ as $n \rightarrow \infty$.

Then,

$$\begin{aligned} d_\infty(x(t), x_0) &\leq d_\infty(x(t), x_n(t)) + d_\infty(x_n(t), x_0) \\ \text{Thus, } H(x, x_0) &= \sup_{0 \leq t \leq \xi} d_\infty(x(t), x_0) \\ &\leq H(x, x_n) + H(x_n, x_0) \\ &\leq \varepsilon + b \end{aligned}$$

for sufficiently large n and arbitrary $\varepsilon > 0$. So $x \in B$. This implies that B is a closed subset of $C([0, \xi], IF_1)$.

Therefore B is a complete metric space.

Next, we will show that P is a contraction mapping.

For $x, y \in B$

$$\begin{aligned} d_\infty(Px(t), Py(t)) &\leq d_\infty\left(x_0 \ominus g(x) \oplus \int_0^t f(s, x(s))ds, x_0 \ominus g(y) \oplus \int_0^t f(s, y(s))ds\right) \\ &\leq d_\infty(g(x), g(y)) + \int_0^t d_\infty\left(f(s, x(s)), f(s, y(s))\right)ds \\ &\leq Kd_\infty(x(t), y(t)) + L \int_0^t d_\infty\left(x(s), y(s)\right)ds \end{aligned}$$

Thus, we obtain

$$\begin{aligned} H(Px, Py) &\leq \sup_{0 \leq t \leq \xi} \left\{ Kd_\infty(x(t), y(t)) + L \int_0^t d_\infty\left(x(s), y(s)\right)ds \right\} \\ &\leq (K + \xi L)H(x, y) \end{aligned}$$

Since $K + \xi L < 1$, P is a contraction map.

Therefore P has a unique fixed point $Px = x \in C([0, \xi], IF_1)$, that is

$$x(t) = x_0 \ominus g(x) \oplus \int_0^t f(s, x(s))ds.$$

Next theorem shows the continuity and the boundedness of the Solution.

4 Continuous Dependence of Mild Solution of Intuitionistic Fuzzy Differential Equation

Theorem 4.1 Suppose that f, g are the same as in Theorem 3.1. Let $x(t, x_0), y(t, y_0)$ be solutions of Eq. (3.1) to x_0, y_0 , respectively. Then there exist constant c_1 such that

$$H(x(\cdot, x_0), y(\cdot, y_0)) \leq c_1 H(x_0, y_0)$$

for any $x_0, y_0 \in IF_1$.

Where

$$H(x, y) = \sup_{0 \leq t \leq \xi} d_\infty(x(t), y(t))$$

for all $x, y \in C([0, \xi], IF_1)$

Proof For any $t \in [0, \xi]$ we have

$$\begin{aligned} d_\infty(x(t, x_0), y(t, y_0)) &\leq d_\infty(x_0, y_0) + d_\infty\left(g(x(\cdot, x_0)), g(y(\cdot, y_0))\right) \\ &\quad + d_\infty\left(\int_0^t f(s, x(s, x_0))ds, \int_0^t f(s, y(s, y_0))ds\right) \\ &\leq d_\infty(x_0, y_0) + d_\infty\left(g(x(\cdot, x_0)), g(y(\cdot, y_0))\right) \\ &\quad + \int_0^t d_\infty\left(f(s, x(s, x_0)), f(s, y(s, y_0))\right)ds \\ &\leq d_\infty(x_0, y_0) + Kd_\infty(x(\cdot, x_0), y(\cdot, y_0)) + L \int_0^t d_\infty\left(x(s, x_0), y(s, y_0)\right)ds \end{aligned}$$

Thus we have

$$H(x(\cdot, x_0), y(\cdot, y_0)) \leq H(x_0, y_0) + [K + L\xi]H(x(\cdot, x_0), y(\cdot, y_0))$$

i.e.,

$$\left(1 - K - L\xi\right)H(x(\cdot, x_0), y(\cdot, y_0)) \leq H(x_0, y_0)$$

Consequently, we obtain

$$H(x(\cdot, x_0), y(\cdot, y_0)) \leq \frac{1}{1 - K - L\xi} H(x_0, y_0)$$

Taking $c_1 = \frac{1}{1-K-L\xi}$, we obtain

$$H(x(\cdot, x_0), y(\cdot, y_0)) \leq c_1 H(x_0, y_0).$$

References

1. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets and Systems* **20**, 87–96 (1986)
2. Atanassov, K.: *Intuitionistic Fuzzy Sets: Theory and Applications*. (1999). Physica-Verlag
3. Ettoussi, R., Melliani, S., Chadli, L.S.: Differential equation with intuitionistic fuzzy parameters. *Notes on Intuitionistic Fuzzy Sets* **23**(4), 46–61 (2017)
4. Ettoussi, R., Melliani, S., Elomari, M., Chadli, L.S.: Solution of intuitionistic fuzzy differential equations by successive approximations method. *Notes on Intuitionistic Fuzzy Sets* **21**(2), 51–62 (2015)
5. Kaleva, O.: Fuzzy differential equations. *Fuzzy Sets and Systems* **24**, 301–317 (1987)
6. Melliani, S., Elomari, M., Chadli, L.S., Ettoussi, R.: Extension of hukuhara difference in intuitionistic fuzzy theory. *Notes on Intuitionistic Fuzzy Sets* **21**(4), 34–47 (2015)
7. Melliani, S., Chadli, L.S.: Introduction to intuitionistic fuzzy partial differential Equations. *Notes on intuitionistic Fuzzy sets* **7**(3), 39–42 (2001)
8. Melliani, S., Elomari, M., Chadli, L.S., Ettoussi, R.: Intuitionistic Fuzzy metric spaces. *Notes on intuitionistic Fuzzy sets* **21**(1), 43–53 (2015)
9. Melliani, S., Chadli, L.S.: Introduction to intuitionistic fuzzy differential equations. *Notes on intuitionistic Fuzzy sets* **6**(2), 31–41 (2000)
10. Zadeh, L.A.: Fuzzy sets. *Information and Control* **8**, 338–353 (1965)

Metaheuristics: Theory and Applications

Harmony Search with Dynamic Adaptation of Parameters for the Optimization of a Benchmark Controller



Cinthia Peraza, Fevrier Valdez and Oscar Castillo

Abstract A fuzzy harmony search algorithm (FHS) is presented in this paper. This method uses a fuzzy system for dynamic adaptation of the harmony memory accepting (HMR) and pitch adjustment (PARate) parameters along the iterations, and in this way achieving control of the intensification and diversification of the search space. This method was previously applied to various benchmark controller cases however in this case we decided to apply the proposed FHS to benchmark controller problem with different types of noise: band-limited white noise, pulse noise, and uniform random number noise to check the efficiency for the proposed method. A comparison is presented to verify the results obtained with the original harmony search algorithm and fuzzy harmony search algorithm.

Keywords Harmony search · Fuzzy logic · Dynamic parameter adaptation · Fuzzy controller · Benchmark problem

1 Introduction

This article focuses on optimizing a benchmark control problem in conjunction with three types of external perturbations: band-limited white noise, pulse noise, and uniform random number noise. The proposed method is a modification of the original harmony search algorithm (HS), which uses the improvisation process of jazz musicians [1]. This method has been used to solve optimization problems, there are also hybrid methods such as in [2]. The main difference between the variants and the existing methods in the literature is that this method is based on the original harmony search algorithm and uses fuzzy logic to dynamically adjust the *HMR* and *PARate*

C. Peraza · F. Valdez (✉) · O. Castillo
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: fevrier@tectijuana.mx

C. Peraza
e-mail: cinthia.peraza18@tectijuana.edu.mx

O. Castillo
e-mail: ocastillo@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_14

parameters, to achieve a control of the algorithm in the search space applied to benchmark control problem. Fuzzy logic is used in order to control the diversification and intensification, processes in the search space and enable finding the global optimum, avoiding stagnation and premature convergence. Fuzzy logic has also recently been used in existing metaheuristic methods because it uses linguistic variables and rules with which fuzzy models make decisions [3–5] and have been used in areas of evolutionary algorithms with fuzzy logic [6–10] engineering problems [11–14] among others.

This article describes the FHS method that can be applied for solving optimization fuzzy logic controller problem. The proposed method uses fuzzy logic to achieve an efficient adjustment of parameters and this methodology is applied to the optimization of a control problem with different types of disturbances.

This article is organized as follows. The original harmony search algorithm is shown in Sect. 2. The proposed method with fuzzy logic is shown in Sect. 3. Section 4 shows the methodology for parameter adaptation. The experimental results are shown in Sect. 5. Conclusions are shown in Sect. 6.

2 Harmony Search Algorithm

This algorithm is inspired by music and was created by ZW Geem in 2001 [1].

This algorithm uses three operators which are: harmony memory accepting (HMR), pitch adjustment (PARate) and randomization, and the range of these parameters is from 0 to 1.

HMR parameter represents the exploitation in a search space and its objective is to choose the best harmonies.

$$HMR \in [0, 1] \quad (1)$$

The pitch adjustment (PARate) is the second component, this parameter is given by Eq. (2).

$$x_{new} = x_{old} + b_p(2rand - 1) \quad (2)$$

Randomization is the third component, this parameter is given by Eq. (3).

$$P_a = P_{lower\ limit} + P_{range} * rand \quad (3)$$

where “rand” is a generator of random numbers in the range of 0 and 1. (Search space).

2.1 Pseudocode for Harmony Search

The pseudocode of this algorithm is explained in detail in Fig. 1.

3 Proposed Method

In previous works the harmony search algorithm has been studied and applied to problems of optimization, benchmark mathematical functions [15–17], benchmark control problems [18–20] among others.

The main difference with the previous works is that are have added a second input to the fuzzy system, which is called diversity and this is calculated with the Euclidean distance.

The HMR and PArate parameters are adjusted dynamically by means of a fuzzy system to control the exploration and exploitation of the search space by fuzzy rules.

The main difference between the original and the fuzzy method is that the original leaves the parameters fixed throughout the iterations and the fuzzy will move those parameters in each iteration.

The FHS uses a fuzzy system to be responsible for dynamically changing the PArate and HMR parameters in the range from 0 to 1 in each iteration number as shown in Fig. 2 and expressed as follows:

```

Objective function  $f(x)$ ,  $x = (x_1, \dots, x_n)^T$ 
Initial generate harmonics (matrices of real numbers)
Define pitch adjustment rate (PArate) and limits of tone
Define acceptance rate of the harmony memory (HMR)
while (t < Maximum number of iterations)
    Generate a new harmony and accept the best harmonies
    Setting the tone for new harmonies (solutions)
    if (rand > HMR)
        Choose an existing harmony randomly
    else if (rand > PArate)
        Setting the tone at random within a bandwidth (2)
    else
        Generate a new harmony through a randomization (3)
    End if
    Accepting new harmonies (solutions) best
End while
To find the best solutions.
    
```

Fig. 1 Pseudocode for harmony search

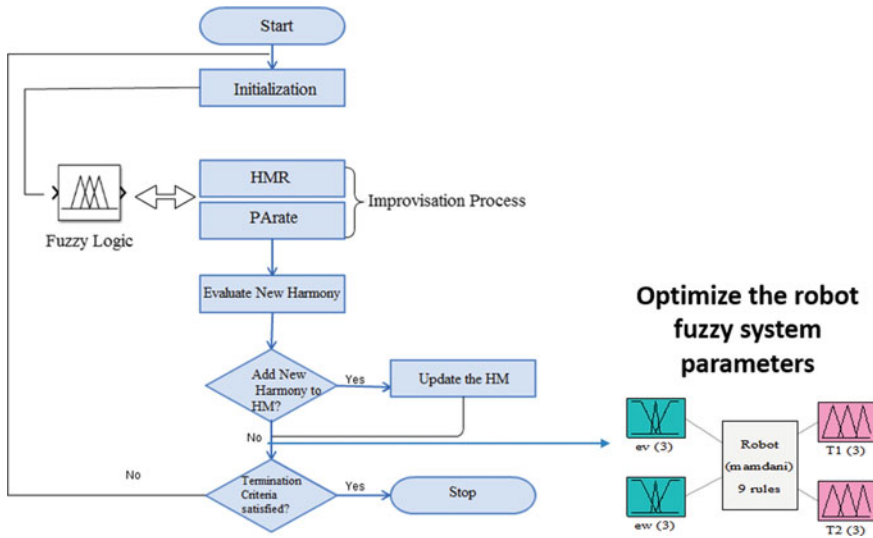


Fig. 2 Proposed fuzzy system

4 Methodology for Parameter Adaptation

Based on the previous study of the HS algorithm, it was decided to adjust the HMR parameter in order to control the exploitation and the PARate parameter to achieve the control of the exploration. These parameters will be representing the outputs of the fuzzy system in a range of 0 to 1.

The inputs of the fuzzy system will be the iterations that are represented by Eq. (4) and a second input is diversity, that is defined by Eq. (5). The design of the proposed method can be find in Fig. 3.

$$\text{Iteration} = \frac{\text{Current Iteration}}{\text{Maximum of Iterations}} \tag{4}$$

$$\text{Diversity}(S(t)) = \frac{1}{n_S} \sum_{i=1}^{n_S} \sqrt{\sum_{j=1}^{n_x} (x_{ij}(t) - \bar{x}_j(t))^2} \tag{5}$$

The design of the input and output variables can be appreciated in Fig. 3; the inputs these are granulated into three triangular membership functions. The linguistic values are Low, Medium and High. The outputs inputs these are granulated into five triangular membership functions. The linguistic values are Low, Medium Low, Medium, Medium High and High.

The rules were created based on the function of each parameter and are defined in Fig. 4.

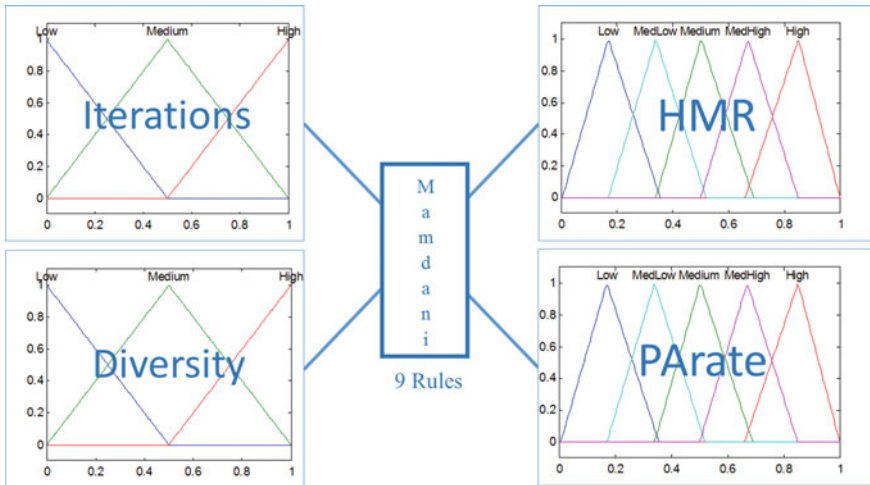


Fig. 3 Fuzzy system for dynamic parameter adaptation

No.	Inputs		Outputs	
	Generations	Diversity	HMR	PArate
1	Low	Low	High	Low
2	Low	Medium	Medium Low	Medium High
3	Low	High	Medium	Medium
4	Medium	Low	Medium	Medium Low
5	Medium	Medium	Medium	Medium
6	Medium	High	Medium	High
7	High	Low	Medium	High
8	High	Medium	Medium High	Medium Low
9	High	High	High	High

Fig. 4 Rules for the fuzzy system FHS

5 Simulation Results and Study Case

To test noise resilience of Fuzzy Controllers, a Fuzzy Controller of a mobile robot is presented in this paper, in conjunction with three types of external perturbations: band-limited white noise, pulse noise, and uniform random number noise. Noise resilience is measured to the root mean square error (RMSE). In other works, these types of noise have been used as shown in [21].

5.1 Study Case

The model which is used in the experiments in this paper is of a unicycle mobile robot, consisting of two driving wheels located on the same axis and a front free wheel. Figure 5 shows a graphical description of the robot model.

The main goal of this controller is to follow a reference trajectory, based on the model of a unicycle mobile robot which is composed of two drive wheels located on the same axis and a front free wheel that is used only for stability.

The operation of the robot model is determined by Eqs. (6) and (7).

$$M(q)\dot{v} + C(q, \dot{q})v + Dv = \tau + P(t) \tag{6}$$

where,

$q = (x, y, \theta)^T$ is the vector of the configuration coordinates.

$v = (v, w)^T$ is the vector of velocities.

$\tau = (\tau_1, \tau_2)$ is the vector of torques applied to the wheels of the robot where τ_1 and τ_2 denote the torques of the right and left wheel, respectively.

$P \in R^2$ is the uniformly bounded disturbance vector.

$M(q) \in R^{2 \times 2}$ is the positive-definite inertia matrix.

$C(q, \dot{q})\vartheta$ is the vector of centripetal and Coriolis forces.

$D \in R^{2 \times 2}$ is a diagonal positive-definite damping matrix.

The kinematic system is determined by Eq. (8):

$$\dot{q} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} \tag{7}$$

where, (x, y) is the position in the $X - Y$ (world) reference frame θ is the angle between the heading direction and the x -axis v and w are the linear and angular

Fig. 5 Mobile robot model

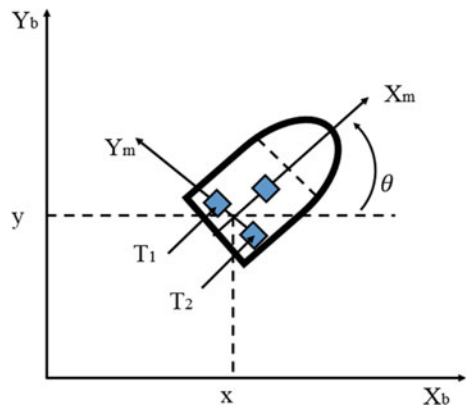


Table 1 Fuzzy rules for the robot mobile controller

Rule number	Inputs			Output	
	Ev	Operator	Ew	T1	T2
1	N	And	N	N	N
2	N	And	Z	N	Z
3	N	And	P	N	P
4	Z	And	N	Z	N
5	Z	And	Z	Z	Z
6	Z	And	P	Z	P
7	P	And	N	P	N
8	P	And	Z	P	Z
9	P	And	P	P	P

velocities, respectively Eq. (8) represents the non-holonomic constraint, which this system has, which corresponds to a non-slip wheel condition preventing the robot from moving sideways.

$$\dot{y} \cos \theta - \dot{x} \sin \theta = 0 \tag{8}$$

This controller is composed of two inputs and two outputs, the first input is the **ev** (error in the linear velocity) which has the two trapezoidal and one triangular membership functions with linguistic variables called N, Z and P. The second input is the **ew** (error in the angular velocity) which has the two trapezoidal and one triangular membership functions with linguistic variables of N, Z and P. The first output is the **T1** (torque 1) which has the three triangular membership functions with linguistic variables of N, Z and P. The second output is the **T2** (torque 2) which has the three triangular membership functions with linguistic variables of N, Z and P. This controller it is of Mamdani type and uses 9 fuzzy rules, which are presented in detailed in Table 1.

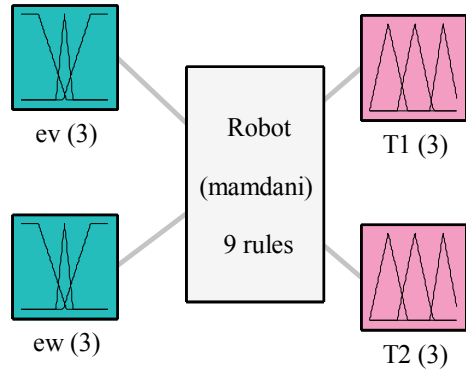
Table 1 presents in detail each of the rules that compose the robot mobile controller to achieve the main objective of this controller. Figure 6 shows the structure of the fuzzy system for this controller.

5.2 Simulations Results

Experimentation was done with various external perturbation scenarios; three specific noise generators were used:

- **Band-limited white noise** (Where the height of the Power Spectral Density of the band-limited white noise was set to 0.1).

Fig. 6 Structure of the robot mobile controller



- **Pulse generated** (the amplitude, period, pulse width (%) and phase delay were set to 1, 1, 5, and 0, respectively).
- **Uniform random number** (The amplitude of the uniform random white noise was set to $[-1, 1]$).

30 experiments were performed using each type of noise, where the objective function is to minimize the root mean square error that is described in Eq. (9) for the robot mobile controller.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (x_t - \hat{x}_t)^2} \tag{9}$$

The parameters used for the experiments are shown in Table 2 and the representation of the harmony for the optimization of the membership functions of the controller is shown in Fig. 7.

The results obtained from the optimization of the robot controller are shown in Table 3, where it can be observed that when using pulse noise, the error is improved.

Table 3 shows the results obtained for the average of the 30 experiments for each type of noise applied to FLC.

Figures 8, 9 and 10 show the best simulation obtained when applying uniform noise, band-limit and pulse respectively.

Table 2 Parameters used for the experiments

Parameters	Fuzzy HS
Harmonies	35
Dimensions	40
HMR	Dynamic
Parate	0.75
Dimension	40
Runs	30

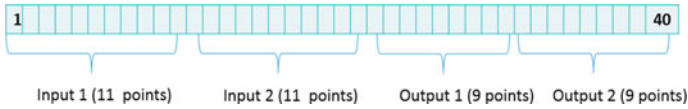


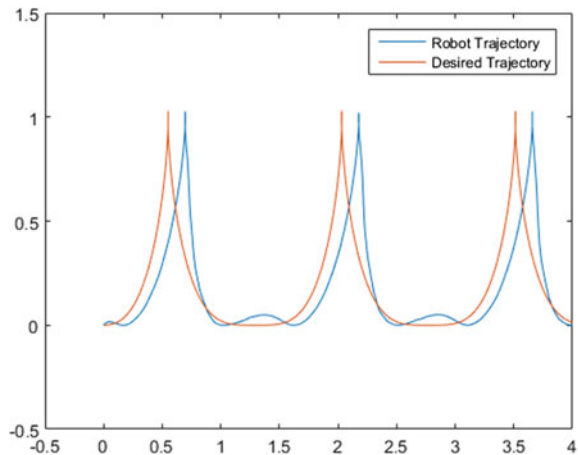
Fig. 7 Harmony for the optimization problem

Table 3 Results of the optimization for mobile robot controller

Type of noise	Uniform random number	Band-limit white noise	Pulse
Best	2.98E-03	7.40E-03	1.51E-03
Worst	1.08E+00	4.91E-01	5.17E-01
Average	2.15E-01	1.46E-01	1.38E-01
Standard deviation	2.70E-01	1.32E-01	1.32E-01

Bold indicates best results

Fig. 8 FLC with uniform random number



6 Conclusions

In this paper a complete fuzzy system based on the HS algorithm is presented, with which the diversity of the solutions is achieved. In this case we applied a fuzzy system to be responsible to dynamically adjust the HMR and PARate parameters and it is applied to the optimization for the robot mobile controller using different types of noise.

A comparison of three types of noise was performed using the fuzzy harmony search algorithm, 30 experiments were carry out. It can be observed that by using the FHS method it is possible to obtain better results by optimizing the membership

Fig. 9 FLC with band limit with noise

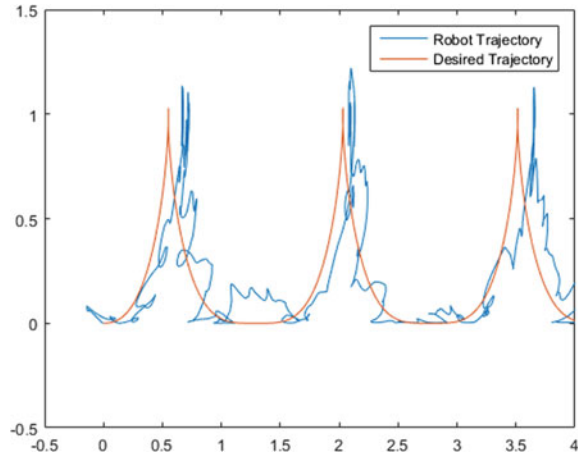
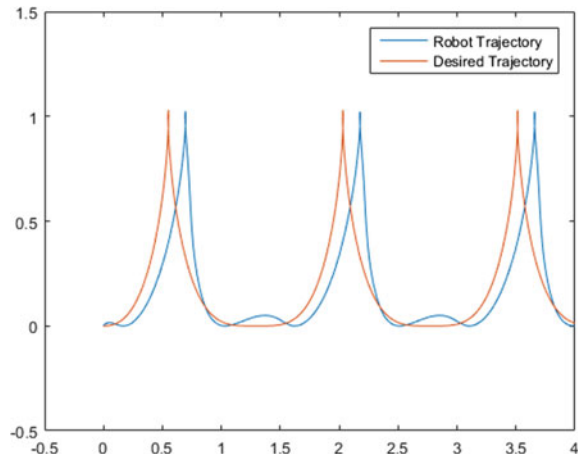


Fig. 10 FLC with pulse



functions of the robot controller using the noise pulse generator, unlike the other types of noise.

As future work we can use the proposed method in different applications, like in [22–27].

Acknowledgements We would like to express our thanks to CONACYT and Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Geem, Z.W., Kim, Z.H., Loganathan, G.V.: A new heuristic optimization algorithm: harmony search. *Simulation* **76**(2), 60–68 (2001)
2. Kar, P., Swain, S.C.: A Harmony Search-Firefly Algorithm Based Controller for Damping Power Oscillations, pp. 351–355 (2016)
3. Zadeh, L.A.: Fuzzy logic. *Computer* **21**(4), 83–93 (1988)
4. Zadeh, L.A.: The concept of a linguistic variable and its application to approximate reasoning—I. *Inf. Sci.* **8**(3), 199–249 (1975)
5. Zadeh, L.A.: Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets Syst.* **100**, 9–34 (1999)
6. Castillo, O., Valdez, F., Soria, J., Amador-Angulo, L., Ochoa, P., Peraza, C.: Comparative study in fuzzy controller optimization using bee colony, differential evolution, and harmony search algorithms. *Algorithms* **12**(1), 9 (2018)
7. Ochoa, P., Castillo, O., Soria, J.: A new approach for dynamic mutation parameter in the differential evolution algorithm using fuzzy logic. In: Melin, P., Castillo, O., Kacprzyk, J., Reformat, M., Melek, W. (eds.) *Fuzzy Logic in Intelligent System Design*, vol. 648, pp. 85–93. Springer, Cham (2018)
8. Bernal, E., Castillo, O., Soria, J., Valdez, F.: Optimization of fuzzy controller using galactic swarm optimization with type-2 fuzzy dynamic parameter adjustment. *Axioms* **8**(1), 26 (2019)
9. Barraza, J., Rodríguez, L., Castillo, O., Melin, P., Valdez, F.: A new hybridization approach between the fireworks algorithm and grey wolf optimizer algorithm. *J. Optim.* **2018**, 1–18 (2018)
10. Rodríguez, L., Castillo, O., García, M., Soria, J.: A comparative study of dynamic adaptation of parameters in the GWO algorithm using type-1 and interval type-2 fuzzy logic. In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, vol. 749, pp. 3–16. Springer, Cham (2018)
11. Al-Betar, M.A., Awadallah, M.A., Khader, A.T., Bolaji, A.L., Almomani, A.: Economic load dispatch problems with valve-point loading using natural updated harmony search. *Neural Comput. Appl.* **29**(10), 767–781 (2018)
12. Alia, O.M.: Dynamic relocation of mobile base station in wireless sensor networks using a cluster-based harmony search algorithm. *Inf. Sci.* **385–386**, 76–95 (2017)
13. Brinda, M.D., Suresh, A., Rashmi, M.R.: Optimal sizing and distribution system reconfiguration of hybrid FC/WT/PV system using cluster computing based on harmony search algorithm. *Clust. Comput* (2018)
14. Chao, F., Zhou, D., Lin, C.-M., Zhou, C., Shi, M., Lin, D.: Fuzzy cerebellar model articulation controller network optimization via self-adaptive global best harmony search algorithm. *Soft. Comput.* **22**(10), 3141–3153 (2018)
15. Peraza, C., Valdez, F., Castillo, O.: Interval type-2 fuzzy logic for dynamic parameter adaptation in the Harmony search algorithm, pp. 106–112 (2016)
16. Peraza, C., Valdez, F., Garcia, M., Melin, P., Castillo, O.: A new fuzzy harmony search algorithm using fuzzy logic for dynamic parameter adaptation. *Algorithms* **9**(4), 69 (2016)
17. Peraza, C., Valdez, F., Castillo, O.: An adaptive fuzzy control based on harmony search and its application to optimization. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) *Nature-Inspired Design of Hybrid Intelligent Systems*, vol. 667, pp. 269–283. Springer, Cham (2017)
18. Peraza, C., Valdez, F., Melin, P.: Optimization of intelligent controllers using a type-1 and interval type-2 fuzzy harmony search algorithm. *Algorithms* **10**(3), 82 (2017)
19. Peraza, C., Valdez, F., Castro, J.R., Castillo, O.: Fuzzy dynamic parameter adaptation in the harmony search algorithm for the optimization of the ball and beam controller. *Adv. Oper. Res.* **2018**, 1–16 (2018)
20. Castillo, O., et al.: Shadowed type-2 fuzzy systems for dynamic parameter adaptation in harmony search and differential evolution algorithms. *Algorithms* **12**(1), 17 (2019)

21. Sanchez, M.A., Castillo, O., Castro, J.R.: Generalized type-2 fuzzy systems for controlling a mobile robot and a performance comparison with interval type-2 and type-1 fuzzy systems. *Expert Syst. Appl.* **42**(14), 5904–5914 (2015)
22. Melin, P., Castillo, O.: Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach. *IEEE Trans. Industr. Electron.* **48**(5), 951–955 (2001)
23. Gonzalez, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
24. Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Appl. Soft Comput.* **53**, 74–87 (2017)
25. Gaxiola, F., Melin, P., Valdez, F., Castro, J.R., Castillo, O.: Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO. *Appl. Soft Comput.* **38**, 860–871 (2016)
26. Castillo, O., Castro, J.R., Melin, P., Rodriguez-Diaz, A.: Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction. *Soft. Comput.* **18**(6), 1213–1224 (2014)
27. Castro, J.R., Castillo, O., Melin, P., Rodríguez Díaz, A.: Building fuzzy inference systems with a new interval type-2 fuzzy logic toolbox. *Trans. Comput. Sci.* **1**, 104–114 (2008)

Evaluation of Parallel Exploration and Exploitation Capabilities in Two PSO Variants with Intra Communication



Yunkio Kawano, Fevrier Valdez and Oscar Castillo

Abstract In this chapter, we propose two different PSO variants with parallel communication to balance the exploration and exploitation of the search space. The idea is that combining these two algorithms in parallel which share information between the two is intended to obtain better results than each of the algorithms separately. It should be said that no adaptation of parameters was applied so the results obtained are not the best compared to other works, but among the algorithms used there were significant differences.

Keywords PSO · Parallel · PSO-Explore · PSO-Exploit

1 Introduction

Nowadays there are different ways to execute an algorithm, the simplest way of programming and running an algorithm is sequentially, which although it is effective is also the slowest way there is. That is why there is another way to run the algorithms which is in parallel, where different processes of the algorithm are executed at the same time, but they have to be processes that are independent, since otherwise they would not receive the necessary information to function correctly within of the algorithm. Although this method is more complex to program, improvements in the execution times of the algorithm can be obtained.

In this case, we worked with two parallel particle swarm optimization algorithms, since one was focused on the exploration and the other on the exploitation of the search space.

In Table 1 the used hardware is described:

The experiments were made within the MATLAB [1] environment to simplify the use of arrays. We used three different PSO algorithms, a PSO algorithm that exploits and another that explores the search space and one more that includes the

Y. Kawano (✉) · F. Valdez · O. Castillo
Tijuana Institute of Tech, Tijuana, BC, Mexico
e-mail: monico.kawano@tectijuana.edu.mx

Table 1 Specs of the used hardware

Specifications	Description
Laptop model	DELL PRECISION M4800
CPU	Intel Core i7-4910MQ
Base frequently	2.90 GHz
# of cores	4/8 Threats
RAM	
Memory RAM	16.0 GB DDR3
Speed clock	1600 MHz
DISK	SSD Kingston A400
Storage capacity	480 GB
Speed read	500 Mb/s
Speed write	450 Mb/s
Software	MATLAB 2014a

Table 2 Set of benchmark mathematical functions name for the realized experiments

Number	Mathematical test function name
1	De Jong's function
1a	Axis parallel hyper-ellipsoid function
1b	Rotated hyper-ellipsoid function
2	Rosenbrock's function
6	Rastrigin's function
7	Schwefel's function
8	Griewangk's function
9	Sum of different power function
10	Ackley's path function

two previous algorithms, in this case making them work at the same time to have communication between them.

To test that the method used in parallel is good, a set of benchmark mathematical functions [2–4] were optimized are shown in Table 2.

The following section shows a brief description of the two used optimization algorithms as well as the modification made to improve the runtime.

2 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) [5–8] is a search and optimization technique in the field of machine learning. Developed by Eberhart and Kennedy in 1995 [9, 10], inspired by social behavior of bird flocking or fish schooling, in which a swarm of

individuals (called particles) moves within a delimited search space, collecting and sharing information about the best position found in the food (or better solution).

The following equation represents the update of the velocity vector of each particle i in each iteration k .

$$v_i^{k+1} = \omega \cdot v_i^k + \varphi_1 \cdot rand_1 \cdot (pBest_i - x_i^k) + \varphi_2 \cdot rand_2 \cdot (g_i - x_i^k)$$

A more detailed description of each factor is present a below:

- v_i^k Is the velocity of the particle i in the iteration k .
- ω Is the inertia factor.
- φ_1, φ_2 These are learning ratios (weights) that control the components cognitive and social.
- $rand_1, rand_2$ Are the random numbers between 0 and 1.
- x_i^k It's the current position of the particle i in the iteration k .
- $pBest_i$ It is the best position or solution found by the particle i so far
- g_i represents the position of the particle with the $pBest_fitness$ of the environment of p_i that is a local best.

The PSO algorithm initiates a swarm of random particles, where each of the contained particles could be a solution to the problem that is being considered. These possible solutions are evaluated in each of the iterations that we have.

The following Fig. 1 shows the pseudocode of the PSO algorithm to explore or Exploit, they basically change the parameters between one algorithm and the other:

In Fig. 1, the first cycle function called “FOR” that appears is to execute the algorithm 30 times and thus obtain the 30 results to be able to perform a statistical test of z-score. Inside the other cycle function called “WHILE” is where all the main processes of the PSO algorithm are made. After that, all that remains is to save the results in an array within the cycle function “FOR” and at the end of that

```

Begin Algorithm
  For i=1 to 30
    Initialize the positions and velocities of all particles
    While 1 to MaxIterations
      Compute Fitness value
      Update best local position
      Update best global position
      Update Velocity
      Update Position
    End While
    Save all Results in array for send to excel.
  End for of i
  Send Results (Runtime and all variables).
End algorithm
    
```

Fig. 1 Pseudocode of PSO algorithm with only traditional processor

same function, the information that is inside the array that was created is sent to a Microsoft Excel sheet.

The PSO in parallel is shown in the following flowchart, are two algorithms briefly explained like the one shown in Fig. 2, but are executed at the same time. As for the communication between the two algorithms, a comparison is made of which of

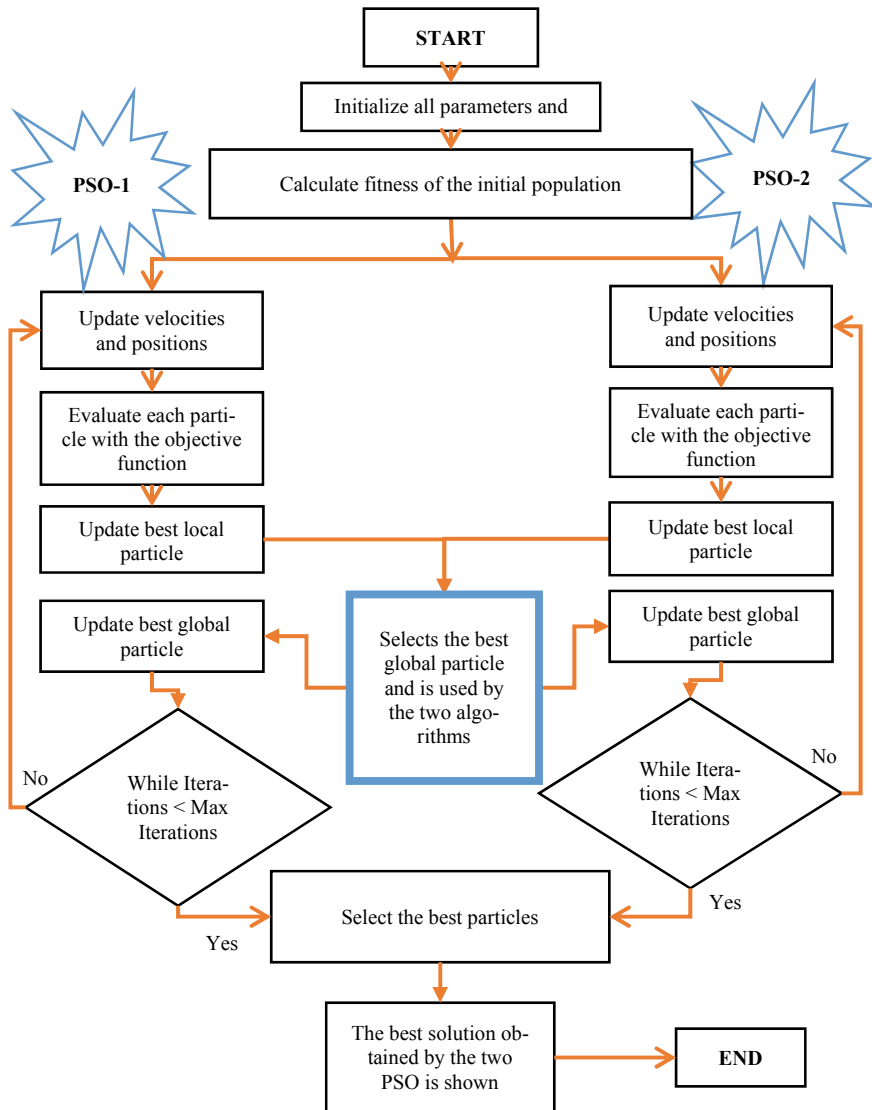


Fig. 2 Flowchart of parallel PSO algorithm

the two algorithms is closest to the objective functions according to the benchmark function that is being used [4, 10, 11].

In Fig. 2, we can find the Parallel PSO algorithm where PSO-1 refers to the algorithm that is dedicated to explore and the PSO-2 is the algorithm that is dedicated to Exploit the search space [12–14]. In the blue box that is almost in the center of the flow chart is the small communication that exist between the two algorithms. When that point reached, a condition is applied, which is that the winner gains share some of their local minimums so that they have more variety and can improve and find a better solution [15–18].

In reference to the programming section and the treatment of the obtained results, it be explained how is the structure to keep all results. Once we have the algorithm that we are going to use, we add a cycle function called “FOR” that encompasses all the algorithm that allows us execute in n-times.

We also need that once they are executed as many times as necessary everything is saved in an arrangement, to send the information to a file, and the workspaces of each of the executions are also saved.

More specifically, each algorithm is executed 270 times, which corresponds to 30 times for each of the 9 benchmark functions that we are using. In this way it allows us to execute or save what we obtained, and thus save time in the data processing.

3 Experimental Results

The results of the performed experiments are presented below, in which each of the results shown will be an average of 30 executions, to be able to use them with a statistical test of z-score.

As previously explained, the results are sent directly to a file, in which an example is shown in Table 3.

Table 4 shows the main parameter values of the parallel particle swarm optimization algorithm:

For the algorithms that were used are the same parameters that the parallel PSO but separately either PSO 1 or 2 [11–14].

Table 3 Example for the file to save information

Run(i)	Best solution	Time	Variables
1	2.036E–11	10	5
...
270	4.134E–12	15	5

Table 4 Parameters used in experiments for parallel particle swarm optimization algorithm

Description	Value
Swarm size	100
Dimensions	5, 10, 20, 40, 80, 160, 320, 640, 1280
Iterations	100
Cognitive parameter for PSO-1	0.065845
Social parameter for PSO-1	0.035954
Cognitive parameter for PSO-2	6
Social parameter for PSO-2	3

3.1 Experimental Results with PSO

The following figures that are presented show the average results obtained in the experiments, which is a comparison between each variant of PSO (in Parallel, to Explore and to Exploit), also different columns are shown which refer to each of the Mathematical benchmark functions that were used [19, 20].

In some graphs, in the part of the bars nothing is shown, it is because the value is very close to zero, but in the same way the values used are described in the lower part of the Figures.

In Fig. 3, it can be seen that the bars corresponding to the values of parallel PSO and the PSO—Exploit do not appear, but it is because the results are equal to zero. In this case, we can note that the Parallel PSO it's better than the other PSO, except for function number 7 that shows the same result for each of the algorithms.

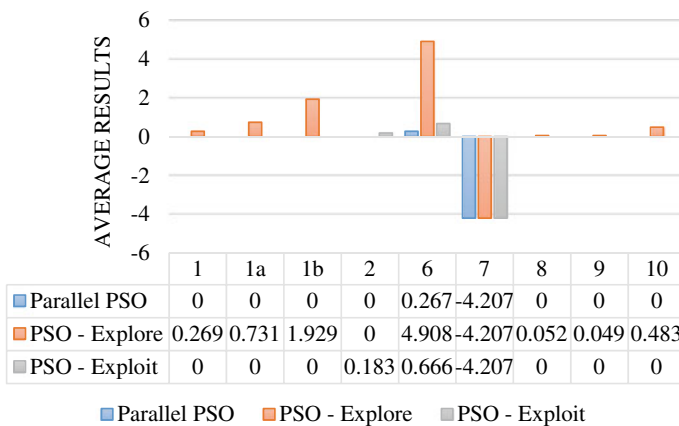


Fig. 3 Comparison of the average results of PSO with each benchmark function for 5 dimensions

In Fig. 4, a comparison of the average results obtained in the experiments working with 10 dimensions is presented, where it is shown that there are higher values for each of the PSOs used, although the results of parallel PSO are better.

Figure 5 shows a very high result and this is why values lower than 1 are not them shown in the graph. But it could be said that the parallel PSO value continues to win in this case.

In Fig. 6, the same value of function “1b” of the previous Fig. 5, remains very high as to what is the PSO to explore with a value of almost 3100, in exchange the other two PSOs are below two, when you are working with 40 dimensions.

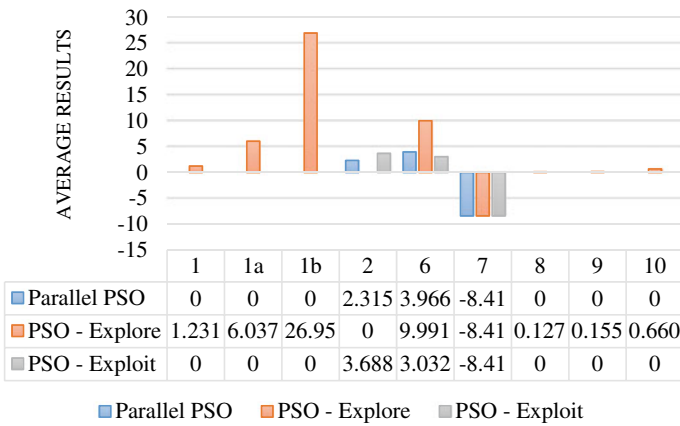


Fig. 4 Comparison of the average results of PSO with each benchmark function for 10 dimensions

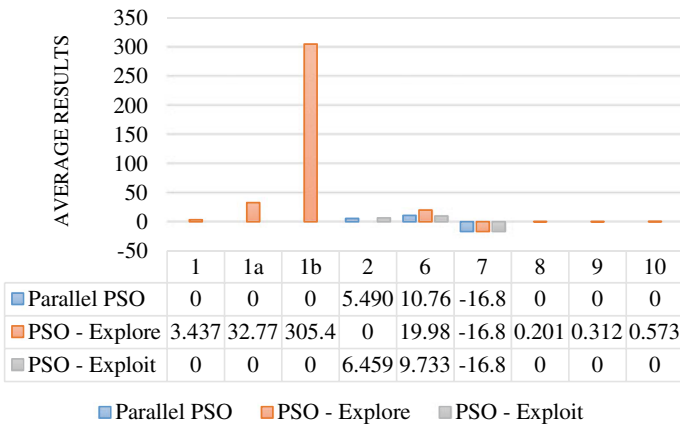


Fig. 5 Comparison of the average results of PSO with each benchmark function for 20 dimensions

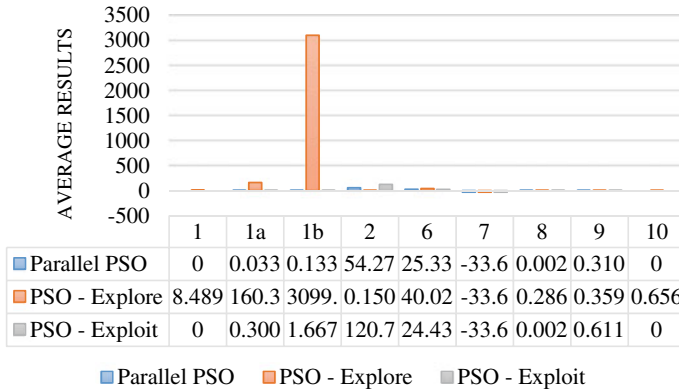


Fig. 6 Comparison of the average results of PSO with each benchmark function for 40 dimensions

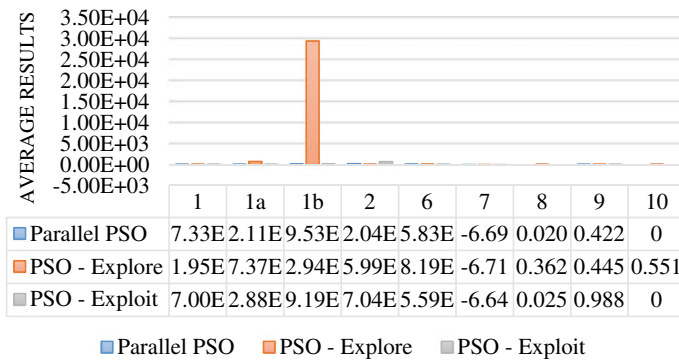


Fig. 7 Comparison of the average results of PSO with each benchmark function for 80 dimensions

In Fig. 7, in this case, it is observed that the PSO to Exploit yields a little better results than the parallel PSO algorithm. But it is a small difference compared to the results obtained with the PSO to explore, and here we work with 80 dimensions.

In Fig. 8, we tested the algorithm with 160 dimensions, therefore the results obtained in some cases are too high. Also no fuzzy system was used for dynamic parameter adaptation of the PSO algorithm. The case of the function “1b” for the PSO to explore still presents a high value in its average result.

Then Fig. 9 shows the results obtained from the different PSO executed using 320 dimensions, although the values are very far from the expected zero, it can be noted that the results for the parallel PSO are a little better than the of the PSO to Exploit.

In Fig. 10, we worked with 640 dimensions and we can note that in the same way as in previous cases the values are not close to zero, but there are differences in the improvement with PSO algorithm in parallel to the comparison of the other two algorithms.

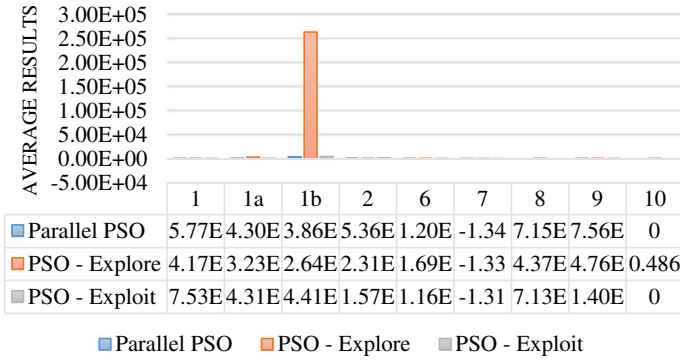


Fig. 8 Comparison of the average results of PSO with each benchmark function for 160 dimensions

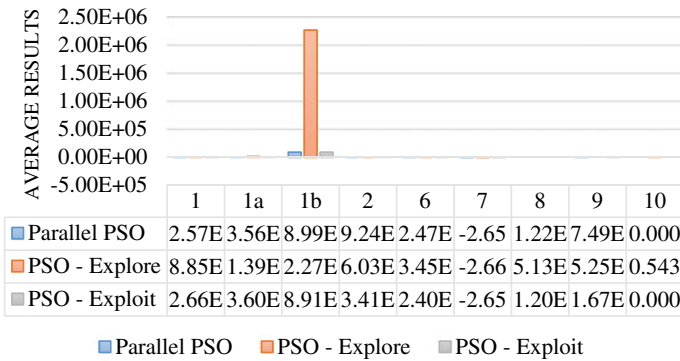


Fig. 9 Comparison of the average results of PSO with each benchmark function for 320 dimensions

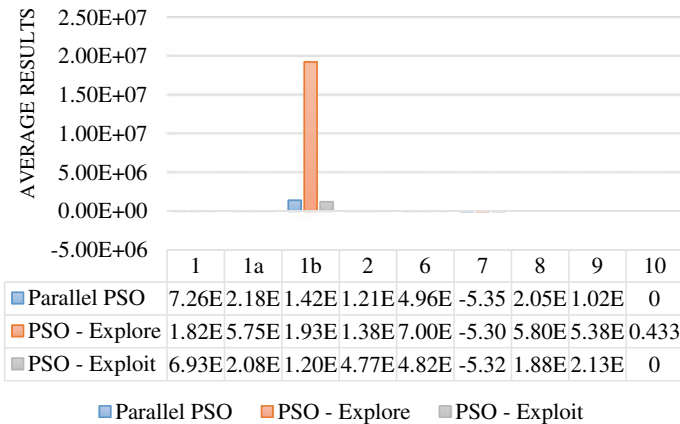


Fig. 10 Comparison of the average results of PSO with each benchmark function for 640 dimensions

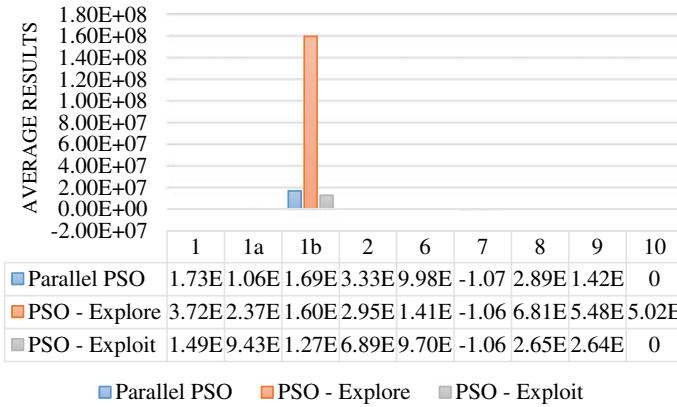


Fig. 11 Comparison of the average results of PSO with each benchmark function for 1280 dimensions

Finally in Fig. 11, we worked with 1280 dimensions and the values obtained show that with PSO in parallel is better in certain functions against the PSO to Exploit and in the same way these two algorithms are better than the algorithm that is to explore.

4 Statistical Test

In this section, each of the statistical tests for the results shown in Sect. 3 will be displayed.

Table 5 shows the main parameters of the statistical test.

The following statistical test prove that in some cases using the Parallel PSO algorithm is better than using it separately. In each of the tables we include the comparison of the two samples for each of the averages of the results in each of the functions used. The results shown from Tables 6, 7, 8, 9, 10, 11, 12, 13, 14 are the Z-score values obtained from the statistical test, and Table 6 refers to Fig. 3.

It is observed that most of the z-scores are significant values for the parallel algorithm versus the PSO variant to explore. Table 7 refers to Fig. 4.

Table 5 Main parameters for statistical test

Parameters	Values
Level of significance	95%
Alpha	5%
H ₀	$\mu_1 \geq \mu_2$
H ₁	$\mu_1 < \mu_2$ (claim)
Critical value	-1.645

Table 6 Statistical test 1: comparison of parallel PSO versus PSO to Explore or PSO to Exploit for 5 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
5	Parallel PSO	Explore	-10.83	-12.35	-13.16	0	-44.33
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	0	0	0	-1.46	-3.82
		Significant evidence	No	No	No	No	Yes
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		0	-13.02	-8.49	-7.73
		Significant evidence		No	Yes	Yes	Yes
		Exploit		0	0	0	0
Significant evidence			No	No	No	No	

Table 7 Statistical test 2: comparison of parallel PSO versus PSO to explore or PSO to Exploit for 10 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
10	Parallel PSO	Explore	-28.30	-27.41	-20.14	3.30	-17.82
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	0	0	0	-1.96	2.76
		Significant evidence	No	No	No	Yes	No
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		0	-23.66	-12.10	-8.71
		Significant evidence		No	Yes	Yes	Yes
		Exploit		0	0	0	0
Significant evidence			No	No	No	No	

Table 8 refers to Fig. 5.

Table 9 refers to Fig. 6.

Table 10 refers to Fig. 7.

Table 11 refers to Fig. 8.

In Table 12 refers to Fig. 9.

Table 13 refers to Fig. 10.

Table 14 refers to Fig. 11.

Table 8 Statistical test 3: comparison of parallel PSO versus PSO to Explore or PSO to Exploit for 20 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
20	Parallel PSO	Explore	-49.37	-52.54	-30.27	2.15	-20.22
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	0	0	0	-0.38	2.27
		Significant evidence	No	No	No	No	No
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		0	-45.76	-19.89	-9.13
		Significant evidence		No	Yes	Yes	Yes
		Exploit		0	0	0	0
Significant evidence			No	No	No	No	

Table 9 Statistical test 4: comparison of parallel PSO versus PSO to Explore or PSO to Exploit for 40 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
40	Parallel PSO	Explore	-83.74	-72.56	-62.10	2.43	-34.56
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	0	-0.12	-0.03	-2.99	2.12
		Significant evidence	No	No	No	Yes	No
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		-2.11	-59.57	-3.06	-8.27
		Significant evidence		Yes	Yes	Yes	Yes
		Exploit		0	0.03	-18.83	0
Significant evidence			No	No	Yes	No	

5 Conclusions

After comparing all the results obtained from the experiments carried out, we can observe that it is suitable for certain benchmark functions where better results are obtained when working in parallel with the two algorithms. Although the results obtained are not the best in the case of comparing them with other similar works in the literature, but there are differences between executing them in parallel where they share their local minimal than when they do each of the algorithms separately

Table 10 Statistical test 5: Comparison of parallel PSO vs PSO to Explore or PSO to Exploit for 80 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
80	Parallel PSO	Explore	-150.85	-89.67	-79.24	2.87	-29.13
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	0.27	-0.96	0.01	-7.25	2.93
		Significant evidence	No	No	No	Yes	No
			F 7	F 8	F 9	F 10	
Parallel PSO	Parallel PSO	Explore		1.58	-85.43	-1.38	-8.43
		Significant evidence		No	Yes	No	Yes
		Exploit		-5.17	-1.20	-33.97	0
		Significant evidence		Yes	No	Yes	No

Table 11 Statistical test 6: Comparison of parallel PSO versus PSO to Explore or PSO to Exploit for 160 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
160	Parallel PSO	Explore	-182.78	-131.15	-98.09	2.77	-60.26
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	-8.98	-0.03	-0.21	-5.57	5.21
		Significant evidence	Yes	No	No	Yes	No
			F 7	F 8	F 9	F 10	
Parallel PSO	Parallel PSO	Explore		-1.01	-114.05	14.77	-7.18
		Significant evidence		No	Yes	No	Yes
		Exploit		-8.24	0.04	-34.07	0
		Significant evidence		Yes	No	Yes	No

where one is dedicated to explore and the other to Exploit. In Sect. 4, the statistical test for each of the tests cases in Sect. 3 are observed. It should be noted that none of the algorithms used have optimization or adaptation of parameters to obtain better results, which is why the results obtained are very far from zero. As future work, we will consider applying the proposed method in other problems, like in [21–28].

Table 12 Statistical test 7: Comparison of parallel PSO versus PSO to Explore or PSO to Exploit for 320 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
320	Parallel PSO	Explore	-281.12	-220.06	-154.02	2.23	-93.04
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	-3.88	-0.93	0.05	-7.61	6.10
		Significant evidence	Yes	No	No	Yes	No
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		0.79	-147.11	9.91	-8.65
		Significant evidence		No	Yes	No	No
		Exploit		0.00	0.76	-40.95	0
Significant evidence			No	No	Yes	No	

Table 13 Statistical test 8: Comparison of parallel PSO vs PSO to Explore or PSO to Exploit for 640 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
640	Parallel PSO	Explore	-488.15	-331.93	-237.92	1.22	-136.41
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	14.60	9.70	2.87	-9.36	8.88
		Significant evidence	No	No	No	Yes	No
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		-10.85	-150.45	18.74	-8.79
		Significant evidence		Yes	Yes	No	Yes
		Exploit		-5.46	7.19	-42.86	0
Significant evidence			Yes	No	Yes	No	

Table 14 Statistical test 9: Comparison of parallel PSO versus PSO to Explore or PSO to Exploit for 1280 dimensions, separated by each of the functions used

Dims	μ_1	μ_2	F 1	F 1a	F 1b	F 2	F 6
1280	Parallel PSO	Explore	-562.71	-329.90	-384.54	1.85	-140.73
		Significant evidence	Yes	Yes	Yes	No	Yes
		Exploit	67.64	29.73	11.07	-8.60	9.49
		Evidence	No	No	No	Yes	No
				F 7	F 8	F 9	F 10
	Parallel PSO	Explore		-7.26	-194.14	46.73	-9.02
		Significant evidence		No	Yes	No	Yes
		Exploit		-6.10	11.93	-66.19	0
Significant evidence			Yes	No	Yes	No	

References

- Colgan, L.: MATLAB in first-year engineering mathematics. *Int. J.Math. Educ. Sci. Technol.* **31**(1), 15–25 (2000)
- Raska, P., Ulrych, Z.: Testing optimization methods on discrete event simulation models and testing functions. *Proc. Eng.* **69**, 768–777 (2014)
- Abiyev, R.H., Tunay, M.: Experimental study of specific benchmarking functions for modified monkey algorithm. *Proc. Comput. Sci.* **102**, 595–602 (2016)
- Uriarte, A., Melin, P., Valdez, F.: An improved particle swarm optimization algorithm applied to benchmark functions. In: 2016 IEEE 8th International Conference on Intelligent Systems (IS), pp. 128–132 (2016)
- Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization: an overview. *Swarm Intell.* **1**(1), 33–57 (2007)
- Valdez, F.: Bio-inspired optimization methods. In: Kacprzyk, J., Pedrycz, W. (eds.) *Springer Handbook of Computational Intelligence*, pp. 1533–1538. Springer, Berlin, Heidelberg (2015)
- Shi, Y., Eberhart, R.C.: Parameter selection in particle swarm optimization. In: *Evolutionary Programming VII*, pp. 591–600 (1998)
- Olivas, F., Valdez, F., Castillo, O.: Particle swarm optimization with dynamic parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions. In 2013 World Congress on Nature and Biologically Inspired Computing, Fargo, ND, USA, pp. 36–40 (2013)
- Kennedy, J.: Particle swarm optimization. In: Sammut, C., Webb, G.I. (eds.) *Encyclopedia of Machine Learning*, pp. 760–766. Springer, US, Boston, MA (2010)
- Eberhart, Shi, Y.: Particle swarm optimization: developments, applications and resources. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*, vol. 1, pp. 81–86, Seoul, South Korea (2001)
- Ghalia, M.B.: Particle swarm optimization with an improved exploration-exploitation balance. In: 2008 51st Midwest Symposium on Circuits and Systems, pp. 759–762, Knoxville, TN, USA (2008)
- Harrison, K.R., Ombuki-Berman, B.M., Engelbrecht, A.P.: Optimal parameter regions for particle swarm optimization algorithms. In: 2017 IEEE Congress on Evolutionary Computation (CEC), pp. 349–356, Donostia, San Sebastián, Spain (2017)
- Lynn, N., Suganthan, P.N.: Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. *Swarm Evol. Comput.* **24**, 11–24 (2015)

14. Zhan, Z., Zhang, J., Li, Y., Chung, H.S.: Adaptive particle swarm optimization. *IEEE Trans. Syst. Man, Cybern. Part B (Cybern.)* **39**(6), 1362–1381 (2009)
15. Ostadmohammadi, B., Arani, P.M., Shariat Panahi, M.: An improved PSO algorithm with a territorial diversity-preserving scheme and enhanced exploration–exploitation balance. *Swarm Evol. Comput.* **11**, 1–15 (2013)
16. Wakasa, Y., Tanaka, K., Nishimura, Y.: Control-theoretic analysis of exploitation and exploration of the PSO algorithm. In: 2010 IEEE International Symposium on Computer-Aided Control System Design, pp. 1807–1812, Yokohama, Japan (2010)
17. Valdez, F., Melin, P., Castillo, O.: Fuzzy control of parameters to dynamically adapt the PSO and GA algorithms. In: International Conference on Fuzzy Systems, Barcelona, Spain, pp. 1–8 (2010)
18. Chen, F., Sun, X., Wei, D., Tang, Y.: Tradeoff strategy between exploration and exploitation for PSO. In: 2011 Seventh International Conference on Natural Computation, Shanghai, China, pp. 1216–1222 (2011)
19. Pohlheim, H.: GEATbx: genetic and evolutionary algorithm toolbox for use with MATLAB version 1.92. User Guide, July 1, 997 (1998)
20. Eberhart, R., Kennedy, J.: A new optimizer using particle swarm theory. In: MHS'95. Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
21. Melin, P., Castillo, O.: Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach. *IEEE Trans. Ind. Electron.* **48**(5), 951–955 (2001)
22. Gonzalez, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
23. Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Appl. Soft Comput.* **53**, 74–87 (2017)
24. Sanchez, M.A., Castillo, O., Castro, J.R., Melin, P.: Fuzzy granular gravitational clustering algorithm for multivariate data. *Inf. Sci.* **279**, 498–511 (2014)
25. Sánchez, D., Melin, P.: Optimization of modular granular neural networks using hierarchical genetic algorithms for human recognition using the ear biometric measure. *Eng. Appl. Artif. Intell.* **27**, 41–56 (2014)
26. Sanchez, M.A., Castro, J.R., Castillo, O., Mendoza, O., Rodriguez-Diaz, A., Melin, P.: Fuzzy higher type information granules from an uncertainty measurement. *Granul. Comput.* **2**(2), 95–103 (2017)
27. Gaxiola, F., Melin, P., Valdez, F., Castro, J.R., Castillo, O.: Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO. *Appl. Soft Comput.* **38**, 860–871 (2016)
28. Castillo, O., Castro, J.R., Melin, P., Rodriguez-Diaz, A.: Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction. *Soft Comput.* **18**(6), 1213–1224 (2014)

Chemical Reaction Algorithm to Control Problems



David de la O, Oscar Castillo and José Soria

Abstract In this paper, we developed an adaptation of the reactions of Chemical Reaction Algorithm (CRA), originally proposed by Astudillo et al. in 2011, which uses fixed parameters in its 4 reactions. We propose a modification to the functions within the chemical reactions, which will help in the optimization of control problems. Using the robot plant “Probot” proposed method show good results in the robot plant.

Keywords Chemical reaction algorithm · Fuzzy · Adaptation · Parameters

1 Introduction

CRA is an optimization algorithm originally proposed by Astudillo [1–4], and this algorithm is basically a static population metaheuristic, which applies an abstraction of chemical reactions as intensifiers (substitution, double substitution reactions) and diversifying (synthesis, decomposition reactions) mechanisms [5, 6].

The performance of fuzzy controllers depends to a large extent on the correct choice of their linguistic control rules and membership function parameters, therefore it is very important to have a technique for adjusting these parameters according to the process that we want to be controlled.

In the literature [7–9] there are many and different possibilities that have been considered to improve the performance of a fuzzy controller, one of which is the adjustment of the fuzzy controller parameters [10, 11]. Thus, a fuzzy controller could be obtained from the knowledge of an expert and then be tuned by automatic optimization techniques [12–17].

When fuzzy systems and optimization algorithms are combined, a synergy is created in which, the main contribution of fuzzy logic is what can be called the computation with fuzzy rules, while the optimization algorithms contribute with a systematic random search methodology.

D. de la O · O. Castillo (✉) · J. Soria
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: ocastillo@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_16

185

The present work focuses on the collaborative use of CRA, and fuzzy logic to solve a control problem and the main contribution of the paper is the modification of the internal operation of the reactions to improve the optimization of the parameters of a fuzzy controller [17–19].

The rest of the paper is organized into four sections as follows: Sect. 1 the theory underlying the present work. Sections 2 and 3 describe the main problem and recent research work is presented, in which issues such as fuzzy logic, CRA algorithm and the operation of an autonomous mobile robot are discussed. Section 4 shows the results of the simulations. Section 5 offers the conclusion.

2 The Chemical Optimization Paradigm

The chemical reaction algorithm is a metaheuristic that performs a stochastic search for optimal solutions in which we do not need to define the search space. This optimization algorithm uses an element (or compound) to represent the solution, and the fitness of the element is evaluated in accordance with the objective function of the problem to be solved [1–3].

This algorithm is a very straightforward methodology that emulates the characteristics of the chemical reactions (synthesis, decomposition, substitution, and double-substitution) to search for the optimal solution [3] to a problem.

This approach is basically a static population metaheuristic, which applies an abstraction of chemical reactions as intensifiers (substitution, double substitution reactions) and diversifying (synthesis, decomposition reactions) mechanisms. The elitist reinsertion strategy allows the permanence of the best elements and thus the average fitness of the entire element pool increases with each iteration. Figure 1 shows the flowchart of the algorithm.

The main components of this chemical reaction algorithm are described below.

The synthesis and decomposition reactions are used to diversify the resulting solutions; these procedures prove to be highly effective and rapidly lead the results to the desired value.

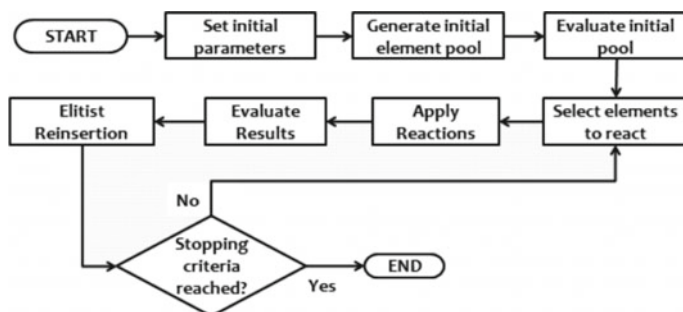


Fig. 1 General flowchart of the chemical reaction algorithm

The single and double substitution reactions allow the algorithm to search for optimal values around a good previously found solution and they are described as follows.

The algorithm works mainly with chemical reactions with a change in at least one substance (element or possible solution), that changes its composition and property sets. Such reactions are classified into 4 types, which are described below.

2.1 Type 1: Combination Reactions



A combination reaction is a reaction of two reactants to produce one product. The simplest combination reactions are the reactions of two elements to form a compound. After all, if two elements are treated with each other, they can either react or not.

2.2 Type 2: Decomposition Reactions



The second type of simple reaction is decomposition. This reaction is also easy to recognize. Typically, only one reactant is given. A type of energy, such as heat or electricity, may also be indicated. The reactant usually decomposes to its elements, to an element and a simpler compound, or to two simpler compounds.

Binary compounds may yield two elements or an element and a simpler compound. Ternary (three-element) compounds may yield an element and a compound or two simpler compounds (see Fig. 2).

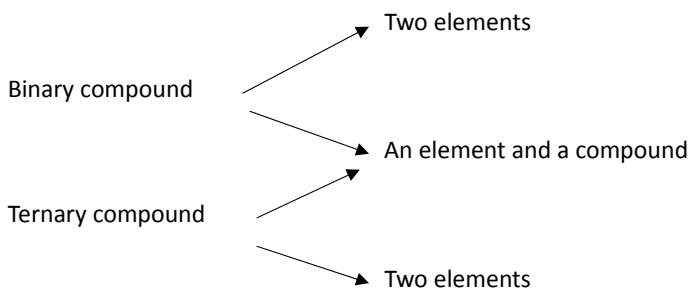


Fig. 2 Decomposition possibilities

2.3 *Type 3: Substitution Reactions*



Elements have varying abilities to combine. Among the most reactive metals are the alkali metals and the alkaline earth metals. On the opposite end of the scale of reactivity's, among the least active metals or the most stable metals are silver and gold, prized for their lack of reactivity. Reactive means the opposite of stable but means the same as active.

When a free element reacts with a compound of different elements, the free element will replace one of the elements in the compound if the free element is more reactive than the element it replaces. In general, a free metal will replace the metal in the compound, or a free nonmetal will replace the nonmetal in the compound. A new compound and a new free element are produced.

2.4 *Type 4: Double-Substitution Reactions*

Double-substitution or double-replacement reactions also called double-decomposition reactions or metathesis reactions, involve two ionic compounds, most often in aqueous solution [1].

In this type of reaction, the cations simply swap anions. The reaction proceeds if a solid or a covalent compound is formed from ions in solution. All gases at room temperature are covalent. Some reactions of ionic solids plus ions in solution also occur. Otherwise, no reaction takes place.

Just as with replacement reactions, double-replacement reactions may or may not proceed. They need a driving force.

In replacement reactions the driving force is reactivity; here it is insolubility or covalence.

At first sight, chemical theory and definitions may seem complex and none or few are re-lated to optimization theory, but only the general schema will be considered here in order to focus on the control application.

The flowchart for this optimization method can be found in Fig. 1, and the following list of steps is presented:

- We start by generating an initial set of elements/compounds.
- We evaluate the original set of elements, to obtain fitness values.
- Based on the above evaluation, we select some of the elements/compounds to "induce" a reaction.
- Taking into consideration the result of the reaction, evaluations of the news element/compounds are obtained and selected elements are those of greater fitness
- Repeat the steps until the algorithm meets the terminating criteria (the desired result in the maximum number of iterations is reached) [3].

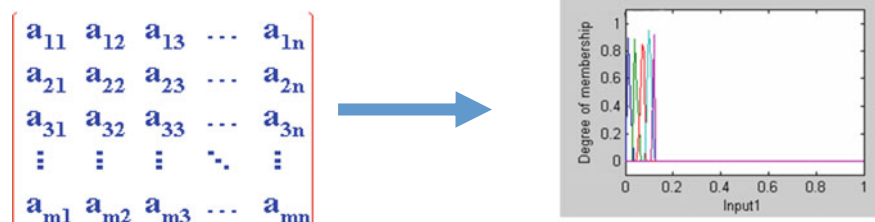
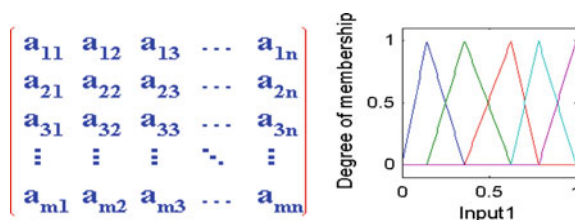


Fig. 3 Representation of elements and membership functions

Fig. 4 Representation of elements and membership functions with proposed method



3 Methodology

The algorithm internally represents a possible solution to the problem through a vector, the size of the vector is determined by the dimension of the problem, we can use n possible solutions, which would turn into a matrix of elements, as possible solutions to the problem. When applying chemical reactions the algorithm uses a scalar value, X , between 0 and 1, which multiplies the element matrix. The result of multiplying a matrix by a scalar value, between 0 and 1, is the reduction of each value, as each element of the matrix represents, a value of the membership function, this makes the membership to shorten its width, therefore increasing the iterations makes the membership functions are all on the left side, with very small values (see Fig. 3).

To solve this problem, we change the operations of the chemical reactions, which will generate an inverse matrix of random numbers, this ensures a randomness in all the elements of the matrix (see Fig. 4), which will later be translated into the values of the parameters of the membership function.

4 Simulation Results

To validate the proposed approach, we performed some tests with the robot plant, the results is show in Table 1. The plant simulated a robot with two wheels controlled and one free wheels.

The media value is $3.15E-07$ and standard deviation is $1.83E-07$, the best values es $8.29E-10$ and poor value is $7.49E-07$.

Table 1 Results of the simulation

Test	Fitness
1	3.70E-07
2	3.55E-07
3	4.01E-07
4	9.53E-08
5	4.01E-07
6	1.81E-07
7	1.65E-07
8	1.81E-07
9	2.44E-07
10	5.07E-07
11	3.31E-07
12	8.29E-10
13	5.07E-07
14	3.60E-07
15	1.70E-09
16	3.73E-07
17	2.26E-07
18	7.49E-07
19	4.97E-07
20	3.60E-07

Fig. 5 Plot of the simulation

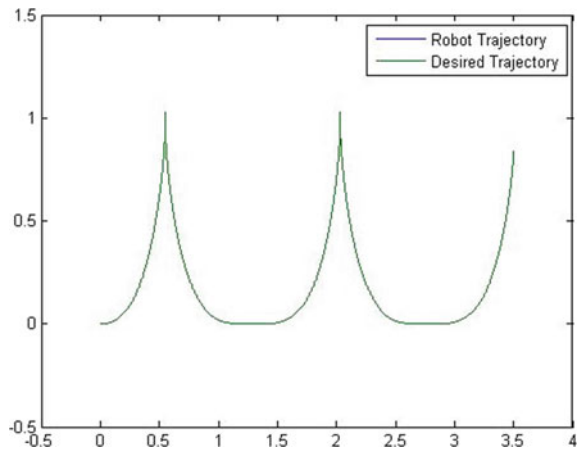
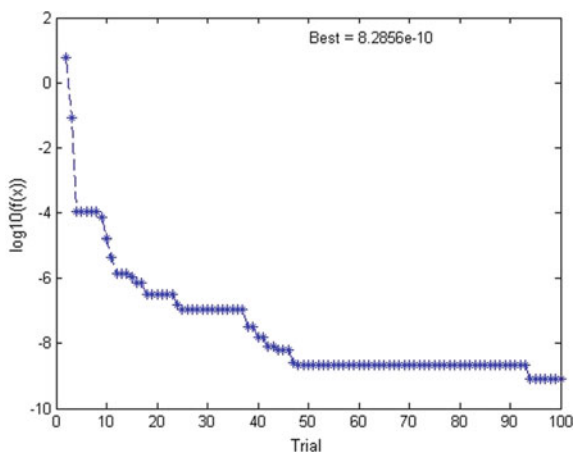


Figure 5 show the results of simulation, where we can see the robot trajectory is similar to desired trajectory. The difference between trajectories is very small.

The figure show the convergency of the simulation (Fig. 6).

Fig. 6 Plot of the convergency



5 Conclusions

This paper proposed a adjust to the functions of chemical reactions, for improving its performance CRA for optimization problem of control. We improve this algorithm by using an adaptation of the chemical reactions. Our proposed method show good results in the test on the plant. As future work we can apply the proposed method to other applications, like in [20–27].

References

1. Astudillo, L., Melin, P., Castillo, O.: Nature optimization applied to design a type-2 fuzzy controller for an autonomous mobile robot. In: 2012 Fourth World Congress on de Nature and Biologically Inspired Computing (NaBIC) (2012)
2. Melin, P., Astudillo, L., Castillo, O., Valdez, F., Garcia, M.: Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm. *Expert Syst. Appl.* **40**(8), 3185–3195 (2013)
3. Astudillo, L., Melin, P., Castillo, O.: Introduction to an optimization algorithm based on the chemical reactions. *Inf. Sci.* **291**, 85–95 (2015)
4. Sanchez, C., Melin, P., Astudillo, L.: Chemical optimization method for modular neural networks applied in emotion classification. In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, vol. 547, pp. 381–390. Springer, Berlin (2014)
5. de la O.D., Castillo, O., Melendez, A., Astudillo, L.: Optimization of a reactive controller for mobile robots based on CRA. In: Annual Conference of the North American Fuzzy Information Processing Society (NAFIPS) held jointly with 2015 5th World Conference on Soft Computing (WConSC), IEEE, pp. 1–6, Aug 2015
6. de la, O.D., Castillo, O., Astudillo, L., Soria, J.: Fuzzy chemical reaction algorithm with dynamic adaptation of parameters. In: Melin, P., Castillo, O., Kacprzyk, J., Reformat, M., Melek, W. (eds.) *Fuzzy Logic in Intelligent System Design*, pp. 122–130. Springer, Cham (2018)

7. Olivas, F., Valdez, F., Castillo, O.: An interval type-2 fuzzy logic system for dynamic parameter adaptation in particle swarm optimization. In: 2014 IEEE Conference on Norbert Wiener in the 21st Century (21CW), vol. 1, no. 6, pp. 24–26, June 2014
8. Amezcua, J., Melin, P.: Optimization of modular neural networks with the LVQ algorithm for classification of arrhythmias using particle swarm optimization. In: Recent Advances on Hybrid Approaches for Designing Intelligent Systems, pp. 307–314 (2014)
9. Valdez, F., Melin, P., Castillo, O.: A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation. *Expert Syst. Appl.* **41**(14), 6459–6466 (2014)
10. Cervantes, L., Castillo, O.: Design of a fuzzy system for the longitudinal control of an F-14 airplane. In: Soft Computing for Intelligent Control and Mobile Robotics, pp. 213–224 (2011)
11. de la, O.D., Castillo, O., Soria, J.: Optimization of reactive control for mobile robots based on the CRA using type-2 fuzzy logic. *Stud. Comput. Intell.* **667**(1), 505–518 (2017)
12. Wang, D., Wang, G., Hu, R.: Parameters optimization of fuzzy controller based on PSO. In: 3rd International Conference on Intelligent System and Knowledge Engineering, ISKE 2008, vol. 1, pp. 599, 603, 17–19 Nov 2008
13. Esmín, A.A.A., Aoki, A.R., Lambert-Torres, G.: Particle swarm optimization for fuzzy membership functions optimization. In: 2002 IEEE International Conference on Systems, Man and Cybernetics, vol. 3, p. 6. 6–9 Oct 2002
14. Fierro, R., Castillo, O.: Design of fuzzy control systems with different PSO variants. *Recent Advances on Hybrid Intelligent Systems*, pp. 81–88 (2013)
15. Fang, G., Kwok, N.M., Quang, H.: Automatic fuzzy membership function tuning using the particle swarm optimization. In: Pacific-Asia Workshop on Computational Intelligence and Industrial Application, PACIIA '08, vol. 2, pp. 324, 328, 19–20 Dec 2008
16. Li, H.X., Gatland, H.B.: A new methodology for designing a fuzzy logic controller. *IEEE Trans. Syst. Man Cybern.* **25**(3), 505–512 (1995)
17. Martínez, R., Castillo, O., Soria, J.: Particle swarm optimization applied to the design of type-1 and type-2 fuzzy controllers for an autonomous Mobile Robot. In: Bio-inspired Hybrid Intelligent Systems for Image Analysis and Pattern Recognition, pp. 247–262 (2009)
18. Melendez, A., Castillo, O.: Optimization of type-2 fuzzy reactive controllers for an autonomous mobile robot. In: Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC), pp. 207–211 (2012)
19. Melendez, A., Castillo, O.: Evolutionary optimization of the fuzzy integra-tor in a navigation system for a mobile robot. In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) Recent Advances on Hybrid Intelligent Systems, volume 451 of Studies in Computational Intelligence, pp. 21–31. Springer, Berlin, Heidelberg (2013)
20. Melin, P., Castillo, O.: Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach. *IEEE Trans. Industr. Electron.* **48**(5), 951–955 (2001)
21. Gonzalez, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
22. Olivas, F., Valdez, F., Castillo, O., Gonzalez, C.I., Martinez, G., Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. *Appl. Soft Comput.* **53**, 74–87 (2017)
23. Sanchez, M.A., Castillo, O., Castro, J.R., Melin, P.: Fuzzy granular gravitational clustering algorithm for multivariate data. *Inf. Sci.* **279**, 498–511 (2014)
24. Sánchez, D., Melin, P.: Optimization of modular granular neural networks using hierarchical genetic algorithms for human recognition using the ear biometric measure. *Eng. Appl. Artif. Intell.* **27**, 41–56 (2014)
25. Sanchez, M.A., Castro, J.R., Castillo, O., Mendoza, O., Rodriguez-Diaz, A., Melin, P.: Fuzzy higher type information granules from an uncertainty measurement. *Granul. Comput.* **2**(2), 95–103 (2017)
26. Gaxiola, F., Melin, P., Valdez, F., Castro, J.R., Castillo, O.: Optimization of type-2 fuzzy weights in backpropagation learning for neural networks using GAs and PSO. *Appl. Soft Comput.* **38**, 860–871 (2016)

27. Castillo, O., Castro, J.R., Melin, P., Rodriguez-Diaz, A.: Application of interval type-2 fuzzy neural networks in non-linear identification and time series prediction. *Soft. Comput.* **18**(6), 1213–1224 (2014)

AMOSA with Analytical Tuning Parameters and Fuzzy Logic Controller for Heterogeneous Computing Scheduling Problem



Héctor J. Fraire Huacuja, Carlos Soto, Bernabé Dorronsoro,
Claudia Gómez Santillán, Nelson Rangel Valdez
and Fausto Balderas-Jaramillo

Abstract In this chapter, an analytical parameter tuning for the Archive Multi-Objective Simulated Annealing (AMOSA) with a fuzzy logic controller is proposed. The analytical tuning is used to compute the initial and final temperature, as well as the maximum metropolis length. The fuzzy logic controller is used to adjust the metropolis length for each temperature. These algorithms are used to solve the Heterogeneous Computing Scheduling Problem. The tuned AMOSA with a fuzzy logic controller is compared against an AMOSA without tuning. Three quality indicators are used to compare the performance of the algorithms, these quality indicators are hypervolume, generational distance, and generalized spread. The experimental results show that the tuned AMOSA with fuzzy logic controller achieves the best performance.

Keywords Heterogeneous computing scheduling problem · Makespan · Energy · Fuzzy logic controller · Multi-objective simulated annealing

H. J. Fraire Huacuja (✉) · B. Dorronsoro · C. G. Santillán · N. R. Valdez · F. Balderas-Jaramillo
Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Mexico City, Mexico
e-mail: automatas2002@yahoo.com.mx

B. Dorronsoro
e-mail: bernabe.dorronsoro@uca.es

C. G. Santillán
e-mail: claudia.gomez@itcm.edu.mx

N. R. Valdez
e-mail: nelson.rangel@itcm.edu.mx

F. Balderas-Jaramillo
e-mail: fausto.balderas@itcm.edu.mx

C. Soto
Universidad de Cádiz, Cádiz, Spain
e-mail: carlos.sotomonterrubio@alum.uca.es

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_17

1 Introduction

In this chapter, the Heterogeneous Computing Scheduling Problem (HCSP) with independent tasks is approached. This problem is addressed because of the importance that energy awareness has acquired in high-performance computing centers during the last years. In 1992, the efficient energy-aware consumption was started by the creation of the Energy Star program [1]. This program searches to improve energy efficiency in all kind of electrical products. It is applied in data centers and distributed systems because of its contribution to the well-being of the environment. As is shown in [2], in 2006 the consumption of energy by servers and data centers in the United States was around 61 billion of kilowatt-hour and the cost of this energy was around 4.5 billion dollars. In November 2008, there was a total of 14 High-Performance Computing Centers (HPCC) which energy consumption was higher than 1 MW. The number of HPPC in June 2013 was 43. This figure for the USA and Europe was almost 41 in 2014. The National University of Defense Technology (NUDT) in China developed a supercomputer with 3,120,000 cores with a performance of 33.9 Pflops and energy consumption of 17.9 MW [3]. Although more cores provide more computing power this has strong implications in energy consumption and thermal conditions. The energy used to run applications in the HPCC, the economic costs and heat problems are continuously increased.

The HCSP with energy consists of minimizes the *makespan* and the energy consumption required by the machines. To achieve these objectives, machines with DVFS (Dynamic Voltage and Frequency Scaling) capabilities are used. These machines can work with different speeds and frequencies, allowing managing energy consumption during the execution of allocated tasks. DVFS is an energy management technique in computer architectures that allows increasing or decreasing the voltage applied [4]. DVFS is normally used in laptops and other mobiles devices as a battery energy saver. An example of this capability happens when a device is in standby in that case the device's processor works at the minimum speed extending the battery life. This problem can be modeled as a scheduling problem, as is done in this chapter.

Currently, several methods exists to solve the Scheduling problem like Non-sorted Genetic Algorithm—II (NSGA-II), Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D), Simulated Annealing (SA), Tabu Search, Greedy Randomized Adaptive Search Procedure (GRASP), and even exact models for *makespan* minimization [5–9]. Some heuristics rules have been created among the metaheuristics to explore the solutions space. The most common rules are min—min, max—min, largest job on fastest resource—shortest job on fastest resource and suffrage [7, 10].

In this chapter, an offline and online tuning parameter for the AMOSA is developed. An offline analytical procedure performs the tuning parameter and an online tuning is done with a fuzzy logic controller (FLC). The remaining content of this chapter is organized as follows. Section 2 describes the addressed problem. Section 3 shows SA's characteristics and analytical tuning. The AMOSA with the analytical

tuning method and the fuzzy logic controller are described in Sect. 4. Experimentation and results are shown in Sect. 5. Section 6 presents the conclusions and the research lines for future works.

2 Problem Description

The research problem HCSP is found in heterogeneous computing systems, where a set of machines are used to execute large loads of works. The machines have the dynamic voltage and frequency scaling technology, are heterogeneous with multiple computing capabilities, and tasks without precedence restrictions are received. DVFS technology means different voltage configurations in machines at different speeds.

Given a set of tasks $T = \{t_1, t_2, \dots, t_n\}$, a set of heterogeneous machines $M = \{m_1, m_2, \dots, m_k\}$, the execution times of every task in every machine are $P = \{p_{1,1}, p_{1,2}, \dots, p_{n,k}\}$. Since machines are DVFS capable and heterogeneous, different voltage levels exist for each machine m_j with a relative speed associated. It is denoted as $V_j = \{v_{i,1}, v_{i,2}, \dots, v_{j,l}\} \forall m_j \in M$ of different l sizes. When the highest voltage is selected speed is equal to 1 (the normal execution time in P), with a lower voltage selected the relative speed decrease for example when *speed* equal to 0.5 (50% of the normal speed).

The model consists of binary variables $x_{i,j,k}$ which is 1 if the task j is allocated in machine i with a voltage configuration k , otherwise it is 0. The objectives considered are in Eq. (1), where f_1 is the makespan and f_2 is the energy. This model is subject to Eqs. (2–5). Equation 2 allows that a task j can only be allocated in a machine i with a voltage configuration k . Equation (2) is the computation of makespan. Equation (4) indicates that each variable x is binary. The values of y are in the set of positive real number (Eq. 5).

$$\min \vec{f} = \left\{ y, \sum_{i \in M} \sum_{j \in T} \sum_{k \in K} \frac{P_{i,j}}{v_{k,j}} V_{k,j}^2 x_{i,j,k} \right\} \quad (1)$$

subject to:

$$\sum_{i \in M} \sum_{k \in K} x_{i,j,k} = 1, \quad \forall j \in T \quad (2)$$

$$\sum_{j \in T} \sum_{k \in K} \frac{P_{i,j}}{v_{k,j}} x_{i,j,k} \leq y, \quad \forall i \in M \quad (3)$$

$$x_{i,j,k} \in \{0, 1\} \quad (4)$$

$$y \in \{\mathbb{R}^+\} \quad (5)$$

where k is the index of the selected voltage in V_j .

The MOP studied in this chapter consists of finding the assignation of machine/voltage/task that minimizes (1). As we mentioned before, this problem has important practical applications but is not an easy task; as is well known, even the scheduling problem with only an objective function belongs to the NP-hard class problem [11]. Thus, heuristic methods should be applied, and Simulated Annealing is one of the most successful for this class of problems.

3 Simulated Annealing

The Simulated Annealing algorithm was proposed by Kirkpatrick [12]. SA is an analogy of the annealing of solids, its fundamentals came from a physical area known as statistical mechanics. In the algorithm, a set of solutions are used to simulate the particles of the solid. The particles are moving from one position to another due to the internal energy and temperature of the system. This energy is the Gibbs energy and the solid reaches the optimal state when this energy is minimal. The internal energy can be modified because the particles change their position and/or the temperature is changed. The situation is far to be simple because for a given temperature the particles change their position which in turn modify the Gibbs energy and the temperature. The process continues until a thermal equilibrium is reached; then, a new temperature is achieved, and the process is started again. In this process, everything happens as if the particles were able to continuously change their states by the action of an intelligent algorithm.

The final state of the solid is the solution to the problem, and the energy is the objective function. The movement of the particles of the solid corresponds to a solution perturbation. The temperature is a control parameter and the thermal equilibrium is the solution of the heuristic. The movement of a solid is simulated with a perturbation function which can be a genetic operator or another function that modifies the solution. The changes from one state to another are accepted by the Boltzmann probability function [13] as in Eq. (6); that means that bad states are accepted by the algorithm to escape from local optima.

$$p_{xy}(T) = \min \left\{ 1; e^{\left(\frac{f(x) - f(y)}{T} \right)} \right\} \quad (6)$$

The main SA's parameters and characteristics are the following:

- **Initial temperature (T_i):** Defines the start of the algorithm. Two situations should be taken into account to define the best initial temperature:
 - If the initial temperature is too high, the algorithm could spend a lot of time in the first iterations.
 - If the initial temperature is too low, the probability to be trapped in local optimum is very high.

- **Final temperature (T_f):** This value defines the stop criterion, and now two situations should be avoided:
 - If a final temperature is too small, the algorithm can be last a lot of time.
 - If the final temperature is too high, the algorithm can finish very soon and the optimal cannot be reached.
- **Cooling scheme:** It is a function which specifies how the temperature is decreased. If the temperature is decreased very slowly, the quality solution is increased, but more execution time is required. The most common is the geometric scheme: $T_{k+1} = \alpha T_k$.
- **Markov's chain or metropolis length:** This is the number of iterations carried out in each temperature T_k . If the temperature decreases very slowly, the quality solution is increased, but a lot of execution time is required.
- **Acceptance criterion:** This is a probability function to accept a new solution given a previous one. In classical SA, the Boltzmann distribution defined by Eq. (6) is used. This acceptance criterion uses energy differences from the current solution and the new solution.

The tuning in SA is important because it helps to improve the search exploration at low temperatures and high temperatures. This improvement is done by setting an adequate metropolis length at each temperature and setting the best initial and final temperatures. Another advantage of the parameters tuning in SA is to avoid the wasting of time at very high and very low temperatures.

Parameter tuning can be done by performing an analytical analysis [14]. The analytical tuning allows adjusting the initial and final temperature, and the Markov's chain length as well. The tuning is done with the following analysis. At high temperatures, the probability to accept a new solution is close to one. The initial temperature (T_i) is associated with the maximum deterioration permitted and the acceptance criterion. Let S_i be the current solution and S_j the new solution, $Z(S_i)$ and $Z(S_j)$ are the associated costs to S_i and S_j . ΔZ_{\max} and ΔZ_{\min} give the maximum and minimum deterioration, the probability of accepting a new solution $P_A(S_j)$ with a maximum deterioration is 1 and T_i can be obtained by Eq. (7):

$$T_i = -\frac{\Delta Z_{\max}}{\ln(P_A(\Delta Z_{\max}))} \quad (7)$$

similarly, the final temperature is associated with the probability to accept a new solution with the minimum deterioration $P_A(\Delta Z_{\min})$:

$$T_f = -\frac{\Delta Z_{\min}}{\ln(P_A(\Delta Z_{\min}))} \quad (8)$$

considering n as the number of steps from T_i to T_f then β is calculated by the next equation:

$$\beta = e^{\frac{\ln L_{\max} - \ln L_i}{n}} \quad (9)$$

The Markov's chain length (L_k) has the following features: (a) at high temperatures few iterations are required because the stochastic equilibrium is reached rapidly, (b) at low temperatures a more exhaustive exploration is needed, so a bigger L_k is used. The Markov's length can be calculated by:

$$L_{\max} = N = C |V_{S_i}| \quad (10)$$

where V_{S_i} is the neighborhood of S_i . In [14] the authors said that to guarantee a good level of exploration in the neighborhood the value of C is established between $1 \leq C \leq 4.6$.

4 Multi-objective Simulated Annealing

The first adaptation of SA to multi-objective optimization was proposed in [13]. This first adaptation consists in the adjustment of the acceptance criterion to combine all the objective functions using aggregative functions. More aggregative functions are proposed in [15], including the weighted sum, weighted product, and scalar linear.

The main differences between SA and MOSA are the computation of the energy and the use of a set of solutions which compose the Pareto front. The energy function is required in the acceptance criterion. Multiple versions of MOSA have been proposed in the last years; one of them is the AMOSA [16].

Figure 1 shows the general MOSA algorithm based on non-dominance. Line 1, the algorithm receives the initial temperature, the final temperature, and the α value. In Line 2, the temperature is initialized. An empty set (F) of solutions is initialized. A random solution is generated in Line 4. Line 5 indicates the stop criterion. Then, at Line 6 the metropolis cycle begins. Inside this cycle (Line 7) a neighbor is generated. If the neighbor dominates the current solution then the new solution becomes the current solution and it is inserted in F ; if the current solution dominates it then the Boltzmann probability is computed in Line 12; otherwise, the solutions are non-dominated, and it is inserted in F . In Line 18, the geometric cooling scheme is applied to reduce the temperature.

In Fig. 1, the Pareto dominance is performed with three possible cases in the evaluation. These cases are in Line 8, if the S_i dominates S_j then S_i is accepted and takes a place in the Pareto front. In Line 11, if S_i dominates S_j then in Line 12, obtaining the energy with an aggregative function the Boltzmann probability is used to accept S_i . In line 15, if both solutions are non-dominated, then the new solution is accepted.

Input: Initial temperature T_i , final temperature T_f , and cooling rate α .
Output: Non-dominated solutions in F .

```

1: procedure MULTI-OBJECTIVE SIMULATED ANNEALING( $T_i, T_f, \alpha$ )
2:    $T_k \leftarrow T_i$ 
3:    $F \leftarrow \emptyset$ 
4:    $S_i \leftarrow \text{generateRandomSolution}()$ 
5:   while  $T_k \geq T_f$  do
6:     while Metropolis length do
7:        $S_j \leftarrow \text{perturbation}(S_i)$ 
8:       if  $S_j \prec S_i$  then
9:          $S_i \leftarrow S_j$ 
10:        update  $F$  with  $S_i$ 
11:       else if  $S_i \prec S_j$  then
12:         if  $e^{\frac{-\Delta(E(S_j) - E(S_i))}{T_k}} < \text{random}(0, 1)$  then
13:            $S_i \leftarrow S_j$ 
14:           update  $F$  with  $S_i$ 
15:         else ▷ If are non-dominated
16:            $S_i \leftarrow S_j$ 
17:           update  $F$  with  $S_i$ 
18:        $T_{k+1} = \alpha T_k$ 
19:   return Get non-dominated solutions from  $F$ .

```

Fig. 1 Multi-objective simulated annealing

4.1 AMOSA

The AMOSA is a variation of MOSA proposed in [16]. One of its features is the use of an archive. Also, it includes the amount of dominance concept to computing the energy between two solutions. A clustering algorithm is used to maintain diversity in the archive.

The main features of AMOSA are: (a) the use of an archive to store all the non-dominated solutions, this archive has two limits known as hard limit (HL) and soft limit (SL), (b) the amount of dominance (ΔDom) concept to represent the area between two solutions, (c) the use of the single linkage clustering algorithm to reduce the size of the archive to HL when the SL size is reached, this clustering algorithm keeps the solutions with more diversity [17]. The amount of dominance allows identifying high and low dominated solutions. It is computed with Eq. (10).

$$\Delta Dom_{a,b} = \prod_{i=1, f_i(a) \neq f_i(b)}^M \frac{|f_i(a) - f_i(b)|}{R_i} \quad (11)$$

The amount of dominance is also used in criterion acceptance as the energy difference between the two solutions. AMOSA is capable to solve problems with 15 objective functions. AMOSA considers three cases that may occur in dominance:

1. The current solution dominates the new solution; in this case, the acceptance criterion is applied (See Fig. 5 from lines 9 to 15).
2. The current solution and the new solution are non-dominated between them; this case is subdivided in three cases:
 - (a) If the new solution is dominated by more than one solution in the archive, then the acceptance criterion is applied with the average amount of dominance.
 - (b) If the new solution is non-dominated concerning all the solutions in the archive, then the new solution is added to the archive.
 - (c) If the new solution dominates more than one solution in the archive, then the new solution is added to the archive and the dominated solutions are removed.
3. The new solution dominates the current solution, this case considers the three same subcases as point 2, but in the acceptance criterion, the minimum amount of dominance is considered.

The analytical tuning for AMOSA is done as in SA but the amount of dominance function is used in the maximum and minimum deterioration functions.

4.2 AMOSA with Fuzzy Logic Controller

The AMOSA with Fuzzy logic controller (AMOSA-FLC) considers the average amount of dominance and the current temperature to adjust the increment or decrement of β . This algorithm includes an analytical tuning to obtain T_i, T_f , and L_{\max} . The analytical tuning is performed offline while the FLC is online after each metropolis cycle. The fuzzy logic controller considers the average amount of dominance of the accepted solutions in each metropolis. The fuzzy logic controller receives two parameters normalized: the temperature and the average amount of dominance; returning the increment of β . The Gaussian membership function is used in the temperature, metropolis length, and $\Delta\beta$.

The membership function for the average amount of dominance with values ranging from 0 to 1 is shown in Fig. 2. This variable is divided into three Gaussian functions from low energy to high energy.

Figure 3 shows the temperature membership functions with ranges between 0 and 1. This variable is divided into five Gaussian functions from very low to very high.

The output membership function for $\Delta\beta$ is shown in Fig. 4. This variable is divided into five Gaussian functions from a high decrease to high increase. The output value is in the range from -0.2 to 0.2 ; this range was set from previous experimentation.

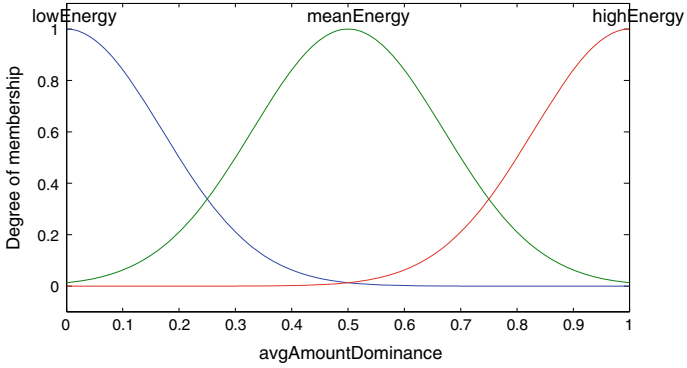


Fig. 2 Input membership function for the average amount of dominance

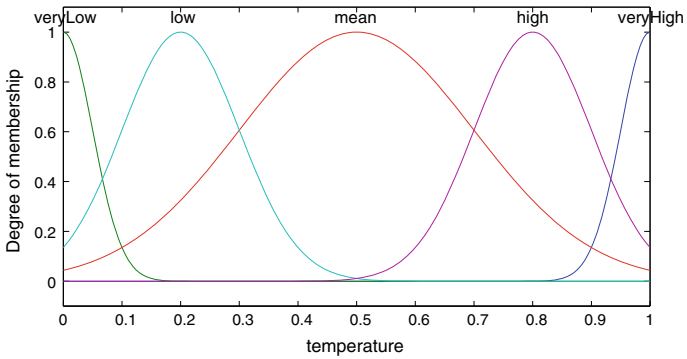


Fig. 3 Input membership function for the current temperature

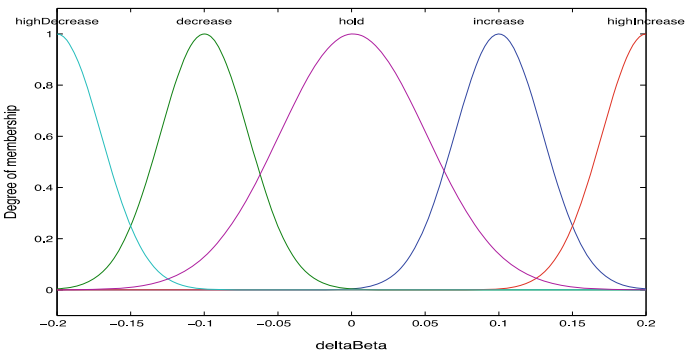


Fig. 4 Output membership function for $\Delta\beta$

Table 1 Fuzzy logic controller rules

T_k	ΔDom average	$\Delta\beta$
Very high	High	Decrease
Very high	Medium	High decrease
Very high	Low	High decrease
High	High	Decrease
High	Medium	High decrease
High	Low	High decrease
Medium	High	Increase
Medium	Medium	Hold
Medium	Low	Increase
Low	High	Increase
Low	Medium	High decrease
Low	Low	Increase
Very low	High	Hold
Very low	Medium	Increase
Very low	Low	Increase

The rules used in the fuzzy logic controller are shown in Table 1. The first column shows the values of the temperature from very high to very low. The ΔDom average is shown in the second column; this average takes values from high to low. The third column indicates the increase or decrease levels for β values. These rules indicate if β should be increased or decreased, thus affecting the metropolis length.

These rules were built with the purpose of increasing the metropolis length when there are lower differences of energy because there are higher chances of improvement; on the other hand, reducing the metropolis length when there are higher differences because there are fewer chances of improvement [18, 19].

Figure 5 show the AMOSA-FLC algorithm. The changes to the original AMOSA are: Line 2, where T_i , T_f , β , and L_{\max} are initialized with their corresponding equations; Line 40, where L_{k+1} is updated with the fuzzy logic controller.

5 Computational Experiments

The objective of the experiments is to measure the performance of the implemented algorithms. The characteristics measured are closeness to the Pareto front, diversity and the spread of the Pareto front generated by each algorithm. The non-parametric Wilcoxon test is performed to asses if exists a significant statistical difference.

The experiments consists of 30 independent runs with an instance set from [20]. AMOSA uses the following parameters: Initial population = 100, $T_i = 4000$, $T_f = 10$, $\alpha = 0.95$, $L = 30$, $HL = 100$ and $SL = 150$. In AMOSA-FLC the parameters

Input: Archive hard limit HL , archive soft limit SL , and cooling rate α .

Output: Non-dominated solutions from archive.

```

1: procedure AMOSA-FLC( $HL, SL, \alpha$ )
2:   initialize  $T_i, T_f, \beta, L_{\max}$  with Equations (7), (8), (9), and (10).
3:    $T_k = T_i$ 
4:    $L_k = T_{\max}$ 
5:    $S_{cur} = \text{random}(\text{Archive})$   $\triangleright$  Randomly chosen from the Archive
6:   while  $T_k \geq T_f$  do
7:     while Metropolis length ( $L_k$ ) do
8:        $S_{new} = \text{perturbation}(S_i)$ 
9:       if  $S_{cur} \prec S_{new}$  then  $\triangleright$  Case 1
10:         $\Delta dom_{avg} = \frac{\sum_{i=1}^k \Delta dom_{i, S_{new}} + \Delta dom_{S_{cur}, S_{new}}}{k+1}$ 
11:        if  $\frac{1}{1+e^{\Delta dom_{avg} * T_k}} < \text{random}(0, 1)$  then
12:           $S_{cur} = S_{new}$ 
13:        if  $S_{cur} \not\prec S_{new}$  and  $S_{new} \not\prec S_{cur}$  then  $\triangleright$  Case 2
14:          if  $S_{new} \succ S_N \in \text{Archive}$  then  $\triangleright$  Case 2 (a)
15:             $\Delta dom_{avg} = \frac{\sum_{i=1}^k \Delta dom_{S_{cur}, S_{new}}}{k}$ 
16:            if  $\frac{1}{1+e^{-\Delta dom_{avg} * T_k}} < \text{random}(0, 1)$  then
17:               $S_{cur} = S_{new}$ 
18:            if  $S_{new} \not\prec \forall S_N \in \text{Archive}$  then  $\triangleright$  Case 2 (b)
19:               $S_{cur} = S_{new}$ 
20:               $\text{Archive.add}(S_{new})$ 
21:              Cluster  $\text{Archive}$  to  $HL$ 
22:            if  $S_{cur} \prec S_N \in \text{Archive}$  then  $\triangleright$  Case 2 (c)
23:               $S_{cur} = S_{new}$ ;
24:              Remove the  $N$  dominates points from the  $\text{Archive}$ 
25:          if  $S_{new} \prec S_{cur}$  then  $\triangleright$  Case 3
26:            if  $S_{new} \succ S_N \in \text{Archive}$  then  $\triangleright$  Case 3 (a)
27:               $\Delta dom_{min} = \min \{ \Delta dom_{new, N} \}$ 
28:              if  $\frac{1}{1+e^{-\Delta dom_{min}}} < \text{random}(0, 1)$  then
29:                 $S_{curr} = \text{point which corresponds to } \Delta dom_{min}$ 
30:              else
31:                 $S_{curr} = S_{new}$ 
32:              if  $S_{new} \not\prec \forall S_N \in \text{Archive}$  then  $\triangleright$  Case 3 (b)
33:                 $S_{curr} = S_{new}$ 
34:                 $\text{Archive.add}(S_{curr})$ 
35:                Cluster  $\text{Archive}$  to  $HL$ 
36:              if  $S_{new} \prec S_N \in \text{Archive}$  then  $\triangleright$  Case 3 (c)
37:                 $S_{curr} = S_{new}$ 
38:                 $\text{Archive.add}(S_{curr})$ 
39:                Remove all the  $N$  dominated points from the  $\text{Archive}$ 
40:               $L_{k+1} = (\beta + FIS(T_k, \Delta dom_{avg}))L_k$ 
41:               $T_{k+1} = \alpha T_k$ 
42:              Cluster  $\text{Archive}$  to  $HL$ 
43:            return  $\text{Archive}$ 

```

Fig. 5 Archive multi-objective simulated annealing

Table 2 Quality indicators results

Algorithm	HV	GD	GS
AMOSA	0.204896	0.228508	0.883791
AMOS-FLC	0.297508	0.118607	0.965100

Table 3 p -values computed by Wilcoxon test

Algorithms	HV	GD	GS
AMOSA versus AMOSA-FLC	0.1499	0.1139	0.0008

are: Initial population = 100, $P_A(\Delta Z_{\min}) = 0.10$, $P_A(\Delta Z_{\max}) = 0.90$, $C = 2$, $\alpha = 0.95$, $HL = 100$, and $SL = 150$. The polynomial mutation is used as a perturbation for all the algorithms. The metropolis length cannot exceed L_{\max} to avoid high values in metropolis length.

The algorithms' performance considers the solutions quality, the spread of the front, and the diversity of solutions. To measure these characteristics three quality indicators are used: hypervolume (HV), generational distance (GD), and generalized spread (GS) [21]. Table 2 shows the experimentation results. The first column contains the algorithm's name. The second column contains the average HV. The third column shows the average GD. The fourth column contains the average spread. The best results in each quality indicator are in bold font.

Table 2 shows that the AMOSA-FLC has better results than AMOSA in HV and GD but has the worst performance in the spread. The AMOSA has the best result in the spread.

The p -values obtained by the Wilcoxon test are shown in Table 3; notice that if the p -value is equal or lower to 0.05. The null hypothesis is rejected and a significant statistical difference between both algorithms is accepted. The first column indicates the two algorithms that are being compared. The second, third, and fourth columns show the p -values obtained for each quality indicator.

6 Conclusions and Future Work

In this chapter, a new fuzzy logic controller for online tuning the metropolis length is presented. The results show that AMOSA-FLC has the best performance in HV and GD. A significant statistical difference in GS can be observed between the performances of AMOSA-FLC and AMOSA. This shows that the parameter tuning achieves the best performance in this algorithm with this problem. Two methods to tune the AMOSA algorithm were applied. The results show that AMOSA-FLC has better quality than AMOSA.

The implementation of diverse rules and new variables, using standard deviation and variances. Besides, the use of variable length chains using special methods to determine the dynamic equilibrium in real time should be analyzed. Additionally, experimentation with diverse functions in the maximum and minimum deterioration

is planned; for example, the number of dominated solutions needs to be taken into account. Hybrid AMOSAs with other heuristics as genetic algorithms should be implemented. Finally, parallel programming technology for AMOSA is one of the future works in this research.

Acknowledgements This project was partially supported by the following Consejo Nacional de Ciencia y Tecnología (CONACyT) and Tecnológico Nacional de México projects: (a) Consolidation National Lab Project [280712]; (b) Project [3058] from the program Catedras CONACyT; (c) Project [6684.18P] TNM. C. Soto and H. Fraire would like to thank CONACyT and Tecnológico Nacional de México for their financial support under the contracts [414092, 254498, 60 02.16-P] and [280 081] (Redes Temáticas).

References

1. Energy, S.: Energy Star. Hist. Energy Star (2011)
2. Liu, Y., Zhu, H.: A survey of the research on power management techniques for high-performance systems. *Softw. Pract. Exp.* **40**(11), 943–964 (2010)
3. TOP500.: The 43rd top500 list published during isc14 in Leipzig (2014)
4. Magklis, G., Semeraro, G., Albonesi, D.H., Dropsho, S.G., Dwarkadas, S., Scott, M.L.: Dynamic frequency and voltage scaling for a multiple-clock-domain microprocessor. *IEEE Micro* **23**(6), 62–68 (2003)
5. Pooranian, Z., a Harounabadi, A., Shojafar, M., Hedayat, N.: New hybrid algorithm for task scheduling in grid computing to decrease missed task. *World Acad. Sci. Eng. Technol.* **55**(7), 924–928 (2011)
6. Pooranian, Z., Shojafar, M., Javadi, B.: Independent task scheduling in grid computing based on queen-bee algorithm. *Int. J. Artif. Intell.* **1**(4), 171 (2012)
7. Chaturvedi, A.K., Sahu, R.: New heuristic for scheduling of independent tasks in computational grid. *Int. J. Grid Distrib. Comput.* **4**(3), 25–36 (2011)
8. Raj, R.J.S., Vasudevan, V.: Beyond simulated annealing in grid scheduling. *Int. J. Comput. Sci. Eng.* **3**(3), 1312–1318 (2011)
9. Pecero, J.E., Bouvry, P., Huacuja, H.J.F., Khan, S.U.: A multi-objective GRASP algorithm for joint optimization of energy consumption and schedule length of precedence-constrained applications. In: 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, pp. 510–517 (2011)
10. Braun, T.D., Siegel, H.J., Beck, N., Bölöni, L.L., Maheswaran, M., Reuther, A.I., Robertson, J.P., Theys, M.D., Yao, B., Hensgen, D., Freund, R.F.: A comparison of eleven static heuristics for mapping a class of independent tasks onto heterogeneous distributed computing systems. *J. Parallel Distrib. Comput.* **61**(6), 810–837 (2001)
11. Garey, M.A., Johnson, S.D.: *Computers and intractability: a guide to the theory of NP-completeness*. Freeman (1979)
12. Kirkpatrick, S.: Optimization by simulated annealing: quantitative studies. *J. Stat. Phys.* **34**(5–6), 975–986 (1984)
13. Serafini, P.: Simulated annealing for multi objective optimization problems. In: *Multiple Criteria Decision Making*. Springer, New York, NY, pp. 283–292 (1994)
14. Frausto-Solís, J., Sanvicente-Sánchez, H., Imperial-Valenzuela, F.: ANDYMARK: an analytical method to establish dynamically the length of the markov chain in simulated annealing for the satisfiability problem. In: *Simulated Evolution and Learning*, Springer, pp. 269–276 (2006)
15. Nam, D., Park, C.H.: Multiobjective simulated annealing: a comparative study to evolutionary algorithms. *Int. J. Fuzzy Syst.* **2**(2), 87–97 (2000)

16. Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K.: A simulated annealing-based multiobjective optimization algorithm: AMOSA. *IEEE Trans. Evol. Comput.* **12**(3), 269–283 (2008)
17. Jain, A.K., Dubes, R.C.: *Algorithms for Clustering Data*. Prentice-Hall, Inc. (1988)
18. Soto, C., Valdez, F., Castillo, O.: A review of dynamic parameter adaptation methods for the firefly algorithm. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) *Nature-Inspired Design of Hybrid Intelligent Systems*. *Studies in Computational Intelligence*, vol. 667. Springer, Cham (2017)
19. Valdez, F., Melin, P.O., Castillo, O.: A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation. *Expert Syst. Appl.* **41**(14), 6459–6466 (2014)
20. Fraire Huacuja, H.J., Gonzalez Barbosa, J.J., Bouvry, P., Pineda, A.A.S., Pecero, J.E.: An iterative local search algorithm for scheduling precedence-constrained applications on heterogeneous machines. In: *6th Multidisciplinary International Conference Schedule Theory Application (MISTA 2013)*, pp. 47–485 (2010)
21. Wu, J., Azarm, S.: Metrics for quality assessment of a multiobjective design optimization solution set. *J. Mech. Des.* **123**(1), 18 (2001)

Medical Applications

A Modular Neural Network Approach for Cardiac Arrhythmia Classification



Eduardo Ramírez, Patricia Melin and German Prado-Arechiga

Abstract In this work we describe a modular neural network approach to use expert modules as a classification model for 12-lead cardiac arrhythmias. The modular neural network is designed using Multilayer perceptron as classifiers. This modular neural network was trained and tested with the Physikalisch-Technische Bundesanstalt diagnostic ECG database (PTB database) of physioBank. The electrocardiograms are preprocessed to improve their classification through the proposed modular neural network. This modular neural network uses the features extracted of each signal such as autoregressive model coefficients, Shannon entropy and multifractal wavelets. We used the twelve electrode signals or leads included in the PTB database, such as *i*, *ii*, *iii*, *avf*, *avr*, *avl*, *v1*, *v2*, *v3*, *v4*, *v5*, *v6*, *vx*, *vy* and *vz*. The modular neural network is composed by twelve expert modules, where each module is used to perform the classification for the specific signal lead. The expert modules are based on the following models: multilayer perceptron with scaled conjugate gradient backpropagation (MLP-SCG). Finally, the outputs from the expert modules are combined using winner-takes-all integration as modular neural network integration method.

Keywords Modular neural network · Multilayer perceptron · 12-lead arrhythmia classification

E. Ramírez · P. Melin (✉) · G. Prado-Arechiga
Tijuana Institute of Technology, Graduate Studies, Tijuana, BC, Mexico
e-mail: pmelin@tectijuana.mx

E. Ramírez
e-mail: eduraf@hotmai.com

G. Prado-Arechiga
e-mail: ogprado@hotmai.com

1 Introduction

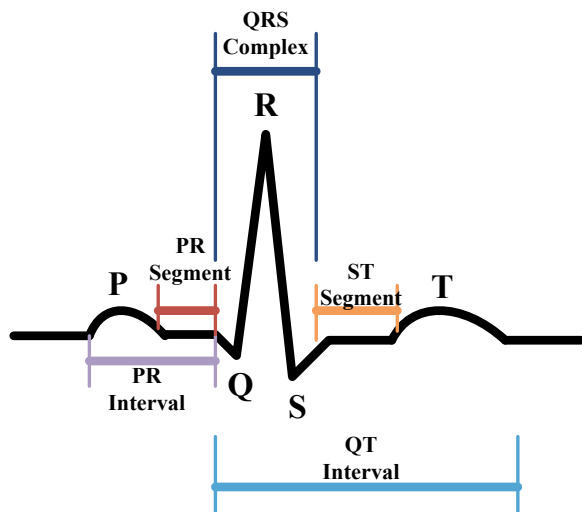
Different approaches have been proposed to try to interpret the electrocardiographic patterns related to diseases that are diagnosed through the electrocardiograms or ECGs. We can mention some of these approaches: Bayesian, linear discriminant systems, statistical and syntactic pattern recognition [1], self-organizing maps, Markov models, learning vector quantization [2, 3], support vector machines [4–6], higher order statistics, multilayer perceptron [5, 7, 8–19], expert systems, fuzzy logic [13, 18], type-2 fuzzy logic [20–25], and hybrid systems [5, 12, 13, 26–31]. Other solutions are based on feature extraction such as PCA, LDA, ICA and Discrete Wavelet Transform [32–36].

The Electrocardiogram or ECG shows the electrical activity of the heart from several different projections as a graphical representation or electrical waves series. These projections focus on the different regions of the heart. The electrical activity of the heart gives important information about health and pathology of a person. The ECG signals are related to a complex chemical, electrical, and mechanical process in heart. The ECG are one among the most important source of diagnostic information. In Fig. 1, we illustrate the normal heartbeat with intervals, segments and fiducial points [37–40].

The standard ECG is referred to as a 12-Lead ECG using 10 electrodes. Three of the leads are simply the results of comparing electrical potentials recorded by two electrodes. The conventional 12 leads are i, ii, iii, avf, avr, avl, v1, v2, v3, v4, v5, v6 [37–40].

We can define cardiac arrhythmia as the alteration in the activity of the heart rhythm, in amplitude, duration or shape of the rhythm. An arrhythmia is an abnormal heart rhythm [37–40].

Fig. 1 Normal heartbeat with intervals, segments and fiducial points



The remaining content of this research is presented as follows. The hybrid intelligent system, proposed method and problem statement are described in Sect. 2. Background theory is presented in Sect. 3. In Sect. 4, we are presented the results of the experiments performed. Finally, in Sect. 5, the conclusions and future works are brief shown. References and acknowledgements are listed to end of this chapter.

2 Problem Statement and Proposed Method

This work is related to cardiac arrhythmia classification using the Physikalisch-Technische Bundesanstalt (PTB) Diagnostic ECG database [7, 41] achieved through the hybridization of two computational intelligence techniques such as fuzzy logic and artificial neural networks. The purpose is design a hybrid intelligent system to be able combine different computational intelligence techniques considering the 12 electrode signals of an electrocardiogram to classify cardiac arrhythmias to support a complete medical diagnosis.

We extracted different samples of heartbeats from the PTB Diagnostic ECG database. The selected samples of heartbeats were preprocessing used autoregressive model coefficients, Shannon entropy and multifractal wavelets. The classes that were included in the PTB ECG database are Myocardial Infarction, Cardiomyopathy, Bundle branch block, Dysrhythmia, Myocardial hypertrophy, Valvular heart disease, Myocarditis, Miscellaneous and Healthy controls.

We selected 211 electrocardiograms of different patients. We are working on 2 classes: myocardial infarction and health controls, 133 samples of myocardial infarction and 78 of healthy controls. The hybrid intelligent system learns the 2 classes using samples of each class. We used 70% for training and 30% for testing.

In this work, we used the conventional twelve leads: i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5 and v6 included in the electrocardiograms of the PTB Diagnostic ECG database, in Figs. 2, 3, 4 and 5, we show some examples of the convention leads. The electrocardiograms used for the classes were belonging of the patients 1 to 103, 108, 111, 120, 128, 138 to 142, 145, 148, 149, 152, 158, 163, 183, 193, 195, 197, 207, 211, 223, 226, 230, 231, 259, 261, 265, 268, 270, 273, 274, 280, 282, 283, 287, 290 to 294 for myocardial infarction and 104, 105, 116, 117, 121, 122, 131, 155, 156, 165, 166, 169 to 174, 180, 182, 184, 185, 198, 214, 229, 233 to 248, 251, 252, 255, 260, 263, 264, 266, 267, 276, 277, 279, 284 for healthy controls, in some cases more than one electrocardiogram of the same patient was used available in the PTB diagnosis ECG database.

The modular neural network is composed for twelve expert modules, where each module is related to an electrode signal or lead in the electrocardiogram being 12-lead cardiac arrhythmia classification solution. We used winner-takes-all method to determine the global classification for the modular neural network using the twelve expert modules. In Fig. 6, we are presented the general architecture of the proposed modular neural network.

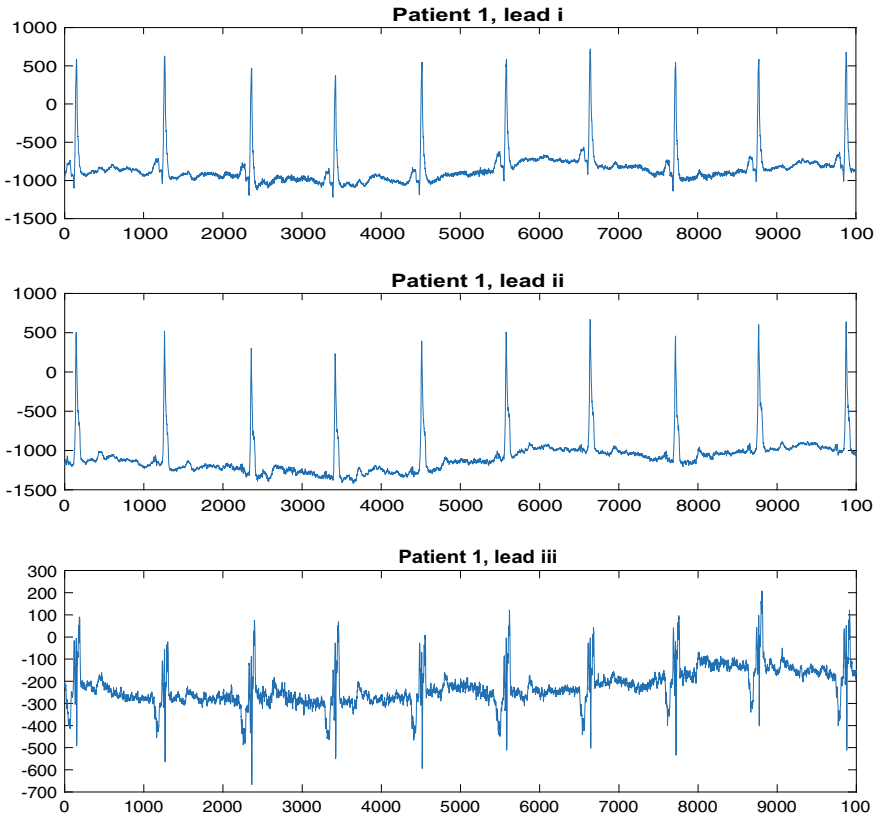


Fig. 2 The conventional leads i, ii, iii in electrocardiograms of the PTB diagnostic database

In the preprocessing phase, we applied a feature extraction process in order to reduce or simply the information of the selected samples of the electrocardiograms as well as for the classifiers captures the differences between the classes to improve the classification rate. We created a set of vectors that represent a complete signal or lead in the electrocardiogram; in other words, each electrocardiogram has its twelve feature vectors to be learning for the expert modules in the proposed modular neural network. The feature vectors are built with autoregressive models, Shannon entropies.

In the following part of this section, we describe a brief summary with some important concepts referent of the techniques used in this proposed work.

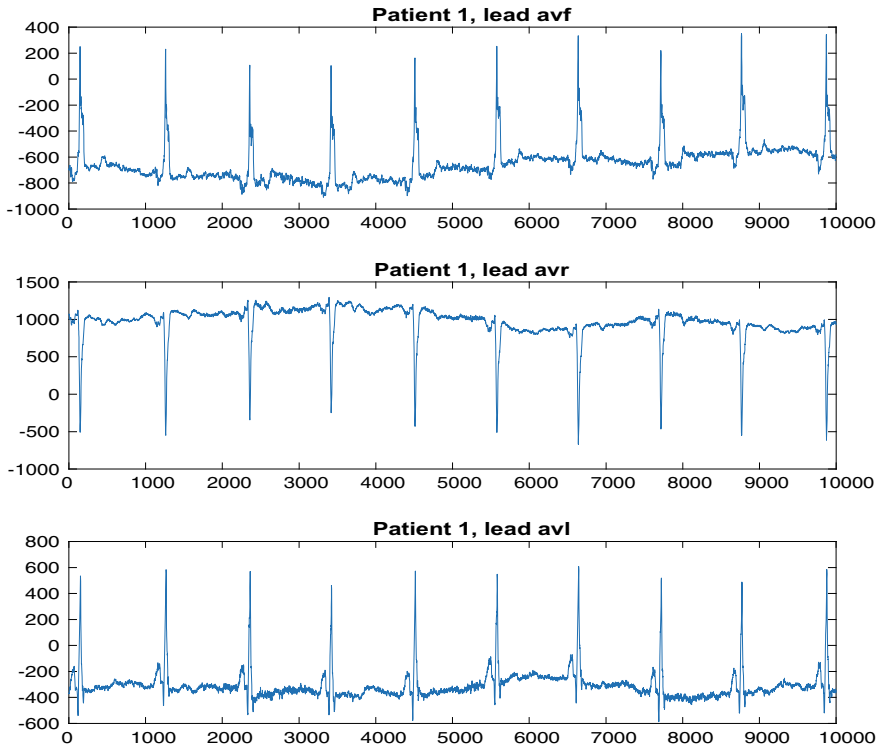


Fig. 3 The conventional leads avf, avr, avl in electrocardiograms of the PTB diagnostic database

3 Important Concept Review

In this section, we describe important concepts such as Multi Layer Perceptron, autoregressive models, Shannon entropy and wavelets.

3.1 Multi-layer Perceptron

An Artificial Neural Network (ANN) is a mathematical and computational model that simulates the abstract structure and functional aspects of biological neural networks. The basic computational elements in an ANN are known as neurons, nodes serve as inputs, outputs or internal processing units. These neurons communicate and pass signals among themselves using what are known as synapses, which neurons are organized and connected define the architecture of the network. The Multi-Layer Perceptron (MLP), the information moves in one direction, forward, from inputs nodes, through the hidden nodes to outputs nodes. The Scaled Conjugate Gradient

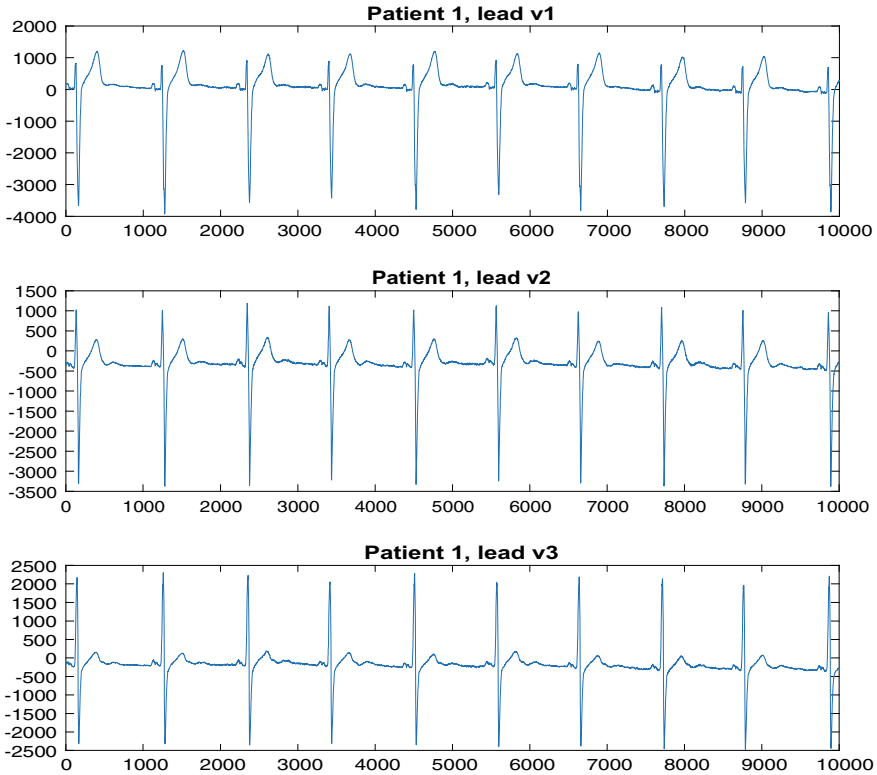


Fig. 4 The conventional leads v1, v2, v3 in electrocardiograms of the PTB diagnostic database

(SCG) is used to more efficiently guide the error down the error surface and provide faster solutions for an extra second derivative of error information and internal adjustments that are made to the learning parameters [42].

3.2 Autoregressive Models, Shannon Entropy and Multifractal Analysis Wavelets

Many observed time series exhibit serial autocorrelation; that is, linear association between lagged observations. This suggests past observations might predict current observations. The autoregressive process models the conventional mean of y_t as a function of past observations, $y_{t-1}, y_{t-2}, \dots, y_{t-p}$. An AR process that depends on p past observations is called an AR model of degree p , denoted by AR(p).

Information entropy is the average rate at which information is produced by a stochastic source of data. The measure of information entropy associated with each possible data value is the negative logarithm of the probability mass function for the

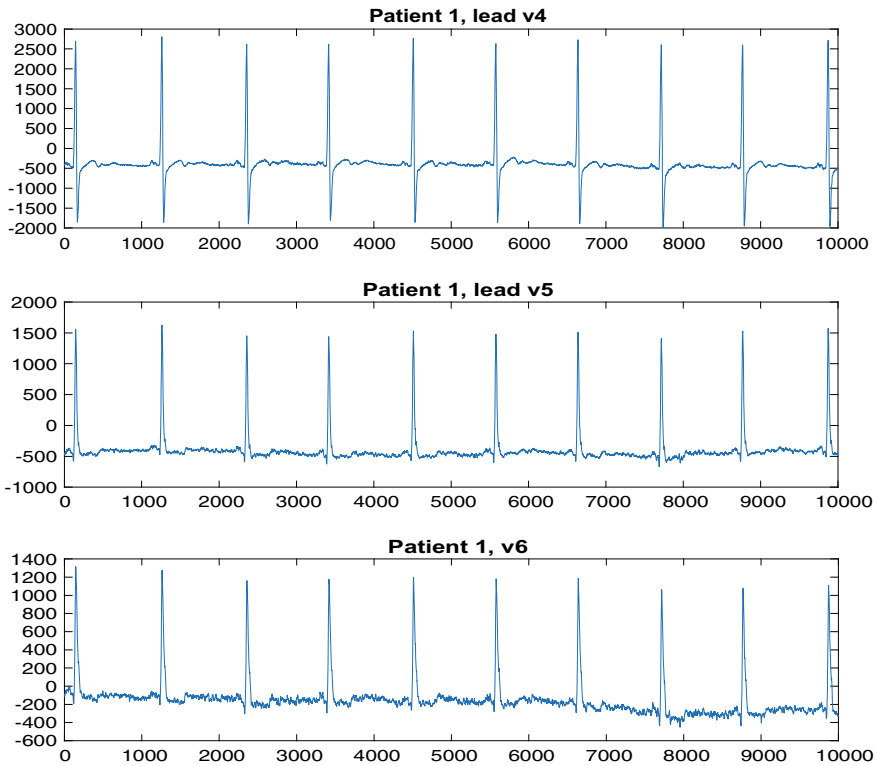


Fig. 5 The conventional leads v4, v5, v6 in electrocardiograms of the PTB diagnostic database

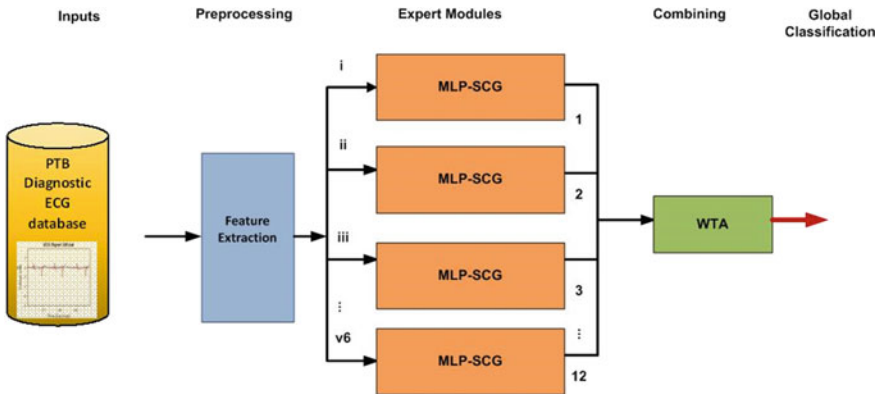


Fig. 6 Modular neural network for 12-lead arrhythmia classification

value, see Eq. 1.

$$S = - \sum_i P_i \log P_i \quad (1)$$

when the data source has a lower probability value, the event carries more information than when the source data has a higher probability value. Generally, entropy refers to disorder or uncertainty, and the definition of entropy used in information theory is directly analogous to the definition used in statistical thermodynamics. The concept of information entropy was introduced by Claude Shannon.

Multifractal analysis is quasi-systematically performed using the coefficients of continuous or discrete wavelet transforms. Wavelet coefficients consist of quantities that mostly concentrate around 0, rendering the numerical computation of negative q moments extremely unstable or even theoretically infinite [43].

Recently, an alternative approach has been proposed the Wavelet Leader (WL). This method is theoretically backed up by a strong mathematical framework. Also, its being defined from an Discrete Wavelet Transform (DWT) [43].

4 Experiments

We performed experiments used 70% of the selected electrocardiograms for training the expert modules in proposed modular neural network and 30% for testing. We have trained classifiers base on MLP-SCG to select the better that represent each expert module by lead or signal electrode from electrocardiogram. The parameters for the MLP-SCG structures were 50, 100 and 150 hidden neurons, 190 input neurons, 2 output neurons, 10,000 epochs, learning rate 0.001. Where the output neurons are the classes used to train the modular neural network such as Myocardial Infarction and Healthy Controls. We selected the better trained classifier in order to form the expert module representation in modular neural network. The outputs of the selected expert modules were integrated used winner-takes-all method to determine the global classification of the proposed modular neural network.

4.1 PTB Diagnostic ECG Database

The PTB Diagnostic ECG Database contains 549 ECG records from 290 subjects, the subjects aged 17 to 87 years old. Each subject is represented by one to five ECG records. Each ECG record includes 15 measured signals, the conventional 12 leads: i, ii, iii, avr, avl, avf, v1, v2, v3, v4, v5 and v6 and 3 Frank lead such as v_x , v_y and v_z . The diagnosis classes included in the database are Myocardial Infarction, Cardiomyopathy, Bundle branch block, Dysrhythmia, Myocardial hypertrophy, Valvular heart disease, Myocarditis, Miscellaneous, Healthy controls [7, 41].

4.2 Results

4.2.1 Expert Modules. Classifiers: MLP-SCG

Firstly, we present the results for the classifiers used to form the expert modules in the proposed modular neural network.

The results of the classifiers above mentioned in experiment section are show in Table 1. We can observe the better results represented for each lead which were selected to form part of the expert modules in the modular neural network. The best result for lead *i* was used MLP-SCG with 150 hidden neurons, for lead *ii* MLP-SCG with 50 hidden neurons, for *iii* used MLP-SCG with 150 hidden neurons, for *avf* MLP-SCG with 150 hidden neurons, for *avr* used MLP-SCG with 50 hidden neurons, for *avl* used MLP-SCG with 150 hidden neurons, for *v1* used MLP-SCG with 150 hidden neurons, for *v2* used MLP-SCG with 100 hidden neurons, for *v3* used MLP-SCG with 50 hidden neurons, for *v4* used MLP-SCG with 150 hidden neurons, for *v5* used MLP-SCG with 150 hidden neurons and for *v6* used MLP-SCG with 150 hidden neurons.

In Fig. 7 is presented the comparative results of classification rate per each lead with specific classifier based on MLP-SCG.

Table 1 Results for testing of MLP-SCG

Lead	MLP-SCG		
	50 hidden neurons	100 hidden neurons	150 hidden neurons
<i>i</i>	84.37	87.5	87.5
<i>ii</i>	81.25	76.56	79.68
<i>iii</i>	81.25	81.25	81.25
<i>avf</i>	76.56	75	81.25
<i>avr</i>	79.68	75	78.12
<i>avl</i>	70.31	73.43	76.56
<i>v1</i>	75	75	78.12
<i>v2</i>	70.31	70.31	65.62
<i>v3</i>	81.25	76.56	76.56
<i>v4</i>	75	81.25	82.81
<i>v5</i>	81.25	82.81	84.37
<i>v6</i>	79.68	78.12	82.81

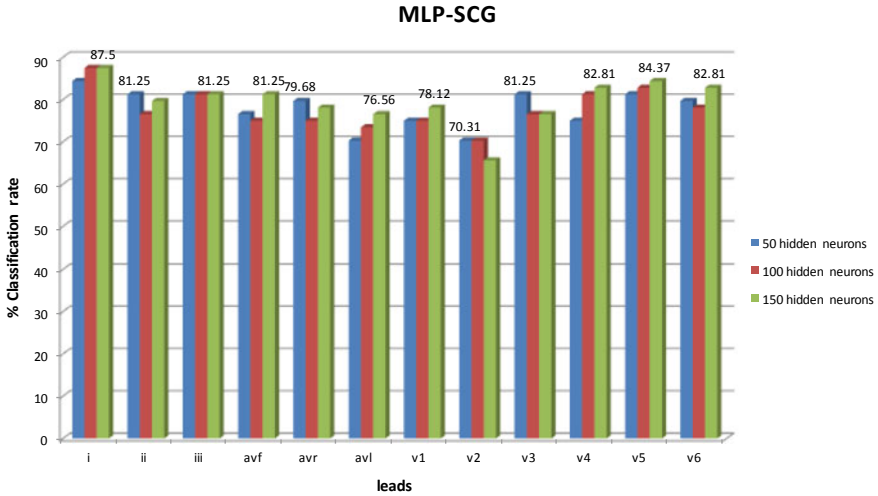


Fig. 7 Comparative results of different values of hidden neurons for MLP-SCG classifier for specific lead

Table 2 Results for global winner-takes-all method, integrating the outputs of the twelve expert modules in proposed modular neural network

Class	Myocardial infarction	Healthy controls
Myocardial infarction	37	3
Healthy controls	4	20

Classification Rate = 89.06%

Bold means the true positive prediction

4.2.2 Modular Neural Network and Winner-Takes-All Integration Method

Finally, we take the output results for the twelve selected expert modules to be integrate through a winner-takes-all method and obtained an 89.06% of global classification rate in modular neural network. In Table 2, we present more details in confusion matrix.

5 Conclusions

In the performed experiments, we found out the feature extraction with AR, Shannon entropy and wavelets of signals ECGs for the 12-lead captures the difference between the classes of the PTB database. We used classifiers based on MLP-SCG to form the expert modules in the modular neural network. We obtained better results used

winner-takes-all as integration method to combine the outputs of the expert module classifiers. But, in the future works, we need optimize the expert modules in order to improve the performance and classification rate of the proposed modular neural network. As well as, we will add the rest classes included in the PTB Diagnostic ECG database.

References

1. Martis, R.J., Achayra, U.R., Prasad, H., Chua, C.K.: Application of higher order statistics for atrial arrhythmia classification. *Biomed. Sign. Process. Control* **8**(6), 888–900 (2013)
2. Amezcua, J., Melin, P.: A modular LVQ neural network with fuzzy response integration for arrhythmia classification. In: *IEEE Conference on Norbert Wiener in the 21st Century* (2014)
3. Melin, P., Amezcua, J., Valdez, F., Castillo, O.: A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias. *Inf. Sci.* **279**, 483–497 (2014)
4. Khalaf, A.F., Owis, M.L., Yassine, I.A.: A novel technique for cardiac arrhythmia classification using spectral correlation and support vector machines. *Expert Syst. Appl.* **42**(21), 8361–8368 (2015)
5. Homaeinezhad, M.R., Atyabi, S.A., Tavakkoli, E., Toosi, H.N., Ghaffari, A., Ebrahimpour, R.: ECG arrhythmia recognition via a neuro-SVM-KNN hybrid classifier with virtual QRS image-based geometrical features. *Expert Syst. Appl.* **39**(2), 2047–2058 (2012)
6. Zhao, Q., Zhang, L.: ECG feature extraction and classification using wavelet transform and support vector machines. *IEEE Int. Conf. Neural Netw. Brain* **2**, 1089–1092 (2005)
7. Boussejot, R., Kreiseler, D., Schnabel, A.: Nutzung der EKG-Signaldatenbank CARDIODAT der PTB über das Internet. *Biomedizinische Technik, Band 40, Ergänzungsband 1 p. S 317* (1995)
8. Wang, J.S., Chiang, W.C., Hsu, Y.L., Yang, Y.T.: ECG arrhythmia classification using a probabilistic neural network with a feature reduction method. *Neurocomputing.* **116**, 38–45 (2013)
9. Javadi, M., Asghar, S.A., Sajedin, A., Ebrahimpour, R.: Classification of ECG arrhythmia by a modular neural network based on mixture of experts and negatively correlated learning. *Biomed. Sign. Process. Control.* **8**, 289–296 (2013)
10. Al Rahhal, M.M., Bazi, Y., Alhichri, H., Alajlan, N., Melgani, F., Yager, R.R.: Deep learning approach for active classification of electrocardiogram signals. *Inf. Sci.* **345**, 340–354 (2016)
11. Megat, M.S.A., Jahidin, A.H., Norall, A.N.: Hybrid multilayered perceptron network classification of bundle branch blocks. In: *IEEE 2012 International Conference Biomedical Engineering Icoobe* (2012). ISBN 978-1-4577-1991-2
12. Castillo, O., Melin, P., Ramirez, E., Soria, J.: Hybrid intelligent system for cardiac arrhythmia classification with fuzzy K-Nearest Neighbors and neural networks combined with a fuzzy system. *Expert Syst. Appl.* **39**, 2947–2955 (2012)
13. Melin, P., Ramirez, E., Prado-Arechiga, G.: Cardiac arrhythmia classification using computational intelligence: neural networks and fuzzy logic techniques. *European Heart J. OXFORD academic* **38**, P6388 (2017)
14. Osowski, S., Markiewicz, T., Hoal, L.T.: Recognition and classification systems of arrhythmia using ensemble of neural networks. *Measurement* **41**(6), 610–617 (2018)
15. Osowski, S., Siwek, K., Siroic, R.: Neural system for heartbeats recognition using genetically integrated ensemble of classifiers. *Comput. Biol. Med.* **41**(3), 173–180 (2011)
16. Jadhav, S.M., Nalbalwar, S.L., Ghatol, A.A.: ECG arrhythmia classification using modular neural network model. In: *IECBES (2012)* ISBN 978-1-4244-7600-8
17. Ozbay, Y., Tezel, G.: A new method for classification of ECG arrhythmias using neural network with adaptive activation function. *Digit. Sig. Proc.* **20**, 1040–1049 (2010)

18. Ozbay, Y., Ceylan, R., Karlik, B.: A fuzzy clustering neural network architecture for classification of ECG arrhythmias. *Comput. Biol. Med.* **36**, 376–388 (2005)
19. Gaetano, D., Panunzi, S., Rinaldi, F., Risi, A., Sciandrone, M.A.: A patient adaptable ECG beat classifier based on neural networks. *Appl. Math. Comput.* **213**(1), 243–249 (2009)
20. Melin, P., Castillo, O.: A review of the applications of type-2 fuzzy logic in classification and pattern recognition. *Expert Syst. Appl.* **40**(13), 5413–5423 (2013)
21. Melin, P., Castillo, O.: A review on the applications of type-2 fuzzy logic in classification and pattern recognition. *Expert Syst. Appl.* **40**, 5413–5423 (2013)
22. Ceylan, R., Ozbay, Y., Karlik, B.: A novel approach for classification of ECG arrhythmias: type-2 fuzzy clustering neural network. *ACM* **36**(3), 6721–6727 (2009)
23. Chua, T.W., Tan, W.W.: Interval type-2 fuzzy system for ECG arrhythmia classification. *Fuzzy Systems in Bioinformatics and Computational Biology*, pp. 297–314. Springer, Berlin (2009). ISBN 978-3-540-89968-6
24. Tan, W.W., FOO, C.L., Chua, T.: Type-2 fuzzy system for ECG arrhythmic classification, *FYZZ-IEEE* (2007). ISBN 1-4244-1209-9
25. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy K-Nearest neighbor algorithm. *IEEE Trans. Syst. Man Cybern.* **15**, 580–585 (1985)
26. Ramirez, E., Castillo, O., Soria, J.: Hybrid system for cardiac arrhythmia classification with Fuzzy K-Nearest Neighbors and neural networks combined by a fuzzy inference system. In: *Softcomputing for Recognition Based on Biometrics, Studies in Computational Intelligence*, vol. 312, pp. 37–53, Springer (2010). ISBN 978-3-642-15110-1
27. Ramirez, E., Melin, P., Prado-Arechiga, G.: Hybrid model based on neural networks, type-1 and type-2 fuzzy systems for 2-lead cardiac arrhythmia classification. *Expert Syst. Appl.* **126**, 295–307 (2019)
28. Wozniak, M., Grana, M., Corchado, E.: A survey of multiple classifier systems as hybrid systems. *Inf. Fusion* **16**, 3–17 (2014)
29. Melin, P., Prado-Arechiga, G., Miramontes, I., Medina, M.: A hybrid intelligent model based on modular neural network and fuzzy logic for hypertension risk diagnosis. *J. Hypertension* **34** (2016)
30. Nazmy, T.M., EL-Messiry, H., AL-Bokhity, B.: Classification of cardiac arrhythmia based on hybrid system. *Int. J. Comput. Appl.* **2** (2010)
31. Shao, Y.E., Hou, C.D., Chiu, C.C.: Hybrid intelligent modeling schemes for heart disease classification. *Appl. Soft Comput.* **14**, 47–52 (2014)
32. Luz, E.J.D.S., Schwartz, W.R., Camara-Chavez, G., Menotti, D.: ECG-based heartbeat classification for arrhythmia detection: a survey. *Comput. Methods Prog. Biomed.* **127**, 144–164 (2015)
33. Elhaj, F.A., Salim, N., Harris, A.R., Swee, T.T., Ahmed, T.: Arrhythmia recognition and classification using combined linear and nonlinear features of ECG signals. *Comput. Methods Prog. Biomed.* **127**, 52–63 (2016)
34. Jovic, A., Bogunovic, N.: Evaluating and comparing performance of feature combinations of heart rate variability measures for cardiac rhythm classification. *Biomed. Sign. Process. Control.* **7**, 245–255 (2012)
35. Martis, R.J., Achayra, U.R., Min, L.C.: ECG beat classification using PCA, LDA, ICA, and Discrete Wavelet Transform. *Biomed. Sign. Process. Control* **8**, 437–448 (2013)
36. Zopounidis, M., Doumpos, M.: Multicriteria classification and sorting methods: a literature review. *Eur. J. Oper. Res.* **138**, 229–246 (2002)
37. Gacek, A., Pedrycz, W.: *Ecg signal processing, Classification and Interpretation, a comprehensive framework of computational intelligence*, Springer (2012) ISBN 978-0-85729-867-6
38. Martindale, J.L., Brown, D.F.M.: *A visual guide to ECG interpretation, second edition*. Wolters Kluwer (2017) ISBN 978-1-4963-2153-4
39. Hampton, J.R., Adlam, D.: *The ECG in practice*, 6th edn. Churchill Livingstone, Elsevier (2013). ISBN 978-0-7020-4643-8
40. Jayasinghe, R.: *ECG Workbook*. Elsevier, Churchill Livingstone (2012). ISBN 978-0-7295-4109-1

41. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P.C.H., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.-K., Stanley, H.E.: physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals. *Circulation*. **101**(23), e215-e220 (2000)
42. Bishop, C.M.: *Neural network for pattern recognition*. Clarendon Press, Oxford, U.K
43. Leonarduzzi, R.F., Schlotthauer, G., Torres, M.E.: Wavelet leader based multifractal analysis of heart rate variability during myocardial ischaemia. In 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (2010)

Particle Swarm Optimization of Modular Neural Networks for Obtaining the Trend of Blood Pressure



Ivette Miramontes, Patricia Melin and German Prado-Arechiga

Abstract In this work, the optimization of a modular neural network for obtaining the trend of blood pressure is presented. Three modules are used, the first for obtaining the systolic pressure trend, the second one for the diastolic pressure trend and the last one for the heart rate trend. For each module of the neural network the layers and neurons are optimized, to find the architecture that will generate optimal results. In this case 47 readings of 50 patients are used to perform the optimization, these readings are obtained through a device called ambulatory blood pressure monitoring which is a non-invasive device that takes the patient's reading every 20 min on the day and every 30 min at night, in twenty-four hours. Once the architectures for each module are obtained by Particle Swarm Optimization, we select the architecture that presented the minimal error and is tested with a set of 25 patients who were not trained by the neural network, to test whether the training was performed properly.

Keywords Fuzzy systems · Optimization · Bio-inspired algorithm · Modular neural networks

1 Introduction

Neural networks have been used successfully in different areas such as pattern recognition [1], robotics [2], function approximation and prediction of time series [3], etc. In the medical area, these have been used for the diagnosis of different diseases, for example in Skin and breast cancer [4, 5], depression and anorexia [6], coronary heart disease [7], glaucoma [8], etc. For this work, a modular neural network is presented to obtain the trend of the blood pressure in a group of patients for a period of 24 h.

With the goal of obtaining a more accurate result of the trend given by the modular neural network, Particle Swarm Optimization (PSO) is used to find the best architecture that optimally solves the given problem. The PSO algorithm is well known for

I. Miramontes · P. Melin (✉) · G. Prado-Arechiga
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: pmelin@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_19

225

providing good results and has been used in different optimization problems [9–12], hence the interest to use this algorithm in this work.

This paper has been organized as follows: in Sect. 2 the literature review is presented, in Sect. 3 a methodology description is presented, in Sect. 4 the results and discussions are presented, and in Sect. 5 the conclusions obtained after carrying out the tests are presented.

2 Literature Review

This section presents the basic concepts necessary to understand the proposed method:

2.1 Particle Swarm Optimization

The Particle Swarm Optimization (PSO) algorithm, was proposed by Kennedy and Eberhart in 1995 [13], and is inspired by the social behavior of bird flocking or fish schooling. This algorithm has the advantages that it is easy to implement and only needs the adjustment of few parameters.

The PSO begins with a random population matrix, the rows in the matrix are called particles, each particle moves around the surface of the cost with a velocity. The algorithm will update the velocity vector for each of the particles, and then add that velocity to the values or position of the particle. These updates are influenced by the following:

1. The best global solution associated with the lowest cost ever found by a particle
2. The best local solution associated with the lowest cost in the current population.

Once this is taken into account, if the best local solution has a lower cost than the cost in the current global solution, then the global solution will be replaced by the best local solution. The PSO has the ability to deal with complex cost functions with many local minimums [14].

2.2 Blood Pressure and Hypertension

Blood pressure is very important for our body, because thanks to it the oxygen and nutrients are taken to our whole body, and this is defined as the force exerted by the blood being pushed against the walls of the arteries inasmuch as that the heart pumps blood [15].

The highest blood pressure is when the heart is between beats pumping blood, and it is called systolic pressure. The lowest blood pressure, which is when the heart is

relaxed between beats, is called diastolic pressure and this is measured in millimeters of mercury. Based on the European guide for the management of hypertension the normal blood pressure in a person is the one that goes below the 139/89 mmHg [16, 17].

Once the concept of blood pressure is understood, then we can define the hypertension as the maintained elevation of blood pressure above normal limits, which, according to the European guide for the management hypertension is above 140/90 mmHg. Hypertension has no clear symptoms, so it is important to have constant checkups, even more, if one of your parents has the disease [18, 19].

3 Methodology

The method is based on a modular neural network (MNN), which is part of a neuro-fuzzy model [20–23] to obtain a medical diagnosis based on the blood pressure of different patients. This modular neural network is the first part of the aforementioned model, which includes the readings of the blood pressure and the heart rate of a group of patients to be trained, that is, the information corresponding to the systolic pressure is the input of the first module, in the second the information of the diastolic pressure and in the third the readings corresponding to the heart rate. This information is trained in each of the modules of the neural network to learn the behavior that the blood pressure of different patients can have, obtaining as output the trend in a period of 24 h, and these measurements are taken by the ABPM [24].

Different parts of the model were optimized with different metaheuristics [25–27], for this case, the PSO was used to optimize the architecture of the neural network in each module, the parameters to be optimized are the number of hidden layers (which can be up to two layers hidden by the information that it handles), the number of neurons by hidden layers (which go from 1 to 30 neurons per hidden layer) and the number of epochs.

The PSO gives us the architecture that has generated the least error, it is for them that the mean square error is used to obtain that information.

$$MSE = \frac{1}{2} \sum_{i=1}^n (\hat{Y}_i - Y_i)^2 \quad (1)$$

4 Results and Discussion

To observe the behavior of the algorithm, 30 experiments were carried out, where in each case the PSO algorithm parameters were changed. The parameters are presented in Table 1 and correspond to the following:

Table 1 PSO parameters for each experiment

Parameters						
No.	popsize	maxit	npar	c1	c2	C
1	10	800	4	0.3	0.3	0.1
2	12	667	4	1.4	1.4	0.3
3	14	571	4	2	2	1.2
4	15	533	4	4.3	4.3	1.9
5	18	444	4	1.1	1.1	0.8
6	20	400	4	1.6	1.6	0.1
7	22	364	4	3.4	3.4	1.6
8	23	348	4	2.3	2.3	1.1
9	25	320	4	0.6	0.6	0.1
10	28	286	4	1.8	1.8	1
11	30	267	4	1	1	0.7
12	31	258	4	3.6	3.6	1.5
13	34	235	4	1.3	1.3	0.2
14	36	222	4	4	4	0.6
15	39	205	4	2.6	2.6	1.8
16	40	200	4	1.9	1.9	1.4
17	42	190	4	1	1	0.4
18	43	186	4	1.3	1.3	1
19	45	178	4	1.7	1.7	0.7
20	47	170	4	4.1	4.1	0.2
21	49	163	4	2.5	2.5	0.4
22	51	157	4	3.8	3.8	1.3
23	52	154	4	0.5	0.5	1.8
24	54	148	4	1.4	1.4	0.5
25	58	138	4	2.7	2.7	1
26	60	133	4	3.1	3.1	2
27	62	129	4	1.9	1.9	0.6
28	66	121	4	1.2	1.2	0.4
29	68	118	4	3.7	3.7	1.9
30	70	114	4	2.6	2.6	0.8

- *popsize*: represents the size of the swarm
- *maxit*: represents the maximum number of iterations in the experiment
- *npar*: represents the dimensions of the problem
- *c1*: represents the cognitive parameter
- *c2*: represents the social parameter
- *C*: represents constriction factor.

The first set of the experiments correspond to the systolic module, in Table 2 the 30 experiments are presented, the variation realized by the algorithm in layers, neurons in the hidden layers and epochs, as well as the best error obtained and the average of all the errors generated in that experiment.

The best-obtained experiment was number 11 with an average error of 0.56, marked in bold, where the best architecture was the following:

Table 2 Experiments of the systolic module

Systolic module							
No.	Layers	Neurons HL1	Neurons HL2	Epochs	Best error	AVG	Time
1	2	10	17	330	0.07	0.60	1:13:39
2	1	23	–	544	0.10	0.54	1:15:34
3	2	16	15	155	0.10	0.57	0:59:06
4	1	4	–	438	0.14	0.65	0:26:05
5	2	27	23	266	0.10	0.57	2:00:30
6	2	12	16	394	0.14	0.61	0:58:45
7	2	16	12	307	0.11	0.59	0:55:34
8	2	5	3	531	0.08	0.65	0:35:52
9	2	18	19	302	0.08	0.59	1:21:35
10	1	27	–	320	0.08	0.57	1:04:46
11	1	14	–	253	0.15	0.53	0:39:00
12	2	15	23	636	0.07	0.60	1:12:04
13	2	26	13	614	0.16	0.57	1:28:30
14	1	5	–	95	0.09	0.53	0:31:56
15	2	10	21	172	0.12	0.78	0:53:49
16	2	13	6	353	0.11	0.60	0:48:52
17	2	21	10	483	0.13	0.57	1:16:02
18	2	26	20	547	0.13	0.60	1:54:36
19	2	25	30	507	0.13	0.55	2:09:48
20	2	20	8	364	0.14	0.56	1:22:08
21	2	26	9	97	0.11	0.56	1:22:43
22	2	20	5	87	0.12	0.56	1:03:17
23	1	25	–	95	0.13	0.57	1:05:18
24	1	7	–	205	0.11	0.56	0:31:49
25	2	7	13	136	0.15	0.67	0:36:59
26	1	4	–	92	0.14	0.62	0:29:52
27	2	11	26	149	0.17	0.64	1:01:05
28	1	19	–	426	0.10	0.56	0:57:39
29	2	11	10	467	0.17	0.61	0:48:18
30	2	21	24	437	0.10	0.56	1:33:05

- Hidden Layers: 1
- Neurons: 19
- Epochs: 426.

While the worst error was generated by the experiment 4, marked in italic.

The second set of experiments correspond to the diastolic module, in Table 3 the 30 experiments are presented, the variation realized by the algorithm in hidden

Table 3 Experiments of the diastolic module

Diastolic module							
No.	Layers	Neurons HL1	Neurons HL2	Epochs	Best error	AVG	Time
1	2	18	10	113	0	0.003	1:06:02
2	1	28	–	80	0	0.003	1:25:42
3	1	19	–	454	0	0.004	0:45:04
4	2	22	28	272	0	0.003	1:56:56
5	1	28	–	99	0	0.003	1:44:30
6	2	<i>1</i>	<i>8</i>	<i>235</i>	<i>0</i>	<i>0.006</i>	<i>0:33:40</i>
7	1	30	–	314	0	0.003	1:28:28
8	1	23	–	401	0	0.003	1:13:59
9	2	19	7	450	0	0.003	1:20:37
10	1	24	–	608	0	0.003	1:10:10
11	2	19	18	370	0	0.003	1:32:32
12	2	21	28	622	0	0.003	1:32:32
13	1	24	–	646	0	0.003	1:06:24
14	1	25	–	318	0	0.003	1:11:17
15	1	22	–	528	0	0.003	0:53:43
16	1	15	–	538	0	0.005	1:07:36
17	1	30	–	524	0	0.003	1:30:37
18	1	28	–	138	0	0.003	1:36:36
19	2	6	16	549	0	0.003	0:49:05
20	1	5	–	377	0	0.003	0:33:35
21	1	14	–	518	0	0.003	0:39:35
22	1	27	–	456	0	0.003	1:11:02
23	2	23	29	271	0	0.003	2:12:06
24	2	12	17	556	0	0.003	1:02:43
25	1	27	–	682	0	0.003	1:24:11
26	2	28	7	64	0	0.003	1:47:41
27	1	28	–	344	0	0.003	1:23:26
28	1	22	–	579	0	0.003	1:02:25
29	1	27	–	183	0	0.003	1:12:19
30	2	28	26	405	0	0.003	2:40:52

layers, neurons in the hidden layers and epochs, as well as the best error obtained and the average of all errors generated in that experiment.

The best-obtained experiment was number 20 with an average error of 0.003, marked in bold, where the best architecture was the following

- Hidden Layers: 1
- Neurons:5
- Epochs: 377.

While the worst error was generated by experiment 6, marked in italic.

The last set of experiments correspond to the heart rate module, in Table 4 the 30 experiments are presented, the variation realized by the algorithm in the hidden layers, neurons in the hidden layers and epochs, as well as the best error obtained and the average of all the errors generated in that experiment.

The best-obtained experiment was number 18 with an average error of 0.010, marked in bold, where the best architecture was the following

- Hidden Layers: 1
- Neurons:27
- Epochs: 140.

While the worst error was generated by the experiment 25, marked in italic.

Training of the modular neural network with the best-obtained architectures is carried out, to test the performance of the modular neural network, 36 test patients are consulted, where the results obtained by each module are compared with the real trend of the ABPM. In Table 5, the experiments performed are presented with 15 patients. It can be observed that all the trends presented are within the margin of error allowed, this means, the variation is minimal compared with the information provided by the ABPM.

In Table 6, the results obtained by the non-optimized modular neural network are presented, it can be observed that there is more variation compared to the information provided by the ABPM.

In Table 7 the success percentages for the 36 patients are presented, as can be observed, the percentage of success of the trend of the three modules is improved for the PSO, being the results of systolic pressure module the one that presents more improvement.

5 Conclusions and Future Work

In this work, the optimization of a modular neural network using the PSO algorithm is presented. The main contribution of this work was to use the PSO algorithm to find the appropriate architecture and with which better results will be provided for each module of the modular neural network to provide a more accurate diagnosis of the complete computational model. Once the experimentation was done and the best

Table 4 Experiments of the heart rate module

Heart rate module							
No.	Layers	Neurons HL1	Neurons HL2	Epochs	Best error	AVG	Time
1	1	9	–	278	0	0.013	0:51:02
2	1	4	–	26	0	0.018	0:28:18
3	2	29	2	675	0	0.011	2:11:44
4	2	7	2	62	0	0.017	1:08:25
5	1	12	–	176	0	0.015	0:45:15
6	2	28	4	506	0	0.012	2:34:25
7	2	2	11	494	0	0.018	0:33:33
8	1	15	–	331	0	0.013	1:16:22
9	1	13	–	447	0	0.013	0:52:25
10	2	14	25	447	0	0.013	1:38:42
11	1	8	–	337	0	0.015	0:45:23
12	2	24	29	393	0	0.012	3:22:05
13	1	26	–	612	0	0.011	1:53:02
14	1	25	–	269	0	0.011	1:48:59
15	2	17	9	600	0	0.012	1:20:55
16	2	8	20	584	0	0.015	1:06:37
17	1	9	–	655	0	0.014	0:45:18
18	1	27	–	140	0	0.010	2:28:36
19	1	16	–	120	0	0.012	1:20:34
20	2	3	24	188	0	0.017	0:48:28
21	2	10	21	482	0	0.014	1:16:25
22	2	3	2	386	0	0.018	0:36:17
23	1	21	–	211	0	0.017	1:26:43
24	2	19	21	673	0	0.012	2:01:54
25	2	2	22	199	0	0.028	0:46:40
26	1	8	–	152	0	0.015	0:49:49
27	1	11	–	492	0	0.014	0:51:24
28	1	1	–	390	0	0.021	0:37:42
29	1	27	–	426	0	0.012	1:48:00
30	2	11	28	457	0	0.014	1:21:25

architectures were found, a significant improvement could be observed in the results provided by the systolic pressure module, while for the remaining two modules the improvement obtained with the architectures provided with the PSO was minimal, so we can conclude that the algorithm only provided significant help for the first module of the modular neural network.

Table 5 Experiments with the optimized MNN in a group of 15 patients

No.	Systolic			Diastolic			Heart rate		
	Real	MNNO	% Success	Real	MNNO	% Success	Real	MNNO	% Success
1	106	101	95	62	59	95	95	94	99
2	107	102	95	61	64	95	69	70	99
3	134	126	94	62	71	87	72	72	100
4	121	115	95	77	75	97	80	80	100
5	106	97	92	65	65	100	73	73	100
6	110	112	98	69	69	100	63	64	99
7	130	130	100	86	82	95	74	74	100
8	117	125	94	73	79	92	71	71	100
9	117	116	99	54	56	96	70	70	100
10	113	114	99	72	72	100	67	67	100
11	121	121	100	78	77	99	77	77	100
12	123	129	95	82	87	94	71	71	100
13	102	105	97	62	65	95	74	74	100
14	121	117	97	70	70	100	70	70	100
15	117	118	99	68	64	94	62	62	100

Table 6 Experiments with the non-optimized MNN in a group of 15 patients

No.	Systolic			Diastolic			Heart rate		
	Real	MNNO	% Success	Real	MNNO	% Success	Real	MNNO	% Success
1	106	115	92	62	60	97	95	92	97
2	107	137	78	61	63	97	69	69	100
3	134	130	97	62	64	97	72	71	99
4	121	114	94	77	68	88	80	80	100
5	106	82	77	65	70	93	73	73	100
6	110	106	96	69	70	99	63	63	100
7	130	122	94	86	80	93	74	74	100
8	117	136	86	73	75	97	71	70	99
9	117	111	95	54	58	93	70	69	99
10	113	115	98	72	79	91	67	66	99
11	121	108	89	78	72	92	77	77	100
12	123	113	92	82	85	96	71	70	99
13	102	30	29	62	63	98	74	74	100
14	121	130	93	70	77	91	70	69	99
15	117	118	99	68	73	93	62	62	100

Table 7 Percentage of success in the non-optimized and optimized MNN

Non optimized			Optimized		
Systolic	Diastolic	Pulse	Systolic	Diastolic	Pulse
90%	95%	99%	95%	96%	100%

As future work, we intend to perform the optimization with other bio-inspired algorithms to observe its performance and if there is a significant improvement in the results provided by the diastolic pressure module and heart rate module. In addition, we can consider other applications following the works in [28–31].

Acknowledgements The authors would like to express thank to the Consejo Nacional de Ciencia y Tecnologia and Tecnológico Nacional de Mexico/Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

- Melin, P.: *Modular Neural Networks and Type-2 Fuzzy Systems for Pattern Recognition*. Springer, Berlin Heidelberg (2012)
- Jin, L., Li, S., Yu, J., He, J.: Robot manipulator control using neural networks: a survey. *Neurocomputing* **285**, 23–34 (2018)
- Melin, P., Pulido, M.: Optimization of ensemble neural networks with type-2 fuzzy integration of responses for the dow jones time series prediction. *Intell. Autom. Soft Comput.* **20**(3), 403–418 (2014)
- Esteve, A.: Dermatologist-level classification of skin cancer with deep neural networks. *Nature* **542**, 115 (2017)
- Álvarez Menéndez, L., de Cos Juez, F.J., Sánchez Lasheras, F., Álvarez Riesgo, J.A.: Artificial neural networks applied to cancer detection in a breast screening programme. *Math. Comput. Model.* **52**(7–8), 983–991 (2010)
- Wang, Y.T., Huang, H.H., Chen, H.H.: A neural network approach to early risk detection of depression and anorexia on social media text. *CEUR Workshop Proc.* **2125** (2018)
- Atkov, O.Y., et al.: Coronary heart disease diagnosis by artificial neural networks including genetic polymorphisms and clinical parameters. *J. Cardiol.* **59**(2), 190–194 (2012)
- Chen, X., Xu, Y., Wong, D.W.K., Wong, T.Y., Liu, J.: Glaucoma detection based on deep convolutional neural network. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 715–718 (2015)
- Li, Z.J., Duan, X.D., Shuo, J.: Facial expression recognition based on PSO optimization. *Appl. Mech. Mater.* **411–414**, 1151–1154 (2013)
- Menke, C.: Application of particle swarm optimization to the automatic design of optical systems. In: *Proc.SPIE, Optical Design and Engineering VII*, vol. 10690 (2018)
- Tarique, A., Gabbar, H.A.: Particle swarm optimization (PSO) based turbine control. *Intell. Control Autom.* **04**(02), 126–137 (2013)
- Liang, X., Li, W., Zhang, Y., Zhou, M.: An adaptive particle swarm optimization method based on clustering. *Soft. Comput.* **19**(2), 431–448 (2015)
- Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of ICNN'95—International Conference on Neural Networks*, vol. 4, pp. 1942–1948 (1995)
- Haupt, R.L., Haupt, S.E.: *Practical Genetic Algorithms*, 2nd edn. A Wiley-Interscience publication, New Jersey (2004)

15. O'Brien, E., Parati, G., Stergiou, G.: Ambulatory blood pressure measurement. *Hypertension* **62**(6), 988–994 (2013)
16. Zanchetti, A., et al.: 2018 ESC/ESH guidelines for the management of arterial hypertension. *Eur. Heart J.* **39**(33), 3021–3104 (2018)
17. Carretero, O.A., Oparil, S.: Essential hypertension. *Circulation* **101**(3), 329–335 (2000)
18. Beevers, G., Lip, G.Y.H., O'Brien, E.: *ABC of Hypertension*, 5th edn. Blackwell Publishing, Malden, MA (2007)
19. Battegay, E.J., Lip, G.Y.H., Bakris, G.L.: *Hypertension: Principles and Practices*. CRC Press, Boca Raton, FL (2005)
20. Melin, P., Miramontes, I., Prado-Arechiga, G.: A hybrid model based on modular neural networks and fuzzy systems for classification of blood pressure and hypertension risk diagnosis. *Expert Syst. Appl.* **107**, 146–164 (2018)
21. Miramontes, I., Martínez, G., Melin, P., Prado-Arechiga, G.: A hybrid intelligent system model for hypertension risk diagnosis. In *Fuzzy Logic in Intelligent System Design*, pp. 202–213 (2018)
22. Guzmán, J.C., Melin, P., Prado-Arechiga, G.: Neuro-fuzzy hybrid model for the diagnosis of blood pressure. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) *Nature-Inspired Design of Hybrid Intelligent Systems*, pp. 573–582. Springer International Publishing, Cham (2017)
23. Pulido, M., Melin, P., Prado-Arechiga, G.: Blood pressure classification using the method of the modular neural networks. *Int. J. Hypertens.* **2019**, 1–13 (2019)
24. Grossman, E.: Ambulatory blood pressure monitoring in the diagnosis and management of hypertension. *Diabetes Care* **36**(2), 307–311 (2013)
25. Guzmán, C.J., Miramontes, I., Melin, P., Prado-Arechiga, G.: Optimal genetic design of type-1 and interval type-2 fuzzy systems for blood pressure level classification. *Axioms* **8**(1), 8 (2019)
26. Miramontes, I., Guzman, C.J., Melin, P., Prado-Arechiga, G.: Optimal design of interval type-2 fuzzy heart rate level classification systems using the bird swarm algorithm. *Algorithms* **11**(12), 206 (2018)
27. Guzman, J.C., Melin, P., Prado-Arechiga, G.: Design of an optimized fuzzy classifier for the diagnosis of blood pressure with a new computational method for expert rule optimization. *Algorithms* **10**(3), 79 (2017)
28. Mendez, G.M., Castillo, O.: Interval type-2 TSK fuzzy logic systems using hybrid learning algorithm. In: *The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05*, pp. 230–235
29. Melin, P., Gonzalez, C.I., Castro, J.R., Mendoza, O., Castillo, O.: Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Trans. Fuzzy Syst.* **22**(6), 1515–1525 (2014)
30. Melin, P., Castillo, O.: Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach. *IEEE Trans. Ind. Electron.* **48**(5), 951–955 (2001)
31. Castro, J.R., Castillo, O., Melin, P., Rodríguez Díaz, A.: Building fuzzy inference systems with a new interval type-2 fuzzy logic toolbox. *Trans. Comput. Sci.* **1**, 104–114 (2008)

Classification of X-Ray Images for Pneumonia Detection Using Texture Features and Neural Networks



Sergio Varela-Santos and Patricia Melin

Abstract Based on the reports of the Center Of Disease Control each year around 50,000 people die because of Pneumonia in the United States, this disease affects the area of the lungs and can be detected (diagnosed) by analyzing chest X-rays. Because of this it's important the development of computational intelligent techniques for the diagnosis and classification of lung diseases, and as a medical tool for the quick diagnosis of diseases, for this work we used a segment of the ChestXRay14 database which contains radiographic images of several lung diseases including pneumonia, we extracted the area of interest from the pneumonia images using segmentation techniques and furthermore we applied a process of feature extraction on the area of interest of the images to obtain Haralick's Texture Features and perform classification of the disease using a neural network with good results on the classification of pneumonia X-ray images from healthy X-ray images.

Keywords Neural networks · Image classification · Texture features · GLCM · X-ray · Pneumonia

1 Introduction

Research on computational intelligent techniques for the detection of diseases has been for many years the main backbone on the development of Computer Aided Diagnosis Systems (CAD) [1–4], this are a combination of hardware and software components that offer information to support the diagnosis done by a medical professional by analyzing medical images or medical data obtained from specialized medical devices, on regard of visual data some of these types of digital medical

S. Varela-Santos · P. Melin (✉)
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: pmelin@tectijuana.mx

S. Varela-Santos
e-mail: sergio.varela@tectijuana.edu.mx

© Springer Nature Switzerland AG 2020
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_20

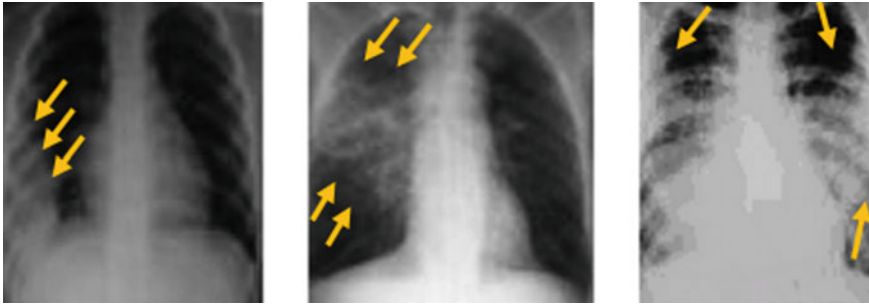


Fig. 1 Pneumonia presented in frontal or PA (posterior anterior) chest X-ray images

images can be Magnetic Resonance Images (MRI), Computer Tomography(CT), X-rays and Ultrasounds [5, 6], this are performed on patients as a medical test to find possible abnormalities that potentially indicate a disease on the patient.

Computational research applied to medical image diagnosis has been for many years a priority for the scientific community, there has been advances in different areas, from Breast lesion [3] or Microcalcification detection [7], Lung diseases [1], and even Heart disease diagnosis [8–10] using computational methods on digital medical images of the patient’s internal structures and organs.

Based on reports by the Center of Disease Control each year around 50,000 people die in the US because of pneumonia [11], also The World Health Organization reports that Pneumonia was the cause of death of 18,458 people in Mexico in 2015 [12] and was responsible for the decease of 920,136 children in the world in 2015 and is still the general cause of 15% of all children’s deaths in the world [13]. This disease develops when an infection causes the air sacs of the lungs to fill up with fluids or pus, this can be caused by a virus or bacteria [14]. The infection can be usually acquired by breathing the air that contains these bacteria. In Fig. 1 we can observe that in a chest X-ray it can be seen as a white opacity in the air space (black space) inside the lung [15, 16].

Following this idea, we present our first attempt to develop a method for classifying pneumonia in frontal chest X-ray images by analyzing features related to the gray levels spatial distribution or “Texture” in the X-ray image to classify it as pneumonia or a healthy chest X-ray.

2 Overview of Chest X-ray Disease Detection Methods

The development of methods for the diagnosis of diseases obtained in chest X-rays has been a subject of research for several years, but until recently databases of X-rays have been either privately owned or inexistent, in 2017 the National Institute of Health released the largest Chest X-ray Database to date on the paper by Wang [17], the public release for scientific research purposes of new databases makes it possible for

the development of new and more sophisticated models, and puts us a step further to diagnosing diseases with almost perfect precision using computationally intelligent techniques.

In Wang et al. [17] a new database called ChestXRay8 and later ChestXRay14 is presented for the detection of diseases in a frontal chest X-ray, this database contains 112,120 frontal chest X-ray images divided in 14 non-proportional classes that correspond to 13 different diseases and a “No Findings” class which corresponds to a picture with a healthy chest X-ray, this are non-proportional because the classes on the database are imbalanced and have different number of samples per class in which each class corresponds to a particular disease. It was reported in this paper for the best convolutional neural network called ResNet-50, a classification accuracy of 63.3% for the Pneumonia class, using all 14 classes of the database. Different approaches have used this database to detect one or all of this diseases, to work with the 14 classes at the same time it is more common to use a convolutional neural network or deep learning technique due to the big quantity of images to train the network with.

In Li et al. [18] two deep learning models called DenseNet-121, DenseNet-RNN were used to diagnose the diseases in ChestXRay14 obtaining a total of 74.5% and 75.1% respectively detecting Pneumonia.

In Rajpurkar et al. [19] a model called CheXNet was introduced using 121 layers to detect all 14 diseases detecting at a 76.8% of accuracy the Pneumonia class from the others, also this model provides a heatmap for the possible localization of the disease based on the prediction done by the convolutional neural network.

In Yao et al. on their model is reported an accuracy of 71.3% on classifying Pneumonia from the other classes. In these examples they sacrifice accuracy in an attempt to classify correctly all of the diseases of the database using a single model, which in a clinical sense can be dangerous when using on real patients an accuracy below the likes of 90%.

Another type of models based on smaller more specialized classifiers on a specific disease tackle this problem on one class or disease at a time comparing with a healthy person's X-ray image.

In Antin et al. [20] the k-means algorithm was used to predict the Pneumonia class taken from the ChestXRay14 database after filtering the images using an anti-aliasing filter.

In Khobragade et al. [21] a method was proposed using preprocessing techniques like intensity based methods using thresholding and discontinuity based methods like the Canny Filter followed by a geometrical feature extraction process and the training of a feed forward neural network as the classifier of the data, finally the reported results was of 92% of classification accuracy.

In R. Pavithra et al. [22] a feed forward neural network was used to identify two classes from a privately owned dataset of 116 images as either Pneumonia or Lung Cancer with 94.8% of accuracy.

3 Basic Concepts and Theory

3.1 *Chest X-ray*

It's a medical test that shows information on the structures inside the chest. An X-ray is a type of high-energy radiation that can go through the body and onto film, making pictures of areas inside the chest, which can be used to diagnose disease. Chest X-rays are common type of exam [Les. R. Folio Chest Imaging]. A chest X-ray is often among the first procedures you'll undergo if your doctor suspects you have heart or lung disease. Also an X-ray is cheaper than other procedures to detect the same diseases like magnetic resonances for example.

3.2 *Artificial Neural Networks*

An Artificial Neural Network is a computational structure inspired by the study of the processing of the biological neurons in the brain. Neurons have the capacity to stimulate the human brain and learn a pattern based on partial or incomplete information to furthermore predict new data of the same type.

This neural networks have connections between the neurons that communicate the information after its processed from one neuron to another, obtaining a piece of transformed data on every neuron to finally obtain the output or the final piece of data that corresponds to the prediction done by the network based on the input value or values.

One of the most popular topologies for artificial neural networks is the multilayer perceptron using the backpropagation algorithm as the training method [34]. A multilayer perceptron is a network topology that corresponds to more than one hidden layer in the neural network.

A common structure of a neural network can be observed in Fig. 2 and is based on an input layer, 1 or more hidden layers, and an output layer which each layer contains 1 or more neurons that apply a simple operation on the data based on the type of transfer function used on the neuron and pass it on the next one, until the output layer obtains the final data which represents the prediction done by the neural network. This neural networks have a training algorithm which is the backpropagation which works by reducing the network error or the difference between the desired output and the actual output calculated by the network in the actual state, following this calculation the network adjusts its weights to get closer to the desired output, this is done several number of times in a cycle called epoch, finally we reach a stopping condition based on the error goal or error precision desired. This training algorithms have been changed over the years adding parameters like momentum and learning rate in order to minimize the error in the network and avoid local minima [34].

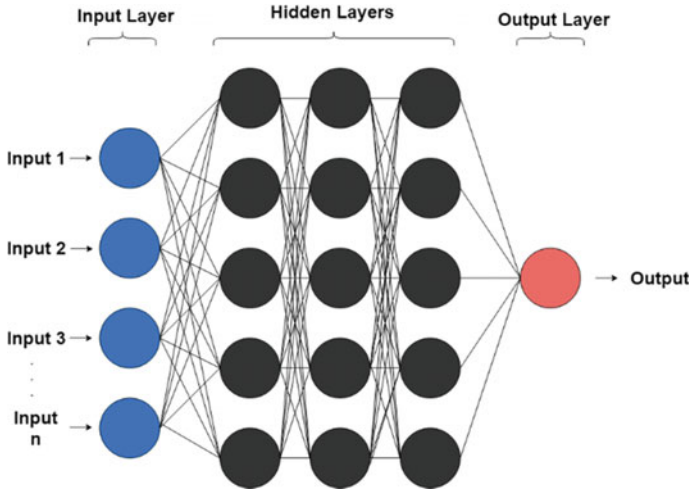


Fig. 2 Example of a neural network architecture

3.3 Feature Extraction

Computationally is easier to work with a representation of an image that maintains its primary information than with the whole image, this is one of the reasons to use a feature extraction process, in the same way it serves as a tool to remove irrelevant information and preserve meaningful information from the image [23]. These features help to obtain possible classification rules that a classifier can take into account to perform with high accuracy. In the case of X-ray images one of the best feature extraction techniques are Texture and Geometrical Features [23], this excel in this regard because of the way the diseases appear on the X-rays, as white clouds or opacities inside the air space also most of the diseases possess a certain geometrical form that can be recognized visually by a trained medical expert.

3.4 Gray Level Co-occurrence Matrix

This matrix stores the number of times a pixel of certain intensity of gray level occurs with another pixel in the image matrix in a given offset which can be horizontal at 0° , vertical at 90° or diagonally at 45° , 135° .

This serves as a feasible way to obtain the spatial relationships between each pixel type in the image, this information gives us a hint at the inner patterns of a gray scale image. Figure 3 explains the construction of a Gray Level Co-Occurrence Matrix or GLCM, first we establish an offset, this is the configuration followed to create the matrix, selecting the orientation and number of pixels for each comparison, in the

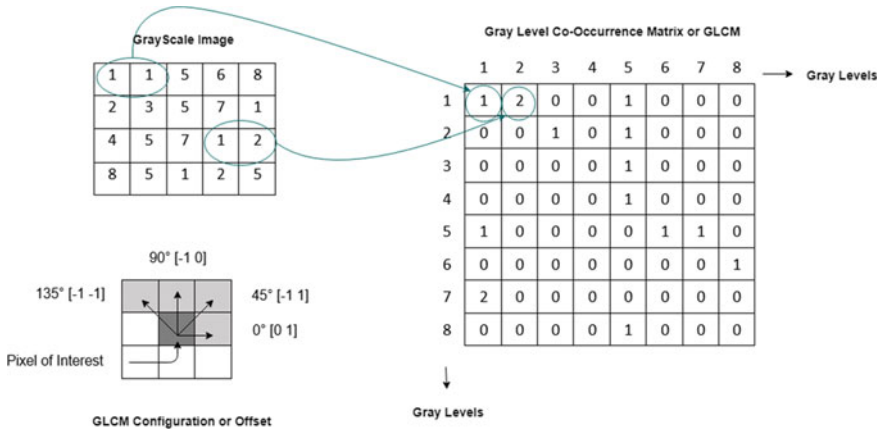


Fig. 3 Example of the construction of the GLCM

figure the example is done using a horizontal (0°) orientation and 1 pixel distance for comparison to the pixel of interest also called [0 1].

3.5 Features Based on Image Texture

This features are based on the spatial distribution of the grey scale levels on segments of the image (neighborhoods). In other words it can be defined as a quantification of the spatial variation of the grey scale level values. This are obtained from calculations done over the Gray Level Co-Occurrence Matrix or GLCM and are unique for each image and GLCM offset.

In texture classification the problem is identifying the given textured set from a given set of texture classes [24]. Each region in a digital image has unique characteristics, the texture analysis algorithms extract distinguishing features from each region to facilitate classification of such patterns by describing a class of digital images based on the values of such features, thus being able to differentiate between classes in pattern recognition and classification problems.

Properties such as gray level co-occurrence, contrast, entropy and homogeneity are computed from image gray levels to facilitate classification. Since texture is a spatial property, a simple one-dimensional histogram is not useful in characterizing texture, for example in the figure we can observe a histogram of 50% white pixels and 50% black pixels but with 3 different texture patterns (block pattern, checkerboard and a striped pattern) [24] this is an easy way to explain the importance of texture analysis as a way of image class recognition. To identify correctly this images, is not enough to count the pixels, but to analyze the inner patterns of the image using texture feature extraction techniques (Fig. 4).

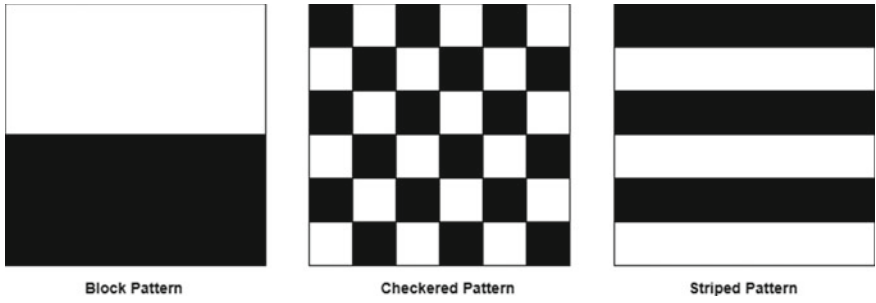
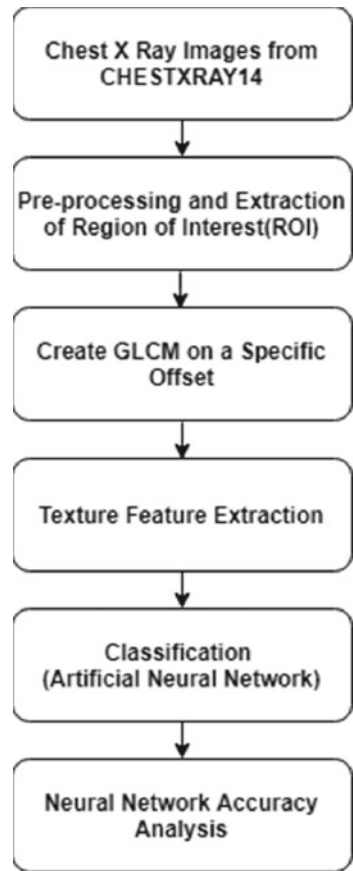


Fig. 4 Figures with same number of white and black pixels but different image texture

Fig. 5 Proposed method to classify pneumonia in chest X-ray images



Texture feature extraction techniques were introduced in 1973 by Haralick [25], this features represent one of the first ways to try to explain patterns in digital images, frequently used in aerial images and microscopic images [26], have also found an application in important areas like disease detection in digital medical images [5, 27], this features are based on second order statistics and are applied on the GLCM to obtain values specific to the image being evaluated and specific to the offset used to create the GLCM.

4 Working with Chest X-ray Images

Regarding some important processes to prepare the data and find the necessary patterns to detect the disease is important to follow certain steps to clean the data and to focus the feature extraction process in the region of interest of the images.

These steps can include a general preprocessing step like Image Enhancement using Histogram Equalization, followed by segmentation of the image to obtain an area of interest which can be the Lung Air Space or the Full Chest Area [23] and finally applying a Feature Extraction Process to obtain Texture or Geometrical Features which can give us the information needed to detect the disease by classifying the X-rays in different classes.

5 Proposed Method

Our approach corresponds to finding features that can give the neural network patterns or classification rules to discriminate between a healthy chest X-ray and a chest X-ray with pneumonia.

The features obtained analyze the distribution of pixels of different tonalities of grey levels, this is a good approach because this features have been used in the past [30] as a reliable method for obtaining relevant information on grey-scale images and more importantly on medical images where the diseases are most of the time visualized as neighborhoods of white pixels over black tissue, and where is not easy to predict the exact place where the disease is forming, because of this reasons we proposed a method to identify the disease using features based on the analysis of texture from images presenting the same disease, to train the network to learn the values that correspond to an image that has the disease. In Fig. 3 an example of the construction of the GLCM.

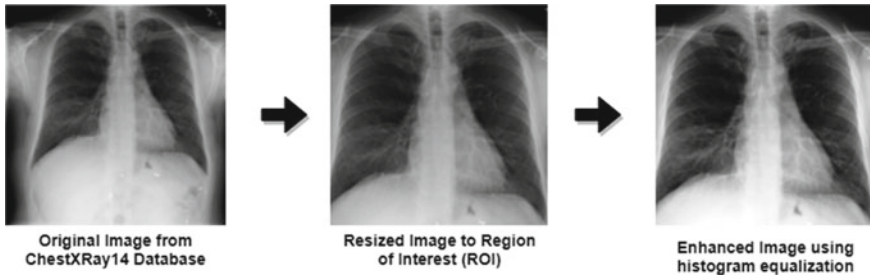


Fig. 6 Process to extract region of interest (ROI) from the chest X-rays

6 Experimental Results

6.1 Preprocessing a Region of Interest for Chest X-rays

To focus the feature extraction process in the affected area, is necessary to obtain a region of interest (ROI) from the image, which is a rudimentary process of noise removal or irrelevant information elimination, to segment a region of interest, we need to know the possible places in which the disease can be found in the image.

For pneumonia, this disease appears as a white cloudy figure inside the area of the lungs, specifically inside the black space also known as air space.

For our approach to obtain the ROI we reduced the images to just the chest area, eliminating almost all of the noise proportionally on all the sample for both classes, following that procedure we use histogram equalization to enhance the images and fix the white and black contrast of the images to see all the tissues, organs and possible abnormalities more clearly, this is done to be able to work with just the relevant information on the problem and filter as much noise as possible. In Fig. 6 we can visually observe the ROI extraction process.

6.2 Gray Co-occurrence Matrix Offset

This matrix as said before can be calculated following a specific configuration, following some simple experiments done over the Gray Co-Occurrence Matrix (GLCM), for our problem a good configuration to use is $[0 \ 2]$, which is the same as 0 degrees and comparison to 2 pixels, so in our case the GLCM is constructed comparing 2 pixels horizontally to the pixel of interest.

We show an example in Fig. 5 of the configuration for the GLCM used for our experimentation (Fig. 7).

Is necessary to mention that a different configuration can be used to perform the same experimentation to obtain different classification results, our offset selection is empirical, and we leave this to future investigation.

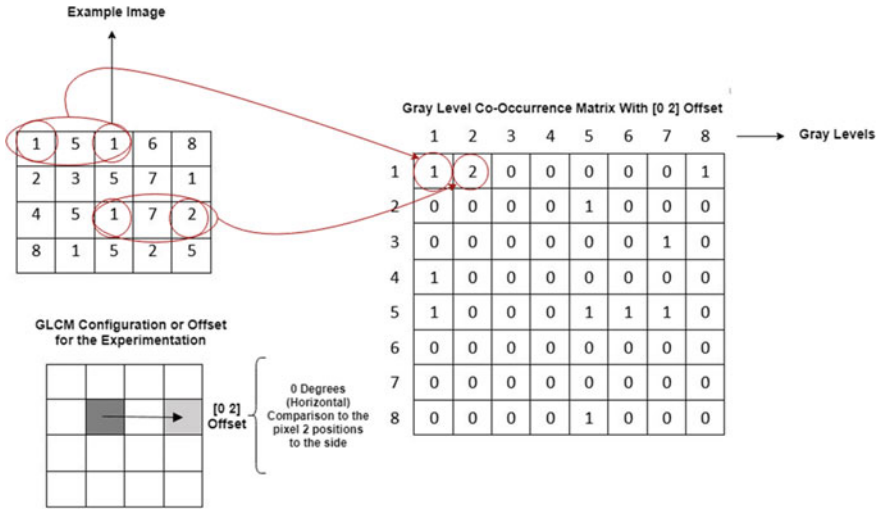


Fig. 7 Offset for the gray level co-occurrence matrix used in the experimentation

6.3 Texture Feature Extraction

After generating the GLCM we can now obtain the features, using the statistical measures explain the image texture, this features represent behaviors of the pixels inside each grayscale image based on the spatial comparisons of each pixel occurrence with one another. In Table 1 we can observe the statistical measures.

6.4 Neural Network Architecture

In order to identify the disease, we need to train a neural network with the values obtained from the feature extraction process to let the network learn the patterns for each class from the feature values, the architecture used for the Neural Network can be observed in Fig. 8.

From a 200 image sample taken from the ChestXRy14 database for the Pneumonia and No Findings class, we obtained using the 19 texture feature extraction techniques 3800 feature values, which represent 1900 feature values per class, for the experimentation it was used 70% of the total feature values proportionally for each class for training, 15% for testing and 15% for validation of the model. In Table 2. We present the parameters used for the Monolithic Neural Network in experimentation with the model (Fig. 9).

Table 1 Comparison of ranges of feature values per class

Features	Abnormal or pneumonia class (Gray level 8)	Normal or no findings class (Gray level 8)
Autocorrelation	38.40414521 ± 14.3279113	24.15126358 ± 13.15889489
Cluster prominence	559.8908926 ± 14.09851438	472.9135127 ± 129.4414881
Cluster shade	13.99245072 ± -56.35808736	21.77201949 ± -6.169078446
Contrast	0.10626073 ± 0.028320815	0.089438446 ± 0.058789461
Correlation	0.994474602 ± 0.942459589	0.99116347 ± 0.982718725
Difference entropy	0.302594191 ± 0.125859643	0.301245369 ± 0.223623608
Difference variance	0.099406891 ± 0.027518747	0.081439852 ± 0.05533326
Dissimilarity	0.09001073 ± 0.027614449	0.089434859 ± 0.058789461
Angular second moment (ASM) or homogeneity or energy	0.338684981 ± 0.121343897	0.20003576 ± 0.118173099
Entropy	2.345102182 ± 1.349190393	2.340466532 ± 1.905439352
Inverse different moment or local homogeneity	0.9863528 ± 0.954995351	0.970605269 ± 0.955282929
Information measures of correlation I	-0.780715218 ± -0.91043362	-0.806051156 ± -0.857759233
Information measures of correlation II	0.981453793 ± 0.912761701	0.982982893 ± 0.96754455
Inverse difference	0.986370764 ± 0.954995827	0.970605269 ± 0.955283168
Maximum probability	0.525827969 ± 0.160312947	0.33936224 ± 0.146239389
Sum average	11.982348 ± 7.160715308	9.062043887 ± 6.558319676
Sum entropy	2.282815058 ± 1.323152331	2.28296997 ± 1.861784155
Sum of squares variance	4.413009599 ± 0.613551502	4.041328722 ± 2.06499368
Sum variance	17.58273046 ± 2.383597995	16.08280667 ± 2.06499368

6.5 Classification Results

Our results were satisfactory following the Texture Features approach in which we accomplished an average classification accuracy of 93% using the neural network with the Gradient Descent with Adaptive Learning and Momentum Algorithm and 90% using the Scaled Conjugate Gradient Algorithm, both experiments were done using the same Neural Network architecture, as for the best training obtained from all the experiments it was a 95% classification accuracy using the Gradient Descent with Adaptive Learning and Momentum Algorithm as can be seen in Fig. 10. In Tables 3 and 4 we present the experimental results done using this approach.

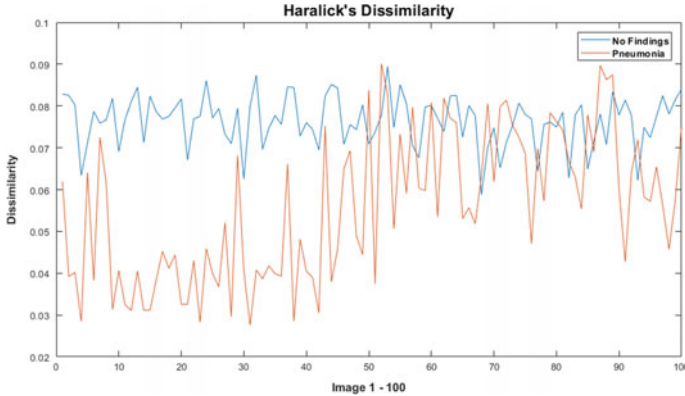


Fig. 8 Example of the distribution of values for all the sample for both classes on one of the texture features

Table 2 Monolithic neural network parameters

Network architecture	Architecture 1	Architecture 2
Inputs	19	19
Hidden layers	3	3
Neurons	19 Input layer /10, 20, 10 hidden layers/1 output layer	19 Input layer/10, 20, 10 hidden layers/1 output layer
Outputs	1	1
Learning rate	–	0.1
Momentum constant	–	0.9
Training algorithm	Scaled conjugate gradient	Gradient descent with adaptive learning and momentum
Performance evaluation measure	Mean squared error	Mean square error

7 Conclusions and Future Work

Texture Features represent a good way to identify diseases in medical images, it serves as a first step in finding the best features that can help identifying pneumonia in X-ray images. To further improve the results of the Neural Network, we can try using more feature extraction methods and obtain features like Geometrical Features to have more possible classification rules and give the Neural Network a broader criteria for discrimination of classes.

In order to optimize the classifier we can use a feature selection process like a Genetic Algorithm to try different combination of features as inputs and obtain the Neural Network with the best accuracy and least features.

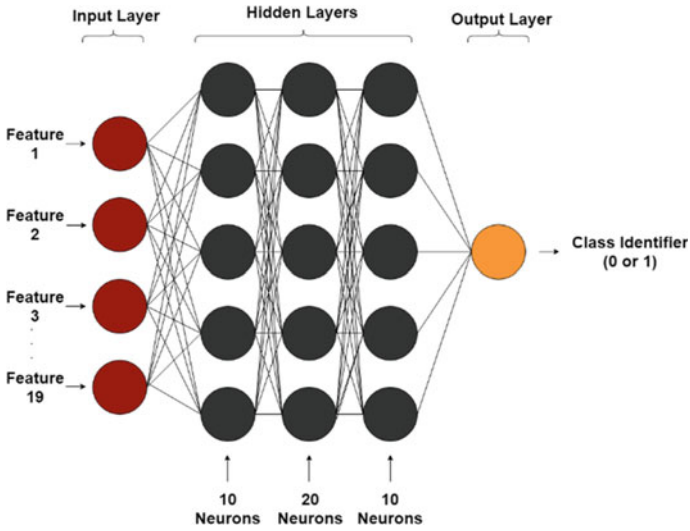


Fig. 9 Neural network architecture used for experimentation

Fig. 10 Confusion matrix for the best neural network obtained using texture features



Table 3 30 experiments using scaled conjugate gradient

No.	Training (%)	Test (%)	Validation (%)	Final accuracy (%)	MSE (Mean Square Error)
1	93.60	90.00	90.00	92.50	0.0632
2	85.00	90.00	93.30	87.00	0.1053
3	88.60	86.70	86.70	88.00	0.0888
4	90.70	83.30	93.30	90.00	0.0788
5	87.90	86.70	83.30	87.00	0.0978
6	90.70	96.70	96.70	92.50	0.0703
7	90.70	86.70	93.30	90.50	0.0774
8	87.90	90.00	90.00	88.50	0.0897
9	90.70	93.30	86.70	90.50	0.0733
10	91.40	83.30	86.70	89.50	0.0853
11	88.60	86.70	86.70	88.00	0.0773
12	88.60	86.70	93.30	89.00	0.0842
13	93.60	83.30	93.30	92.00	0.0575
14	90.70	86.70	90.00	90.00	0.0824
15	89.30	86.70	96.70	90.00	0.0799
16	90.70	86.70	90.00	90.00	0.0801
17	91.40	86.70	96.70	91.50	0.0704
18	91.40	93.30	93.30	92.00	0.0625
19	92.10	93.30	90.00	92.00	0.0615
20	95.00	83.30	93.30	93.00	0.0477
21	92.90	93.30	93.30	93.00	0.0588
22	90.70	90.00	86.70	90.00	0.0737
23	85.50	90.00	83.30	85.50	0.0898
24	95.50	86.70	90.00	93.00	0.0408
25	94.30	80.00	96.70	92.50	0.0389
26	87.90	90.00	93.30	89.00	0.0901
27	91.40	93.30	96.70	92.50	0.0706
28	88.60	96.70	86.70	89.50	0.0946
29	89.30	90.00	93.30	90.00	0.0738
30	92.10	83.30	83.30	89.50	0.0658
Best	95.50	96.70	96.70	93.00	0.0389
Mean	90.70	86.70	91.65	90.00	0.07555

Table 4 30 experiments using gradient descent with adaptive learning and momentum

No.	Training (%)	Test (%)	Validation (%)	Final Accuracy (%)	MSE (Mean Square Error)
1	92.10	83.30	90.00	90.50	0.0759
2	95.70	93.30	83.30	93.50	0.0447
3	85.70	83.30	83.30	85.00	0.0931
4	80.00	86.70	76.70	80.50	0.17
5	93.60	90.00	93.30	93.00	0.0529
6	97.10	86.70	90.00	94.50	0.0214
7	86.40	93.30	96.70	89.00	0.0919
8	93.60	86.70	96.70	93.00	0.0553
9	95.00	93.30	86.70	93.50	0.0393
10	90.00	86.70	86.70	89.00	0.0723
11	91.40	90.00	86.70	90.50	0.0731
12	90.70	83.30	93.30	90.00	0.0388
13	94.30	93.30	83.30	92.50	0.0424
14	90.70	83.30	93.30	90.00	0.0745
15	92.90	90.00	86.70	91.50	0.0584
16	90.70	90.00	90.00	90.50	0.0732
17	95.70	83.30	90.00	93.00	0.0374
18	93.60	86.70	100	93.50	0.0542
19	91.40	90.00	90.00	91.00	0.0671
20	90.00	76.70	90.00	88.00	0.0895
21	92.10	93.30	93.30	92.50	0.067
22	92.90	96.70	93.30	93.50	0.0594
23	92.90	90.00	90.00	92.00	0.0665
24	96.40	93.30	93.30	95.50	0.0342
25	92.90	90.00	90.00	92.00	0.0602
26	92.90	93.30	93.30	93.00	0.052
27	93.60	83.30	93.30	92.00	0.0687
28	91.40	93.30	93.30	92.00	0.0712
29	91.40	86.70	90.00	90.50	0.0695
30	94.30	90.00	93.30	93.50	0.0637
Best	97.10	96.70	100.00	95.50	0.0214
Mean	92.90	90.00	90.00	92.00	0.0637

As future work we can envision using type-2 fuzzy logic to improve results as in [28–31] and using granularity as in [32–34].

Acknowledgements we would like to express our gratitude to CONACYT, Tijuana Institute of Technology for the facilities and resources granted for the development of this research.

References

1. Horváth, G., et al.: A CAD system for screening X-ray Chest radiography. In: Dössel O., Schlegel, W.C. (eds.) World Congress on Medical Physics and Biomedical Engineering, IFMBE Proceedings, Munich, Germany, vol. 25(5). Springer, Berlin, Heidelberg, 7–12 Sept 2009
2. Doi, K.: Computer-aided diagnosis in medical imaging: historical review, current status and future potential. *Comput. Med. Imaging Graph.* **31**(4–5), 198–211 (2007)
3. Baker, J.A.: Computer-aided detection (CAD) in screening mammography: sensitivity of commercial CAD systems for detecting architectural distortion. *Am. J. Roentgenol* **181**(4), 1083–1088 (2003)
4. Cheng, J.-Z.: Computer-aided diagnosis with deep learning architecture: applications to breast lesions in US images and pulmonary nodules in CT scans. *Sci. Rep.* **6**, 1–13 (2016)
5. Miranda, E., Aryuni, M., Irwansyah, E.: A survey of medical image classification techniques. In: 2016 International Conference on Information Management and Technology (ICIMTech), pp. 56–61, IEEE, Bandung, Indonesia (2016)
6. Van Ginneken, B., Ter Haar Romeny, B.M., Viergever, M.A.: Computer-aided diagnosis in chest radiography: a survey. *IEEE Trans. Med. Imaging* **20**(12), 1228–1241 (2001)
7. Rubio, Y., Montiel, O., Sepúlveda, R.: Microcalcification detection in mammograms based on fuzzy logic and cellular automata. In: Melin, P., Castillo, O., Kacprzyk, J. (eds.) Nature-Inspired Design of Hybrid Intelligent Systems. Studies in Computational Intelligence, vol. 667. Springer, Cham (2017)
8. González, B., Valdez, F., Melin, P., Prado-Arechiga, G.: Echocardiogram image recognition using neural networks. In: Castillo, O., Melin, P., Pedrycz, W., Kacprzyk, J. (eds.) Recent Advances on Hybrid Approaches for Designing Intelligent Systems. Studies in Computational Intelligence, vol. 547. Springer, Cham (2014)
9. González, B., Melin, P., Valdez, F., Prado-Arechiga, G.: Ensemble neural network optimization using a gravitational search algorithm with interval type-1 and type-2 fuzzy parameter adaptation in pattern recognition applications. In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications. Studies in Computational Intelligence, vol. 749. Springer, Cham (2018)
10. González, B., Valdez, F., Melin, P., Prado-Arechiga, G.: A gravitational search algorithm for optimization of modular neural networks in pattern recognition. In: Castillo, O., Melin, P. (eds.) Fuzzy Logic Augmentation of Nature-Inspired Optimization Metaheuristics. Studies in Computational Intelligence, vol. 574. Springer, Cham (2015)
11. Pneumonia – National Center for Health Statistics – Center for Disease Control and Prevention, <https://www.cdc.gov/nchs/fastats/pneumonia.htm>, last accessed 2018/09/03
12. Pérez-Padilla, J.R.: Muertes respiratorias en México 2015. *Neumol Cir Torax* **77**(3), 198–202 (2018)
13. Pneumonia—World Health Organization. <https://www.who.int/es/news-room/fact-sheets/detail/pneumonia>. Last accessed 05 November 2018
14. Pneumonia—Radiological Society of North America. <https://www.radiologyinfo.org/en/info.cfm?pg=pneumonia>. Last accessed 07 October 2019
15. Duong, M., Timoney, P., MacNicholas, R., Kitchen, J., Mustapha, M., Arunasalam, U.: ABC’s of chest X-rays”. *Trinity Stud. Med. J.* **2** (2011)

16. Pneumonia—University Of Virginia—Pneumonia Pathology. <https://www.med-ed.virginia.edu/courses/rad/cxr/pathology3chest.html>. Last accessed 07 October 2019
17. Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., Summers, R.M.: Chest X-ray 8: hospital-scale chest X-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2017)
18. Li, J., Xu, Z., Zhang, Y.: Diagnosing chest x-ray diseases with deep learning
19. Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C., Shpanskaya, K., Lungren, M., Ng, A.Y.: CheXNet: radiologist-level pneumonia detection on chest X-rays with deep learning (2017)
20. Antin, B., Kravitz, J., Martayan, E.: Detecting pneumonia in chest X-rays with supervised learning
21. Khobragade, S., Tiwari, A., Patil, C.Y., Narke, V.: Automatic detection of major lung diseases using chest radiographs and classification by feed-forward artificial neural network. In: 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES), IEEE, Delhi, India (2016)
22. Pavithra, R., Pattar, S.Y.: Detection and classification of lung disease—pneumonia and lung cancer in chest radiology using artificial neural networks. *Int. J. Sci. Res. Publ.* **5**(10), 1–5 (2015)
23. Bankman, I.N.: *Handbook of Medical Image Processing and Analysis*, 2nd edn. Academic Press, San Diego, CA, USA (2008)
24. Haralick, R.M., Shapiro, L.G.: *Computer and Robot Vision*. Vol. 1, 1st edn. Addison-Wesley Publishing Company, Inc, USA (1992)
25. Haralick, R.M.: Textural features for image classification. *IEEE Trans. Syst. Man Cybern.* **SMC-3**(6), 610–621 (1973)
26. Tuceryan, M., Jain, A.K.: *Texture analysis*. *Handbook of Pattern Recognition and Computer Vision*. 1st edn. World Scientific Publishing, Singapore (1993)
27. Karthikeyan, S.: Performance analysis of gray level co-occurrence matrix texture features for glaucoma diagnosis. *Am. J. Appl. Sci.* **11**(2), 248–257 (2014)
28. Melin, P., González, C.I., Castro, J.R., Mendoza, O., Castillo, O.: Edge-detection method for image processing based on generalized type-2 fuzzy logic. *IEEE Trans. Fuzzy Syst.* **22**(6), 1515–1525 (2014)
29. González, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
30. González, C.I., Melin, P., Castro, J.R., Mendoza, O., Castillo, O.: An improved Sobel edge detection method based on generalized type-2 fuzzy logic. *Soft. Comput.* **20**(2), 773–784 (2016)
31. Ontiveros, E., Melin, P., Castillo, O.: High order α -planes integration: a new approach to computational cost reduction of general type-2 fuzzy systems. *Eng. Appl. AI* **74**, 186–197 (2018)
32. Melin, P., Sánchez, D., Castillo, O.: Genetic optimization of modular neural networks with fuzzy response integration for human recognition. *Inf. Sci.* **197**, 1–19 (2012)
33. Melin, P., Sánchez, D.: Multi-objective optimization for modular granular neural networks applied to pattern recognition. *Inf. Sci.* **460–461**, 594–610 (2018)
34. Sánchez, D., Melin, P., Castillo, O.: Optimization of modular granular neural networks using a firefly algorithm for human recognition. *Eng. Appl. of AI* **64**, 172–186 (2017)

Segmentation and Classification of Noisy Thermographic Images as an Aid for Identifying Risk Levels of Breast Cancer



Pilar Gomez-Gil, Daniela Reynoso-Armenta, Jorge Castro-Ramos, Juan Manuel Ramirez-Cortes and Vicente Alarcon-Aquino

Abstract During 2016, breast cancer was the second cause of death among women within ages of 24–30. This makes mandatory to find reliable strategies that support physicians and medical services for early diagnosis of such disease. Currently, mammography is considered the gold instrument for assessing the risk of having breast cancer, but other methods are being analyzed, looking for systems that may be cheaper and easier to apply, including the use of thermographic images. In this paper, we present an analysis of the performance of a system based on a “Feed-Forward Neural Network” (FFNN), for the identification of two and three levels of risk cancer. Indeed, a system based on a “Regions-Convolutional Neural network” (R-CNN) for automatic segmentation of the breast is proposed. Both systems were tested in a private database developed by the “Center for Studies and Cancer Prevention, A.C.” located in Oaxaca, Mexico, which presents important challenges as class unbalances, a slack recording with respect to application of the protocol and noise. The systems were evaluated using three subsets of the database, built using images with different levels of challenges. Our results showed that a FNN classifier performed well only with data strictly following the protocol, while the levels of performance with noisy data are not yet acceptable for real applications. In the other hand, the results obtained by the automatic segmentation based on R-CNN were competitive, encouraging for more research in this area.

P. Gomez-Gil (✉) · D. Reynoso-Armenta · J. Castro-Ramos · J. M. Ramirez-Cortes
National Institute of Astrophysics, Optics and Electronics, Tonantzintla, Puebla, Mexico
e-mail: pgomez@inaoep.mx

D. Reynoso-Armenta
e-mail: danimichelle.ra@gmail.com

J. Castro-Ramos
e-mail: jcastro@inaoep.mx

J. M. Ramirez-Cortes
e-mail: jmram@inaoep.mx

V. Alarcon-Aquino
University of Americas Puebla, San Andrés Cholula, Puebla, Mexico
e-mail: vicente.alarcon@udlap.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_21

Keywords Deep convolutional neural networks (CNN) · Breast cancer · Regions-CNN · Classification · Image processing · Thermography

1 Introduction

According to the United States Food and Drug Administration (FDA), a mammogram is currently the best screening tool for the diagnosis of breast cancer [1], being able to obtain sensitivity values from 75 to 90%. In spite of that, other methods are currently being investigated or already used as support tools to detect breast cancer. Among them, the use of thermographic images of the chest has received a lot of attention and is currently being applied as a first test in some countries, in particular in regions with a limitation in their financial resources. Thermograms are useful for screening because the human body produces heat, which is visualized in the infra-red spectra; tumors are highly fed by blood vessels, which generate a heater zone around them with respect to the vicinity zones and such differences may be observed in thermographic images. It is important to highlight that important government health regulators, as the FDA, have made very clear that a thermogram is not a substitute for a mammogram [2]. However, thermographic screening has been approved by the FDA as a complementary method to be used with other testing screens, as a mammography. In spite of that, it is important to analyze the suitability of this technique as an aid for preliminary diagnosis of breast cancer, because the equipment required for this methodology is less costly than other devices, making this technique very convenient as an initial test suitable for poor areas, where the budgets for screening equipment are very limited. Another advantage of thermography is that it may be used in patients younger than 40 years old, which normally are banned to mammography.

The aim of our ongoing research is to build a reliable system able to support the decision of a physician for the first diagnosis of different levels of risk of breast cancer, based on the use of thermograms. It is also our goal to produce a system able to work well with images captured using noise images generated with relaxed protocols, considering that in rural areas the medical staff has limited knowledge of imaging techniques or no infrastructure for supporting hard protocols. As a first step, in this paper we present an analysis of the performance of a segmentation algorithm and a classifier for the identification of two and three levels of risk of breast cancer. The analysis was carried out using a private database, called TH-CEPREC, which was built and labeled by the staff of the “Center for Studies and Cancer Prevention” (CEPREC by its acronym in Spanish: “Centro de Estudios para Prevención de Cáncer, A.C.”) located in Juchitán, Oaxaca, Mexico [3]. The database contains thermographic images of women, whose risk of cancer was labeled by a physician into 5 different risks of malignity, as described in Sect. 3.1. TH-CEPREC presents an important unbalance in the number of samples of each level of malignity. To preliminary solve this inconvenience we manually re-labeled the data into broader two and three levels of risks, in order to have more samples for each class. Indeed, the TH-CERPED contains images highly noisy, showing different calibrations of the

camera and presenting a low signal-to-noise rate. For that reason, we split the level of the problem complexity by analyzing three different subsets of data, as described in Sect. 3.2. In this work we also present the results obtained by a proposed system able to automatic segment the Region of Interest (ROI), that is, the breasts in the images, which was carried out using an “Regions-Convolutional Neural deep neural network” (R-CNN) [4].

The paper is organized as follows: Sect. 2 presents some published works dealing with domains similar to the one presented here. Section 3 details the design of the main system, including the characteristics of the database, the feature extraction process and the neural networks used for segmentation and classification. Section 4 explains the experimental setup and results applying the three different combination of features. Section 5 presents some conclusions and our suggestions for the next steps of this research.

2 Related Work

The number of publications related to breast cancer identification in thermographic images is fewer than the ones found using other kind of images, probably due to the complexity of this task and low reliability of thermogram for diagnosis. However, according to [5], such number has increased from 11 in 2000 to more than 55 in 2018. Other important problem faced in this area is a lack of public benchmark databases with a representative amount of samples. Notice that in the work of [6], some FFNNs were implemented for the classification of normal and abnormal thermograms, trained and tested using only 19 images, which makes difficult to generalize such results. Recent works have used larger databases, but such data is not available for free. An example of this is the work of [7], where 65 images taken at different resolutions but without following a specific protocol, were used for classifying normal and abnormal images. Nevertheless, encouraging results have been published lately. For example, in the same work of [7] a featured extraction method was proposed, combining Markov Random Fields (MRF) and a modification of Local Binary Patterns (LBP). Features were obtained of both breasts of the patient and a fusion was implement using a Hidden Markov Model (HMM). They obtained a false negative rate of 8.3% and a false positive rate of 5% using a testing set of 32 images.

Some other works proposed rather simple approaches. For example, in [8] thermographic images were divided in normal and abnormal according the change in the average of the temperature in the breast regions (ROI). Sixty patients with possible damaged tissue in the mama were analyzed calculating the media, variance, bias and kurtosis over the histogram of the image. Two classes of tissue were identified: hypo echoic and cystic; authors reported ranges of performance of 89–91% for the first class and 92–30% for the second. For the localization of the ROI, (tissue) they reported a range of 53–61%. The work of [9] presented a preliminary work for identification of breast cancer, based on the segmentation of heat zones in the image. They

applied a k-means algorithm for clustering pixels into three regions, using color. In such article an automatic classification component is not included; instead, the different segments of the clustered image are presented to an expert to be identified. Images were taken using a FLIRE30 Infrared Camera, with a resolution of 160×160 , stored in a JPEG format; a protocol similar to the one proposed in CEPRAC was followed for the acquisition of images.

3 Materials and Methods

The design of the segmentation system and the classifier was carried out following the workflow depicted in Fig. 1. This includes the following components: acquisition,

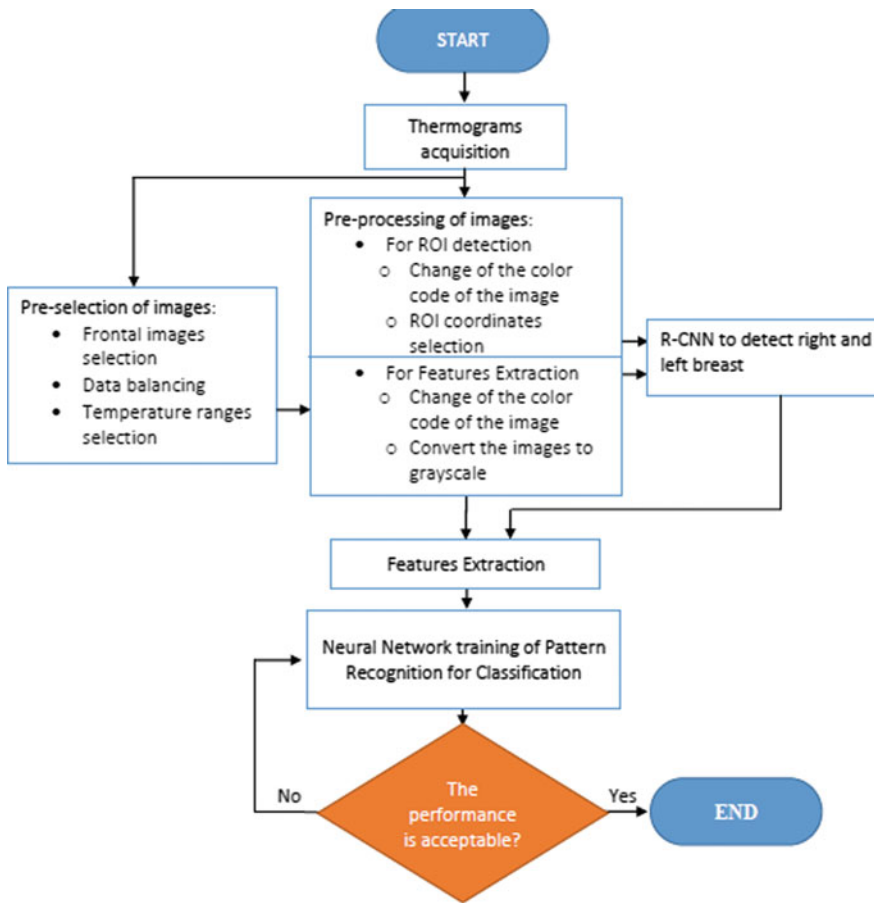


Fig. 1 Design of the thermographic images classifier using FFNN [10]

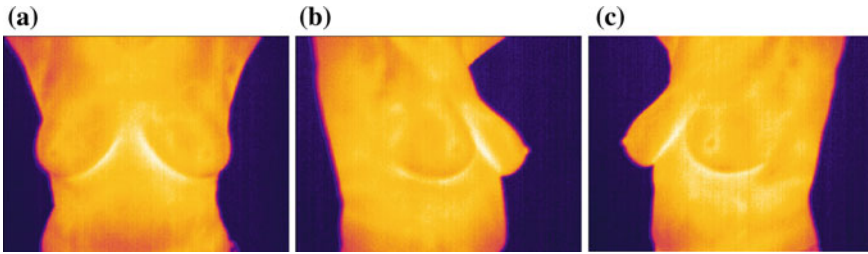


Fig. 2 An example of high quality images in TH-CEPREC taken by **a** front, **b** right lateral, **c** left lateral

selection and pre-processing of the images, the design of an automatic segmentation procedure using a R-CNN, the selection of features suitable for this problem and the design of a neural-net-based classifier. Next, we present some details of each component.

3.1 Acquisition of the Images

TH-CEPREC contains thermographic images and clinical information of around 600 women treated by CEPREC staff during 2008–2009. Images were taken using a particular protocol developed by the center, following some international recommendations, similar to the ones proposed by [2]. Images were captured using a camera DL-700C, with a resolution of 320×240 pixels and a spectral range of 8 to $14 \mu\text{m}$. The rest of patient's information mainly related to physical registry and physical tests were stored in Excel database files.

The CEPREC protocol recommends to take images in a closed room kept to a temperature of 18°C , capturing only the thorax part of the human body and shooting three images: by front, by right lateral and by left lateral parts; more details about the protocol for taking the images are described in [3]. Figure 2 presents an example of a set of 3 images taken following the protocol. However, such instructions were not followed during all shooting sessions. Other important lack of protocol's compliance was that a standard camera calibration was not performed, producing images stored in several palettes, whose colors represent different temperatures. Indeed, some patient's expedients contain less than 3 images (Fig. 1).

3.2 Selection and Pre-processing of Images

In the labeling developed by CEPREC, malignity is classified using a code called “thermos biological score” [8, 11, 12]. It is defined as follows: TH1 indicates a normal condition with no blood vessels; TH2 represents a normal tissue with blood

vessels; TH3 corresponds to an image with some abnormalities, but that may not be classified as abnormal; TH4 correspond to abnormal images and TH5 represents “very abnormal” images. This implies that TH1 represents no risk while TH5 corresponds to images with malign tumors. In the database TH-CEPREC, TH5 label was supported with a tissue biopsy.

TH-CEPREC presents a high imbalance with respect to the number of elements belonging to each malignity, which is appreciated in Table 1. As a preliminary way to overcome this problem, we defined three new classes, identified in this article as “levels of risk:” “healthy”, which includes images belonging to TH1 and TH2; “hyper-vascularization,” including malignity TH3 and “high-risk,” which includes malignity TH4 and TH5 (Fig. 3).

Using these data, 3 subsets of TH-CEPREC were created. The first one, called *Frontal-123*, contains the best 41 images of each level of risk, with respect to their resolutions. The second subset was created selecting images complying with some rationality with respect to temperature. Consider that the protocol suggested 18 °C. as room temperature and in the other hand, a temperature greater than 41 °C was not expected to be present in the body zones of the images; however, values outside of such ranges were found in some images. Therefore, we calculate the mean (μ) and standard deviation (σ) of the maximum and minimum temperatures for all the 123 selected images. Such values $\mu \pm \sigma$ were 22.5 ± 8.22 °C for the minimum temperature and 31.0 ± 5.62 °C for the maximum temperature. It resulted that 84 images contained minimum and maximum temperatures inside such ranges, then we selected them to form another subset identified as *CorrectRanges-84*. The last dataset, identified as *Representative-6*, contains 2 images of each level of risk, taken from subset *CorrectRanges-84*, that show the best quality from the point of view of the requirements specified by an expert physician. Only two were selected because no more than that high quality images for class TH5 were available.

As we stated before, images were taking using different pallets, therefore the code of color of the images was changed. This change also helped to improve the contrasts in the information being analyzed in the image. For automatic segmentations, images were converted to a palette that assigns a code known as rainbow [13], where pixels corresponding to the lowest temperature are assigned to navy blue and highest temperatures correspond to white color. Before the extraction of features, images were first converted to rainbow code, and then converted to a gray-scale using the function `rgb2gray` provided by Matlab [14].

Table 1 Number of patients in TH-CEPREC by level of malignity and risk

Malignity level	Number of patients	Level of risk
TH5	13	High-risk
TH4	38	
TH3	285	Hyper-vascularization
TH2	4	Healthy
TH1	235	

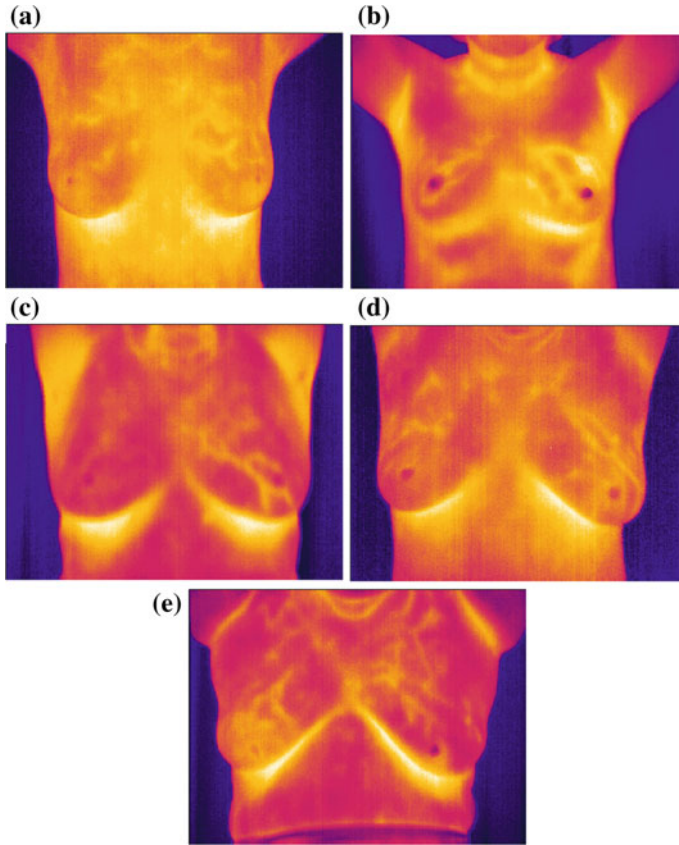


Fig. 3 Examples of images by malignity **a** TH1, **b** TH2, **c** TH3, **d** TH4, **e** TH5 (taken from TH-CEPREC, 2008–2009)

3.3 Automatic Segmentation

For the experiments reported here, ROI is defined as a rectangle into the image containing a breast. Each image may contain one or two ROIs, depending upon if a mastectomy was performed in the patient or not. Segmentation of the ROI was done both manually and automatically, as described next. Manual segmentation is required for labeling the database used for training the automatic segmentation software. Indeed, it was used to validate the classifier in a separate way from automatic segmentation. Selection of the ROI areas was performed visually, obtaining the coordinates directly from the color images.

Automatic segmentation was implemented using transfer learning and a deep learning network known as: R-CNN (Regions with convolutional neural networks) proposed by [4]. Transfer learning in neural networks consists on training a system using a big amount of data and then modifying their weights using a small amount

of data that belongs to other task, with the aim of learning such new task. The advantage of transfer learning is that basic features may be trained by a deep neural network using a context where lots of data are available. There are several models for transfer learning, in this research we used an application provided by Matlab R2017 titled “Object Detection Using Deep Learning” [15], which is based in a CNN (Convolutional Neural Network) trained using the public database CIFAR-10 for classification. CIFAR-10 database contains 50,000 images of 10 classes. Figure 4. shows a snapshot of the architecture used by the Matlab application to define the CNN network. After training, the CNN is converted to an R-CNN, which aims to identify a ROI, processing only regions of the image that are most probable to include the ROI. This network is re-training now using as inputs the coordinates of the ROIs of our thermographic images. After performing the segmentation, the application returns the coordinates of the most possible region with a breast, as well as a confidence index. Figure shows 2 examples of automatic segmentation. In order to obtain both breast of the same image, the procedure was executed twice, covering the first found ROI during the second execution (Fig. 5).

```

layers =
  15x1 Layer array with layers:
    1  **  Image Input           32x32x3 images with 'zerocenter' normalization
    2  **  Convolution          32 5x5 convolutions with stride [1 1] and padding [2 2 2 2]
    3  **  ReLU                  ReLU
    4  **  Max Pooling          3x3 max pooling with stride [2 2] and padding [0 0 0 0]
    5  **  Convolution          32 5x5 convolutions with stride [1 1] and padding [2 2 2 2]
    6  **  ReLU                  ReLU
    7  **  Max Pooling          3x3 max pooling with stride [2 2] and padding [0 0 0 0]
    8  **  Convolution          64 5x5 convolutions with stride [1 1] and padding [2 2 2 2]
    9  **  ReLU                  ReLU
   10  ** Max Pooling          3x3 max pooling with stride [2 2] and padding [0 0 0 0]
   11  ** Fully Connected      64 fully connected layer
   12  ** ReLU                  ReLU
   13  ** Fully Connected      10 fully connected layer
   14  ** Softmax               softmax
   15  ** Classification Output crossentropyex
  
```

Fig. 4 CNN architecture used by [15] for a first step in object segmentation

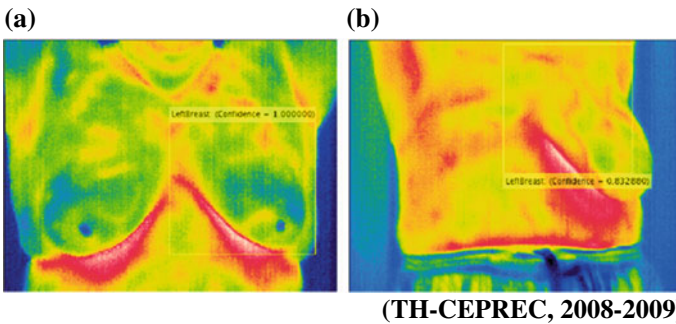


Fig. 5 Two examples of automatic segmentation of a breast in a thermographic image of (TH-CEPREC)

3.4 Selection of Features

Features were selected according to the characteristics that physicians observe in the thermograms when looking for tumors. The most relevant of them include the analysis of the bilateral symmetry between both breasts and the homogeneity in the color of the images. For that reasons, we decide to use features based on texture, histograms and Fourier transform calculated over a gray level image to capture such information, which are:

- (1) Correlation of the Local Binary pattern (LBP) obtained from both breasts measurement, producing 10 values.
- (2) Symmetry among different zones of both breast images (10 values).
- (3) Correlation of the histograms of both breasts (1 value).
- (4) Differences between intensities of both breast (1 values).
- (5) Standard deviation between intensities for both breasts (2 values).
- (6) Correlation of the Fourier transform of both breasts (1 values).
- (7) Correlation between quadrant 1 and 2 in both breast.

Details about calculating these values are given in [10]. In order to find the features that best suit for this problem, we analyzed 3 combinations of these 7 features:

Case 1: includes the 7 features.

Case 2: includes the first 5 features.

Case 3: includes the first 3 features.

3.5 Design of the Classifier

The classifier is based on a FFNN, which contains non-linear nodes organized in one input layer, one hidden layer and one output layer. Several features were tested as well as several numbers of hidden layers of the neural network, in order to obtain the one that best fits the problem. The neural network was implemented using Matlab. We evaluated networks with 18, 20, 22, 24, 26, 28, 30, 40 and 50 hidden nodes. A sigmoid function was used as activation function.

A three-cross validation was used in order to get the best configuration of the network. For the experiments, 70% of the data was used for training, 15% was used for validation and 15% was used for testing. For the experiments reported here, performance was calculated as the percentage of correct hits of the classifier over the total of hits.

4 Results

We present the results obtained for the R-CNN applied for automatic segmentation, and the classification using a FFNN with manual segmentation as well as a classification using a FFNN with automatic segmentation when the obtained ROI's obtained a good reliability. Results are analyzed for each case of feature combination and for each dataset.

4.1 Evaluation of Automatic Segmentation

The R-CNN network was trained and adjusted using a set of 30 images. After that, it was tested using the 3 data sets named in Sect. 3.2 which are *Frontal-123*, *CorrectRanges-84* and *Representatives-6*. This evaluation was measured in three ways:

- (a) Reliability score, provided by the “Matlab application for object detection,” which returns a value in the range [0.0, 1.0].
- (b) Visual evaluation score, which assigns the following values according to the success of the segmentation:
 1. Breast was not detected.
 2. Breast was partially detected (less than 80% of the correct area).
 3. Breast was correctly detected (more than 80% of the correct area).

The average value was calculated as the assigned visual values of each image divided by 3.

- (c) Percentage of acceptance. This considers an image segment as “acceptable” when it covers at least 80% of the correct ROI.

Table 2 presents the results of these measurements for all datasets. As expected, the database *Representative-6* obtains the best results of segmentation.

4.2 Evaluation of Classification

Tables 3, 4 and 5 presents the results of three-fold validation for all databases and the 3 combinations of features, as described in Sect. 3.4, using manual segmentation. From such tables, we can notice that in the best images (*Representative-6*) the classifier obtains perfect classification when using two and three classes. The worst classification is obtained using the database *Frontal-123* with 3 classes. It was observed during the validation that images belonging to TH3 were confused with TH4, which, according to our clustering were part of different levels of risks (see

Table 2 Performance obtained by the automatic segmentation

Database	Score	First ROI	Second ROI
Frontal-123	Reliability	2.72	2.35
	Visual evaluation	0.95	0.79
	Percentage of acceptance	94.2	81.2
CorrectRanges-84	Reliability	2.75	2.42
	Visual evaluation	0.96	0.82
	Percentage of acceptance	91.5	83.3
Representatives-6	Reliability	2.66	2.66
	Visual evaluation	0.99	0.94
	Percentage of acceptance	100	100

Table 3 Performance obtained by the classifier using manual segmentation and features-case 1

Features-Case 1	2 classes		3 classes	
	Training (%)	Testing (%)	Training (%)	Testing
Frontal-123	70.7	83.3	58.6	55.6
CorrectRanges-84	100	87.5	63.8	61.5
Representatives-6	100	100	100	100%

Table 4 Performance obtained by the classifier using manual segmentation and features-case 2

Features-case 2	2 classes		3 classes	
	Training (%)	Testing (%)	Training (%)	Testing (%)
Frontal-123	87.9	75.0	55.2	55.6
CorrectRanges-84	95.0	75.0	74.1	53.8
Representatives-6	100	100	100	100

Table 5 Performance obtained by the classifier using manual segmentation and features-case 3

Feature set 3	2 classes		3 classes	
	Training (%)	Testing (%)	Training (%)	Testing (%)
Frontal-123	72.4	83.3	49.4	50
CorrectRanges-84	82.5	87.5	51.7	53.8
Representatives-6	100	100	100	100

Sect. 3.2). With respect to training, we notice that the network was not able to obtain a 100% training rate for the datasets *Frontal-123* and *CorrectRanges-84*.

The classifier was also executed using the ROIs automatically obtained in dataset *Representative-6*, obtaining also a 100% of performance for the three cases of features.

5 Conclusions and Future Work

In this paper we present the results of using a private thermographic database for the identification of 2 and 3 levels of risk of breast cancer. The database contains images with unbalanced classes taken with different calibrations and color codes, therefore a selection and preprocessing of the images was executed, in order to enhance them and to produce datasets with different quality and complexity in the separation of classes. Indeed, several features were designed to represent the subjective visual rules applied by a physician for the identification of the risk levels. Combinations of these features were tested using data with three different levels of complexity. Also, a deep R-CNN neural network was implemented for the automatic segmentation of breast in the thermographic image.

From the experiments executed so far, we conclude that when images are obtained in ideal conditions, as the ones contained into set *Representatives-6*, it is possible to obtain excellent results during classification, which suggest that the features selected may be representing the subjective rules followed by physicians. However, when images are noisy the classification results are poor, as is the case of data set *Frontal-123*, and the use of different combinations of features does not represent a significant difference on the performance. With respect to database *CorrectRanges-84*, the performance was fairly poor for 2 classes and less than acceptable for 3 classes. With respect to the automatic segmentation, for the case of high quality images (*Representative-6*) the result was excellent and fairly well when applying noisy images (*Frontal-123* and *CorrectRanges-84*). Even though, more research is required in order to improve such values. An important limitation for this study was that the number of images by some classes was very small, making difficult the training and generalization of the classifier.

Several activities may be executed in order to improve this work. First, the implementation of other features should be considered, as well as an improvement in the enhancement methods applied to the images, in order to obtain the best of them. With respect to the definition of the relations among levels of risks with respect to the original labeling TH1 to TH5, a non-supervised clustering may be applied, in order to identify which labeling are closer to each other. Another important work will be to evaluate the performance of the CNN network for classification of the images. Indeed, it is possible to apply some automatic methods for balancing the classes in TH-CEPREC may be applied, in order to improve the training of classifiers.

Acknowledgements The authors would like to thank Dr. Francisco Gutierrez and the rest of the staff of CEPREC for their kindly support and advice during the development of this research. This work was supported by INAOE. D. Reynoso thanks the National Council of Science and Technology in Mexico (CONACYT) for the scholarship provided during the development of this work.

References

1. U. S. Food and Drug Administration. Mammography: What You Need to Know, 27 October 2017. [Online]. Available: <https://www.fda.gov/ForConsumers/ConsumerUpdates/ucm420463.htm>. Accessed 16 Febrero 2018
2. U.S. Food and Drug Administration.: Breast Cancer Screening: Thermogram No Substitute for Mammogram, 30 10 2017. [Online]. Available: <https://www.fda.gov/ForConsumers/ConsumerUpdates/ucm257499.htm>. Accessed 16 Febrero 2018
3. Centro de Estudios y Prevención del Cáncer A.C.: Centro de Estudios y Prevención del Cáncer, A.C. (2018) [Online]. Available: <http://ceprec.org/index/index.html>. Accessed 13 Febrero 2019
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH (2014)
5. Gonzalez-Hernandez, J., Recinella, A., Kandlikar, S., Dabydeen, D., Medeiros, L., Phatak, P.: Technology, application and potential of dynamic breast thermography for the detection of breast cancer. *Int. J. Heat Mass Trans.* **131**, 558–573 (2019)
6. Koay, J., Herry, C., Frize, M.: Analysis of breast thermography with an artificial neural network. In: 26th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEMBS '04, Vol. 1, pp. 1159–1162 (2004)
7. Sossa Azuela, J.H., Rodríguez Morales, R.: Capítulo 2. Proceso de captación y formación de una imagen. In: *Procesamiento y Análisis Digital de Imágenes*, Madrid, Alfaomega pp. 45–59 (2011)
8. Zare, I., Ghafarpour, A., Zadeh, H., Haddadnia, J., Mohammad, S., Isfahani, M.: Evaluating the thermal imaging system in detecting certain types of breast tissue masses. *Biomed. Res.* **27**(3), 670–675 (2016)
9. Hankare, P., Shah, K., Nair, D., Nair, D.: Breast cancer detection using thermography. *Int. Res. J. Eng. Technol.* **3**(4), 2356–2395 (2016)
10. Reynoso Armenta, D.M.: Diagnosis of breast cancer through the processing of thermographic images and Neural Networks. Master thesis in Optics, Tonantzintla, Puebla (2017)
11. Luna, J.G.V., Gutierrez Delgado, F.: Feasibility of new-generation infrared screening for breast cancer in rural communities. *US Obstetrics and Gynecology, Touch Briefings.* **5**, 52–56 (2010)
12. Hobbins, W.: In abnormal thermogram-significance in breast cancer. *Interamer. J Rad.* **12**, 337 (1987)
13. Flir®.: Veterinary applications of thermography. Accessed 10 November 2017. [Online]. Available: http://www.flir.com/uploadedFiles/Thermography/MMC/Brochures/T820340/T820340_EN.pdf
14. MathWorks®.: Help « rgb2gray, » . Available: <https://www.mathworks.com/help/matlab/ref/rgb2gray.html>. Accessed 31 October 2017
15. MathWorks®.: Object Detection Using Deep Learning. Available: <https://www.mathworks.com/help/vision/examples/object-detection-using-deep-learning.html?requestedDomain=www.mathworks.com>. Accessed 16 Febrero 2018]

Robotic Applications

Acceleration of Path Planning Computation Based on Evolutionary Artificial Potential Field for Non-static Environments



Ulises Orozco-Rosas, Kenia Picos and Oscar Montiel

Abstract In this work, a mobile robot path-planning algorithm based on the evolutionary artificial potential field (EAPF) for non-static environments is presented. With the aim to accelerate the path planning computation, the EAPF algorithm is implemented employing novel parallel computing architectures. The EAPF algorithm is capable of deriving optimal potential field functions using evolutionary computation to generate accurate and efficient paths to drive a mobile robot from the start point to the goal point without colliding with obstacles in static and non-static environments. The algorithm allows parallel implementation to accelerate the computation to obtain better results in a reasonable runtime. Comparative performance analysis in terms of path length and computation time is provided. The experiments were specifically designed to show the effectiveness and the efficiency of the mobile robot path-planning algorithm based on the EAPF in a sequential implementation on CPU, a parallel implementation on CPU, and a parallel implementation on GPU.

Keywords Path planning · Evolutionary artificial potential field · Mobile robots · Graphics processing unit · Heterogeneous computing

U. Orozco-Rosas · K. Picos
CETYS Universidad, Centro de Innovación y Diseño (CEID), Av. CETYS Universidad No. 4, El Lago, 22210 Tijuana, Baja California, Mexico
e-mail: ulises.orozco@cetys.mx

K. Picos
e-mail: kenia.picos@cetys.mx

O. Montiel (✉)
Instituto Politécnico Nacional, CITEDIPN, Av. Instituto Politécnico Nacional No. 1310, Nueva Tijuana, 22435 Tijuana, Baja California, Mexico
e-mail: oross@ipn.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_22

1 Introduction

The path planning task is one of the most critical processes in terms of computation time for autonomous navigation in mobile robots. In this work, the mobile robot path-planning algorithm based on the evolutionary artificial potential field (EAPF) is implemented on parallel architectures to accelerate the path planning computation. The EAPF algorithm is capable of generating an optimal collision-free path to take the robot from the start to the goal point [1].

The main objective of this work is to present a comparative performance analysis in terms of path length and computation time for the different implementations on the central processing unit (CPU) and the graphics processing unit (GPU) for path planning in static and non-static environments. The experiments were performed over four different test environments. These environments were specifically designed to test the effectiveness and the efficiency of the EAPF algorithm in a sequential implementation on CPU, a parallel implementation on CPU, and parallel implementation on GPU. For offline path planning in static environments and online path planning in non-static environments considering new static obstacles and dynamic obstacles.

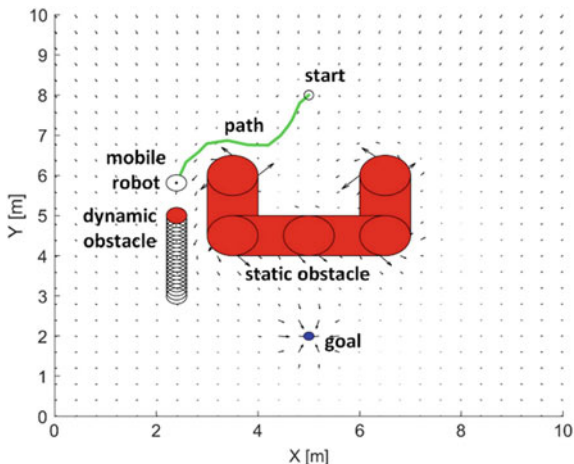
The organization of this work is as follows: in Sect. 2, the path planning problem is stated. In Sect. 3, a review of the artificial potential field (APF) method is given. In Sect. 4, the evolutionary artificial potential field (EAPF) algorithm is defined. In Sect. 5, the parallel evolutionary artificial potential field (parallel EAPF) algorithm is described. In Sect. 6, the mobile robot path-planning algorithm based on the sequential/parallel EAPF for non-static environments is presented. In Sect. 7, the experimental framework is explained, and the comparative performance analysis is presented. Finally, in Sect. 8, the conclusions of this work are given.

2 Path Planning Problem Formulation

Path planning is a problem that requires finding a continuous path Q_G between a given start point q_0 and a goal point q_f for a mobile robot, subject to a variety of constraints [2]. Under this general formulation, in this work, we assume that the mobile robot environment *map* is conceived as a two-dimensional map, which includes a set of obstacles, where the obstacles can be static or dynamic [3], as it can be seen in Fig. 1.

Let, q be the instantaneous position of the mobile robot, which is represented by its coordinates, radius, and orientation, i.e., $q(x, y, r, \theta)$. In the mobile robot model, it has been assumed a circular occupancy area, as is exemplified in Fig. 1. The path planning goal is to find a feasible sequence Q_G of configurations (path) that can drive the mobile robot from the start point q_0 to the goal point q_f in the environment *map* [4].

Fig. 1 Path planning problem formulation



3 Artificial Potential Field Method

Path planning is a widely studied problem with many challenges prevailing [5]. The path planning problem has been addressed from different approaches: one of these approaches is the artificial potential field (APF) method [6]. In 1985, Khatib proposed the APF method for local planning [7]. The main idea of the APF method is to establish an attractive potential field, $U_{att}(q)$, around the goal point, as well as to establish a repulsive potential field, $U_{rep}(q)$, around the obstacles. The two potential fields, attractive and repulsive form the total potential field, $U_{total}(q)$, [8].

In the APF method, the mobile robot is considered as a point in the configuration space [9], and it moves under the influence of the total potential field, $U_{total}(q)$, described by (1). The APF method searches the falling direction of the potential field to find a collision-free path, which is built from the start point to the goal point [10]. The APF method is operated by the gradient descent method, which is directed toward minimizing the total potential field $U_{total}(q)$ in each position of the mobile robot. Therefore, the goal point represents an attractive pole and the obstacles represent repulsive forces in the surface where the mobile robot will perform its movement.

The total potential field $U_{total}(q)$ is the sum of the attractive potential field, $U_{att}(q)$, and the repulsive potential field, $U_{rep}(q)$, and it is described by (1),

$$U_{total}(q) = U_{att}(q) + U_{rep}(q). \quad (1)$$

The attractive potential field $U_{att}(q)$ is described by (2),

$$U_{att}(q) = \frac{1}{2}k_a(q - q_f)^2, \quad (2)$$

where q represents the mobile robot position vector in a workspace of two dimensions. The vector q_f represents the goal coordinates, and k_a is a positive scalar-constant that represents the proportional gain of the attractive potential field function. The expression $(q - q_f)$ is the distance between the robot position and the goal position.

The repulsive potential field $U_{rep}(q)$ has a limited range of influence; that condition prevents the movement of the mobile robot will be affected by distant obstacles. The repulsive potential field $U_{rep}(q)$ is described by (3),

$$U_{rep}(q) = \begin{cases} \frac{1}{2}k_r \left(\frac{1}{\rho} - \frac{1}{\rho_0} \right)^2 & \text{if } \rho \leq \rho_0 \\ 0 & \text{if } \rho > \rho_0 \end{cases} \quad (3)$$

where ρ_0 represents the limit distance of influence of the repulsive potential field, ρ is the shortest distance to the obstacle, and k_r is a positive scalar-constant that represents the proportional gain of the repulsive potential field function.

The generalized total force $F_{total}(q)$ is obtained by the negative gradient of the total potential field, $U_{total}(q)$, and it is described by (4),

$$F_{total}(q) = -\nabla U(q). \quad (4)$$

The APF method requires the appropriate values of the proportional gains (k_r and k_a) to make the mobile robot to successfully reach the goal point. In that sense, many ways can be used to know the adequate value of this proportional gains, the most common methods are mathematical analysis and approximate methods, e.g., evolutionary computation [1]. In this work, a genetic algorithm (GA) is employed to search for the appropriate values of the proportional gains [11].

4 Evolutionary Artificial Potential Field Algorithm

In 2000, Vadakkepat et al. proposed the evolutionary artificial potential field (EAPF) to derive optimal potential field functions using a genetic algorithm (GA) [12]. In the EAPF method, the APF is blended with a GA to find the optimal values for the proportional gains, k_r and k_a . The GA is an adaptive heuristic search algorithm premised on the evolutionary ideas of natural selection and genetic [13]. The basic concept of a GA is designed to simulate the process in natural systems necessary for the evolution [14], wherein the most basic form, a GA can be algorithmically modeled for computer simulation using the difference equation described by (5),

$$P(t + 1) = s(v(P(t))) \quad (5)$$

where t represents the time, the new population $P(t + 1)$ is obtained from the current population $P(t)$ after it was operated by random variation v , and selection s [15].

Fig. 2 Bit-string representation of a chromosome

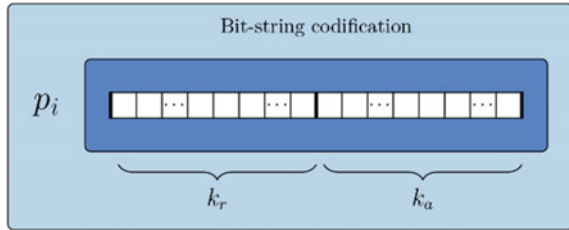


Figure 3 shows the EAPF algorithm for path planning [1]. At the beginning, $t = 0$, the EAPF algorithm performs the creation of a random initial population of chromosomes, $P(t)$. The initial population is composed by chromosomes $p_i, i = 1, 2, \dots, n$, where n is the number of chromosomes in the population, $n \in \mathbb{N}$ [8]. Each chromosome p_i is codified with a pair of proportional gains (see Fig. 2) in the form described by (6),

$$p_i = [k_r, k_a]. \tag{6}$$

After the initial population creation, the APF method for path generation and evaluation is performed in sequential form. Figure 3 shows that the APF method is composed of three stages. First, the total potential field is computed employing (1). Then, the total potential force is computed by (4). Last, the path length S is computed by (7). The path length is the distance to travel by the mobile robot from the start point to the goal point [16]. The path length S is computed by the evaluation function described by (7),

$$S = \sum_{i=0}^m \|q_{i+1} - q_i\| \tag{7}$$

where m is the number of mobile robot configurations from the start point to the goal point.

Once that the APF method for path generation and evaluation has been applied, the EAPF algorithm continues with the evolutionary process to evolve the population $P(t)$ through the genetic operators of selection, crossover, and mutation of the GA [17].

The selection operator drives the GA to improve the population fitness over the successive generations [8]. In the selection operator, the best chromosomes are chosen according to their evaluation value. The higher the evaluation value is, the more chance a chromosome must be selected. Hence, selection tends to eliminate those chromosomes that present a weak evaluation value. The selection mechanism determines which chromosomes will be retained as parents for being used by the crossover operator, and the genetics underlying their behavioral traits are passed to their offspring [15].

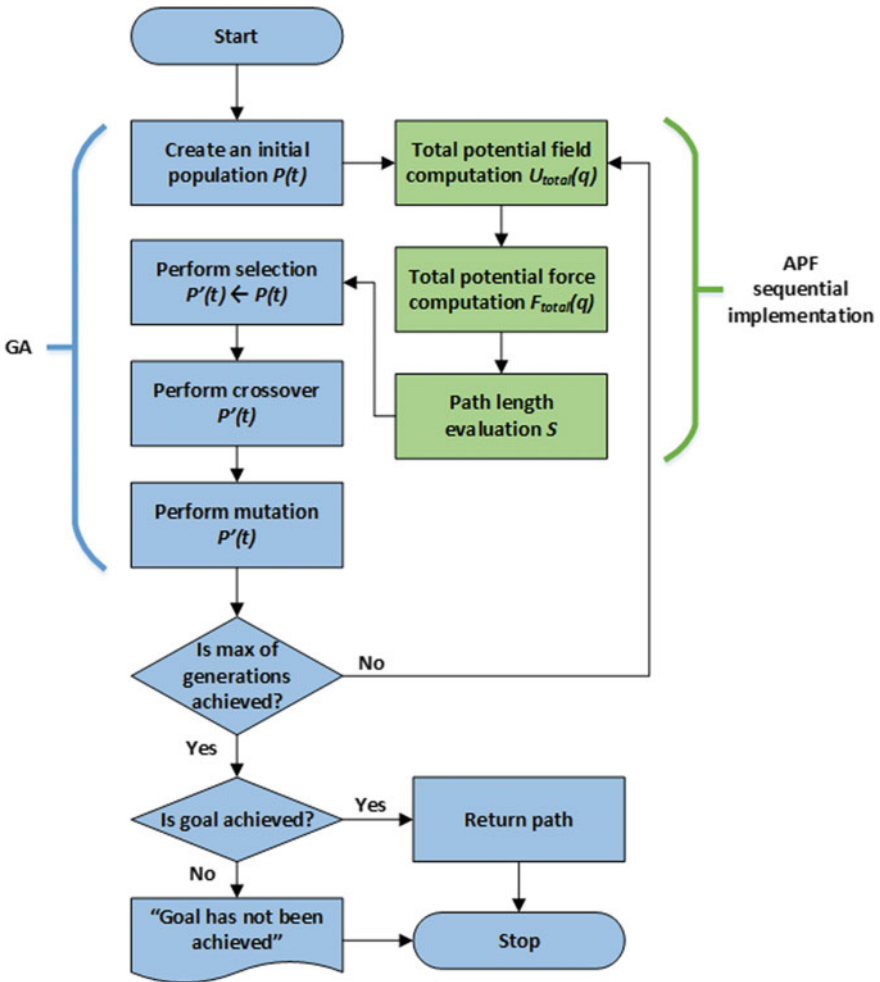


Fig. 3 Evolutionary Artificial Potential Field (EAPF) algorithm for path planning

In this work, in the EAPF algorithm of Fig. 3, the strategy of elitist selection is applied over the population, where the chromosomes with the best evaluation values (best chromosomes of the population) are chosen as parents guaranteeing a place in the next process. An asymptotic convergence is guaranteed by employing the elitist selection, which always retains the best chromosomes in the population [18].

After the selection process, the crossover operator is applied over the population $P(t)$. Holland [19] indicates that crossover provides the primary search operator. The crossover operator randomly chooses a locus and exchanges the subsequences before and after that locus between two chromosomes to create two offspring. For example, the strings 11111111 and 00000000 could be crossed over the fourth locus in each to

produce the two offspring 11110000 and 00001111. The crossover operator roughly mimics biological recombination between two single-chromosomes organisms [14].

A single point crossover is employed. The selected chromosomes generate offspring via the use of a crossover operator. Crossover is applied to two chromosomes called parents, which create two new chromosomes called offspring. This is done by selecting a random position along the coding and splicing the section that appears before the selected position. A new chromosome will be formed by the first string with the section that appears before the selected position of the first parent and the second string with the section that appears after the selected position of the second parent, and vice versa for the second new chromosome [20].

Then, after the termination of the crossover operator, the mutation operator is performed over the population $P(t)$. The mutation operator randomly alters a certain percentage of the bits in the list of chromosomes. Mutation is the second way in which a genetic algorithm explores a cost surface. This can introduce traits which are not in the original population, therefore, it keeps the genetic algorithm from converging prematurely before sampling the entire cost surface [21].

A single-point mutation changes a value of 1–0 and vice versa. In this work, mutation points are randomly selected from the total population. Increasing the number of mutations, the algorithm freedom to search outside the current region of the variable space also increases. It also tends to distract the algorithm from converging on local minima [8]. The genetic operators described above are illustrated in Fig. 4: these genetics operators are employed by the EAPF algorithm to evolve the proportional gains required by the APF method to generate the path (solution).

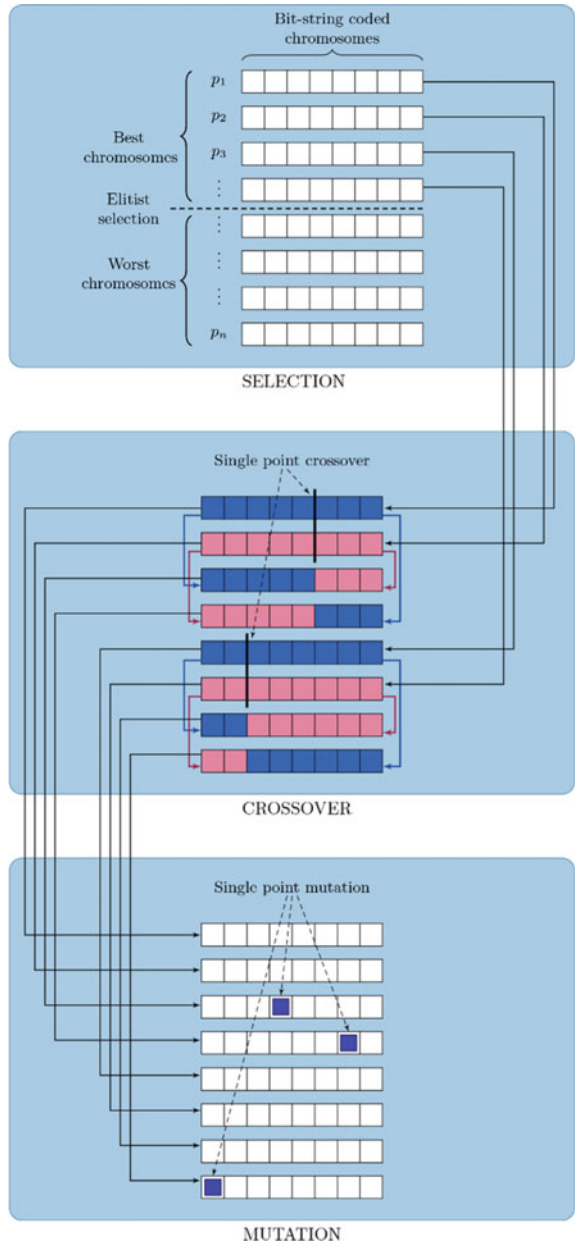
The EAPF algorithm of Fig. 3 iterates until the maximum number of generations is achieved. Therefore, the EAPF algorithm evolves through the GA the proportional gains that are codified in each chromosome to obtain the corresponding optimal values for the path generation through the APF method. Finally, if the goal point is achieved with the best chromosome of the population the generated path is returned; otherwise, a message of goal point not achieved is returned.

5 Parallel Evolutionary Artificial Potential Field Algorithm

Figure 5 shows the parallel EAPF algorithm for path planning. The parallel EAPF is an improved version of the EAPF algorithm presented in Fig. 3, where the APF method has been parallelized to accelerate the generation and evaluation path process. The APF method in the EAPF algorithm represents the most expensive process in computational time; that is the main reason for the parallel implementation. From simple computation perspective, parallel computing can be defined as a form of computation in which many calculations are carried out simultaneously, operating on the principle that large problems can often be divided into smaller ones, which are then solved concurrently [22].

In this work, the parallel EAPF algorithm is implemented on the different cores of the CPU or the GPU to obtain a feasible, safe and efficient path in reasonable run time

Fig. 4 Genetic operators employed by the EAPF algorithm



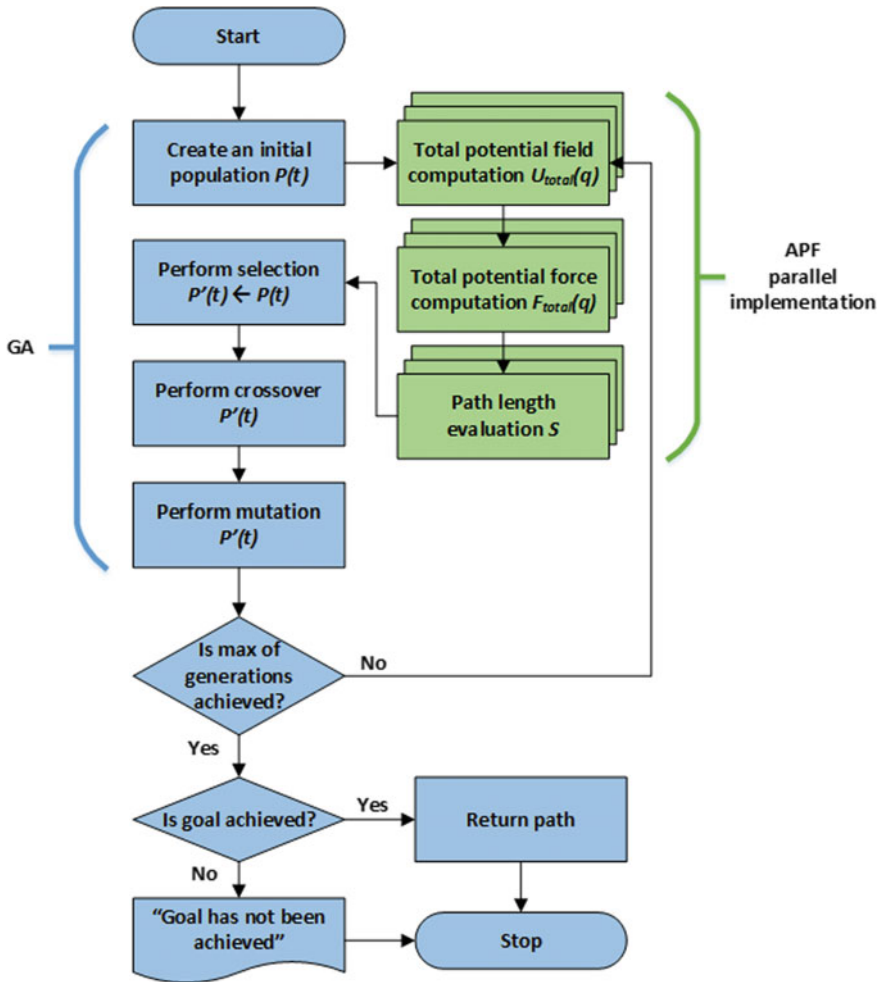


Fig. 5 Parallel Evolutionary Artificial Potential Field (parallel EAPF) algorithm for path planning

to drive the mobile robot to its goal. The parallel EAPF algorithm is the integration of the APF method with a GA and parallel computation for taking advantages of novel processors’ architectures (CPU/GPU).

The parallel EAPF algorithm employs parallel computation of the total potential field $U_{total}(q)$, parallel computation of the total potential force $F_{total}(q)$, and parallel computation of the path length evaluation S . The reasons are founded in taking advantages of novel processors’ architectures to counteract the expensive computational time required in the path generation and evaluation [8].

There are two fundamental types of parallelism in applications, task parallelism, and data parallelism. Task parallelism arises when many tasks or functions that can be operated independently and massively in parallel. Task parallelism focuses on

distributing functions across multiple cores. Data parallelism arises many data items can be operated on at the same time. Data parallelism focuses on distributing the data across multiple cores [22]. In the parallel EAPF algorithm, data parallelism was employed because there were many possible paths to evaluate, and each path is composed of data containing the values of the repulsive proportional gain k_r and the attractive proportional gain k_a .

6 Mobile Robot Path-Planning Algorithm Based on EAPF for Non-static Environments

Three tasks must be carried out by a mobile robot to enable the execution of the autonomous navigation [23]. These activities are: world perception consisting in mapping and modeling the environment, path planning for obtaining an ordered sequence of objective points and convert this sequence into a path, and the path tracking in controlling the robot to follow the path [24]. The selection of an appropriate algorithm in every stage of the path planning is very important to ensure that the autonomous navigation process will run smoothly [25].

The sequential/parallel EAPF algorithms described in Fig. 3 and Fig. 5 are employed as the main component of the mobile robot path-planning algorithm for non-static environments. The mobile robot path-planning algorithm is employed to obtain a feasible, safe and efficient path in reasonable run time to drive the mobile robot in an autonomous form to its goal. The parallel EAPF algorithm is the integration of the APF method [7] with evolutionary algorithms [12], in specific a GA, and parallel computation for taking advantages of novel processor architectures [8]. Due to the integration of the parallel EAPF to the mobile robot path-planning algorithm, we obtain a flexible path planner capable of performing the online mobile robot path planning for non-static environments [26].

Figure 6 shows the mobile robot path-planning algorithm for non-static environments using the parallel EAPF algorithm. The algorithm requires three input parameters to work: the start point q_0 of the mobile robot, the goal point q_f that the mobile robot must attain, and the environment configuration *map*. In the beginning, a configuration counter i is set to zero.

The process starts with the path planning using the sequential/parallel EAPF algorithm to generate the path Q_G . After that, the environment is sensed (perception) and the environment configuration *map* is updated. If the environment configuration has changed, the mobile robot path-planning algorithm starts from the beginning, but in this circumstance, the mobile robot start point q_0 will be the last point visited for the mobile robot during the navigation. Otherwise, if the configuration of the environment has not changed, the mobile robot will navigate one step, i.e., the mobile robot motion control will be performed to go from $Q_G(i)$ to $Q_G(i + 1)$.

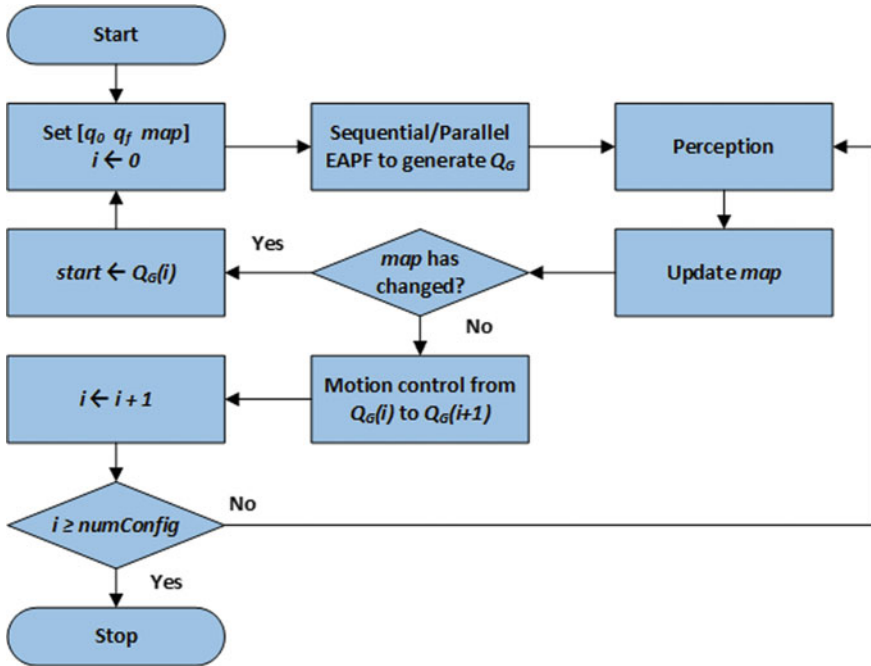


Fig. 6 Mobile robot path-planning algorithm based on sequential/parallel EAPF for non-static environments

The process will be repeated from the perception stage of the environment until the number of configurations *numConfig* is achieved. When the number of configurations has been achieved, the mobile robot has reached the goal location.

7 Experiments and Results

In this section, we describe the experiments and results of the mobile robot path-planning algorithm (Fig. 6) in offline mode for static environments and in online mode for non-static environments with new static and dynamic obstacles. Furthermore, a comparative study of three different implementations is presented, the sequential EAPF algorithm (Fig. 3) and the parallel EAPF algorithm (Fig. 5) in CPU and GPU implementations to evaluate the acceleration of the path planning computation.

To achieve the comparative, all the experiments were performed on a quad-core Intel i7-7700HQ CPU (2.80 GHz) with 8 GB of RAM, running Windows 10 with Visual Studio 2017, MATLAB 2018A, and CUDA 10.0. The GPU employed is an NVIDIA GeForce GTX 1060 (Pascal) with 1280 CUDA cores. The EAPF algorithm (Fig. 3) was implemented in C++, the CPU version of the parallel EAPF algorithm (Fig. 5) was implemented in C++ using the Open Multi-Processing (OpenMP)

application programming interface, the GPU version of the parallel EAPF algorithm (Fig. 5) was implemented using the Compute Unified Device Architecture (CUDA), and the mobile robot path-planning algorithm (Fig. 6) was implemented in C++.

For consistency among the experiments, we used the same parameter values for the sequential EAPF algorithm and the parallel EAPF algorithm in all the experiments. The next parameters yielded the best results for the implementations:

- The population size is varied from 16 to 512 chromosomes for the performance evaluation.
- Each chromosome consists of two genes, k_r and k_a .
- The proportional gains are constrained. Hence, $\{k_r, k_a | 0 < k_r, k_a \leq 10\}$.
- Single point crossover is employed. A random point in two chromosomes is chosen. Offspring are formed by joining the genetic material to the right of the crossover point of one parent, and the genetic material to the left of the crossover point of the other parent.
- The mutation rate was set to 0.40.
- The elitist selection was employed.
- The selection rate was set to 0.50.
- The stop condition is the maximum number of generations; it was set to 10.
- Considering that a mobile robot configuration is defined as $q(x, y, r, \theta)$. In all the experiments, we considered that the mobile robot in the start point q_0 is centered over the start point and oriented to the first point to visit.
- The maximum number of mobile robot configurations (*numConfig*) was set to 2000.

Figure 7 shows the test environments that indicates the mobile robot's mission with a start and goal point configuration, q_0 and q_f respectively; as well as the environment layout. In the beginning, the test environments are formed by fixed obstacles. Each test environment was designed to evaluate the performance and accuracy of the mobile robot path-planning algorithm with its sequential and parallel EAPF implementations.

Figure 7a shows the test Environment 1, that is constituted by one U-shaped obstacle. Where the mission of the mobile robot is to go from the start point q_0 at (5.00, 8.00) to the goal point q_f at (5.00, 2.00). Figure 7b shows the test Environment 2, that is constituted by two L-shaped obstacles. Where the mission of the mobile robot is established as start point q_0 at (2.00, 3.75) and goal point q_f at (8.00, 6.25). Figure 7c shows the test Environment 3, that is constituted by three obstacles, two L-shaped obstacles and one barrier, where the mission of the mobile robot is to go from the start point q_0 at (5.50, 9.00) to the goal point q_f at (4.50, 3.00). Figure 7d shows the test Environment 4, that is constituted by three obstacles, two L-shaped obstacles and one small barrier, where the mission of the mobile robot is established as start point q_0 at (2.00, 7.00) and goal point q_f at (8.00, 3.00).

Environment 1 to Environment 4 represent well-known difficult problems, e.g., trap sites due to local minima, path-following prediction problems, problematic areas to reach because the goal is very close to an obstacle, and other situations described in [27, 28]. These environments are challenging problems for testing path planning

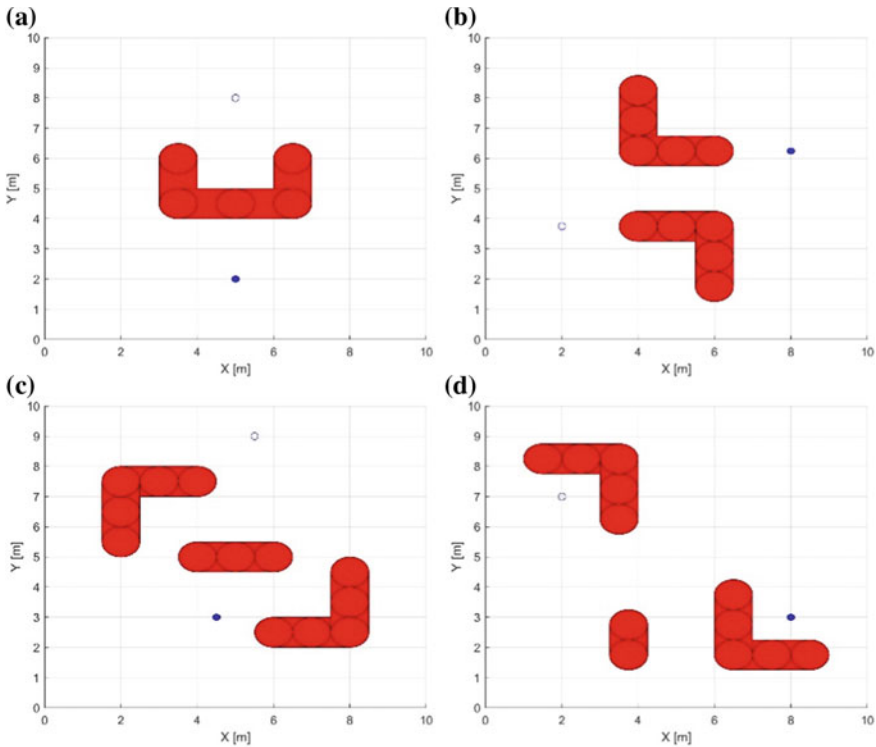


Fig. 7 Test environments, each environment indicates the start point, the goal, and the environment layout (obstacles configuration)

algorithms. These test environments represent just a sample of the types of scenarios that the mobile robot can expect to find in typical real-world indoor environments, which are composed of walls, corridors, barriers, U-shaped and L-shaped obstacles, among others.

7.1 Offline Path Planning

In offline path planning, all the obstacles on the environment are in known positions [29]. The purpose is to find a feasible collision-free path between the start point q_0 and the goal point q_f in a static environment composed of walls and obstacles. For this test, the cost of the path is obtained using the path length [2], which is described by (7).

The test environments depicted in Fig. 7 were employed to test the mobile robot path-planning algorithm described in Fig. 6 in offline mode, i.e., the test environment will be static with no changes. Figure 8 shows the best path (solution) for each test

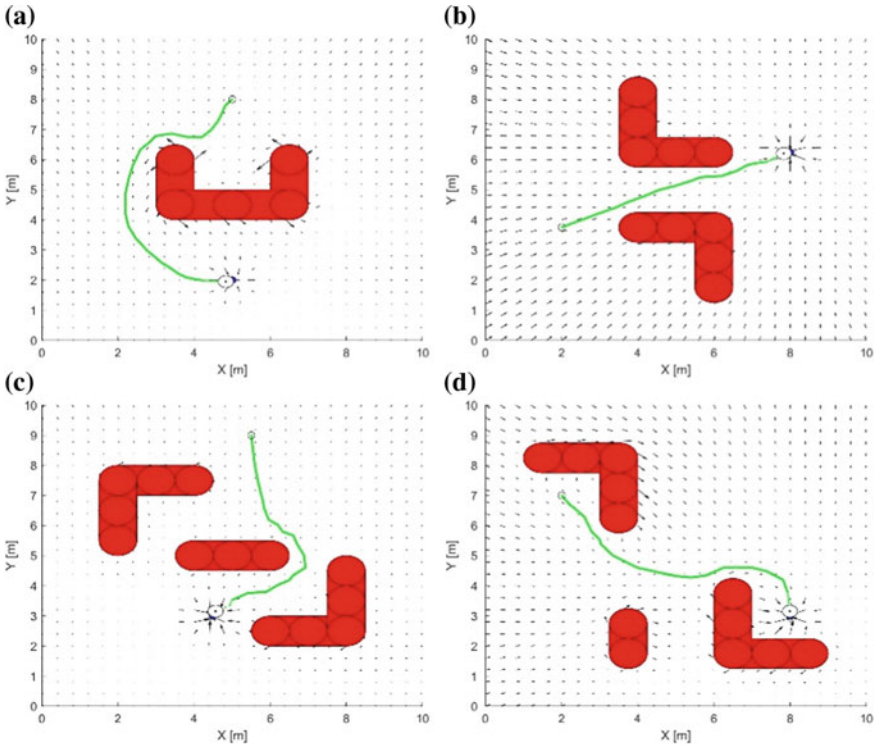


Fig. 8 Resultant path in each test environment obtained with the mobile robot path-planning algorithm in offline mode

environment. These paths have the shortest path lengths obtained with the algorithm of Fig. 6 in offline mode.

Table 1 shows the results obtained by the mobile robot path-planning algorithm described in Fig. 6 in offline mode with different population sizes. Therefore, the number of evaluations performed for the sequential/parallel EAPF algorithm in each test is based on the multiplication of the population size and the maximum number of generations, e.g., for a population size of 16 and a maximum number of generations (stop condition) of 10, we have $16 * 10 = 160$ evaluations.

Table 1 shows the proportional gains (k_r and k_a) for the best results obtained in terms of path length. The best, worst, average, and standard deviation of thirty independent test in each population size (16, 32, ..., 512).

In Table 1, the results show that the best paths were found by the mobile robot path-planning algorithm with a population of 512 chromosomes. It can be observed that the quality of the solutions (shorter paths) increase as the population size grew up.

Table 2 shows the performance results of thirty independent run test for the mobile robot path-planning algorithm in offline mode for each population size in each test environment. The objective was to provide the speedup among the two different

Table 1 Results of thirty independent run test for the mobile robot path-planning algorithm in offline mode for each population size in each test environment

	Population size	Prop. Gains	Path length (m)			
			Best	Worst	Average	Std. Dev.
Environment 1	16	(3.385, 0.560)	9.376	11.830	9.735	0.535
	32	(9.005, 1.527)	9.365	10.210	9.552	0.181
	64	(6.299, 1.060)	9.345	10.097	9.485	0.163
	128	(6.488, 1.092)	9.343	9.629	9.428	0.071
	256	(7.689, 1.305)	9.334	9.472	9.385	0.032
	512	(5.886, 0.992)	9.330	9.396	9.357	0.018
Environment 2	16	(8.893, 8.572)	6.602	6.672	6.631	0.093
	32	(2.156, 2.162)	6.454	6.628	6.586	0.082
	64	(1.115, 1.417)	6.426	6.607	6.511	0.073
	128	(1.220, 2.100)	6.423	6.604	6.471	0.053
	256	(0.658, 1.193)	6.412	6.497	6.444	0.029
	512	(0.420, 1.015)	6.390	6.466	6.426	0.017
Environment 3	16	(3.811, 1.434)	7.795	8.609	7.909	0.159
	32	(3.051, 1.151)	7.775	8.278	7.863	0.098
	64	(2.187, 0.820)	7.761	7.953	7.808	0.035
	128	(2.103, 0.793)	7.756	7.884	7.792	0.023
	256	(0.938, 0.349)	7.744	7.806	7.779	0.015
	512	(0.951, 0.354)	7.730	7.790	7.773	0.013
Environment 4	16	(0.878, 0.308)	8.419	9.307	8.788	0.285
	32	(2.814, 1.101)	8.324	8.451	8.373	0.040
	64	(2.825, 1.111)	8.311	8.383	8.355	0.022
	128	(2.138, 0.831)	8.307	8.349	8.323	0.015
	256	(2.205, 0.862)	8.304	8.334	8.316	0.010
	512	(2.146, 0.821)	8.301	8.333	8.313	0.009

parallel implementations, parallel CPU and parallel GPU. In parallel computing, the speedup refers to how much a parallel algorithm is faster than its equivalent in the sequential algorithm.

The speedup is defined by a relation described by (8),

$$\text{Speedup} = \frac{T_1}{T_p} \quad (8)$$

where T_1 is the execution time in a processor and T_p is the runtime on N_p processors. In this work, we have considered T_1 as the execution time taken by the mobile robot path-planning algorithm using the sequential EAPF algorithm, and T_p as the

Table 2 Performance results of thirty independent run test for the mobile robot path-planning algorithm in offline mode for each population size in each test environment

	Population size	Sequential CPU (s)	Parallel CPU (s)	Speedup	Parallel GPU (s)	Speedup
Environment 1	16	1.333	0.300	4.444	2.033	0.656
	32	2.567	0.567	4.529	2.133	1.203
	64	4.867	0.833	5.840	2.267	2.147
	128	10.033	1.700	5.902	2.400	4.181
	256	19.867	3.067	6.478	2.467	8.054
	512	40.933	5.833	7.017	3.700	11.063
Environment 2	16	0.900	0.200	4.500	2.100	0.429
	32	1.800	0.300	6.000	2.200	0.818
	64	3.500	0.600	5.833	2.400	1.458
	128	7.100	1.100	6.455	2.600	2.731
	256	14.200	1.900	7.474	2.700	5.259
	512	28.500	3.700	7.703	3.500	8.143
Environment 3	16	1.267	0.300	4.222	2.067	0.613
	32	2.467	0.533	4.625	2.333	1.057
	64	5.400	0.933	5.786	2.500	2.160
	128	10.967	1.733	6.327	2.533	4.329
	256	22.267	3.200	6.958	2.600	8.564
	512	46.500	5.967	7.793	3.900	11.923
Environment 4	16	1.700	0.300	5.667	2.700	0.630
	32	3.000	0.600	5.000	2.500	1.200
	64	5.800	1.100	5.273	2.500	2.320
	128	11.900	1.900	6.263	2.600	4.577
	256	24.000	3.500	6.857	2.700	8.889
	512	49.400	8.000	6.175	5.00	9.880

execution time taken by the mobile robot path-planning algorithm using the parallel EAPF algorithm in CPU and GPU, respectively.

For the performance evaluation, we ran the mobile robot path-planning algorithm thirty times for the three different programming implementations to obtain the average time of execution, and the speedup among the different test environments in each population size.

For the performance evaluation, we ran the mobile robot path-planning algorithm thirty times for the three different programming implementations to obtain the average time of execution, and the speedup among the different test environments in each population size.

Table 2 shows a summary of the obtained results. It can be observed that the speedup was increased in both cases, parallel CPU and parallel GPU, when the

population size grows. These results are due to the increment of the number of evaluations. It can be clearly observed the advantages of the parallel CPU implementation for small population size, and the advantage is more significant for the GPU implementation with a large population size, where the best speedup was 11.923 in the test environment 3 for a population size of 512 chromosomes.

7.2 Online Path Planning Considering New Static Obstacles

Real-world applications frequently face the mobile robot with unknown or partially known environments, which requires the ability to respond and taking decisions without detrimental of controllability, this action is referred as online path planning [26]. Some works perform offline path planning to know the initial conditions of the environment, and once the navigation starts they change to the online mode in order to adjust the path already known, considering the modified environment [1].

For this experiment, the mobile robot path-planning algorithm based on the sequential/parallel EAPF (Fig. 6) is going to perform the online path planning with new random static obstacles placement. We start with a known environment; hence, the position of the static obstacles was established in advance, in the test environment. Once the best path was obtained in offline mode, the random interfering obstacles are placed on the path, afterward remaining static.

Figure 9 illustrates the online path planning experiment considering the new static obstacles. For this experiment, test Environment 3 (Fig. 7c) was employed, which is found in many real-life scenarios.

First, the path planning is performed in offline mode because we just know the environment information described by the original test Environment 3, see Fig. 8c. The minimum path length found to reach the goal is 7.730 m, using the best set of gain parameters (0.951, 0.354), k_r and k_a , respectively. As it is described in Table 1.

At position (5.800, 6.700), a new static obstacle is added to change the environment configuration. After a while, when the mobile robot has traveled a distance of 1.519 m, it reaches the position (5.674, 7.491). The mobile robot senses the new obstacle; it calculates the obstacle position to update the environment layout *map*, as shown in Fig. 9a.

The mobile robot path-planning algorithm based on the EAPF (Fig. 6) now has a different environment layout; hence, it is necessary to update the path by recalculating the set of gain parameters to update the path to reach the goal point.

Next, at position (7.000, 5.000) a second new static obstacle is added to change the environment configuration. After a while, when the mobile robot has traveled a distance of 2.210 m, it reaches the position (6.722, 5.752). The mobile robot senses the second new obstacle; it calculates the obstacle position to update the environment layout, as shown in Fig. 9b.

Then, at position (7.500, 1.500) a third, new static obstacle is added to change the environment configuration. After a while, when the mobile robot has traveled a distance of 6.181 m, it reaches the position (8.300, 1.507). The mobile robot senses

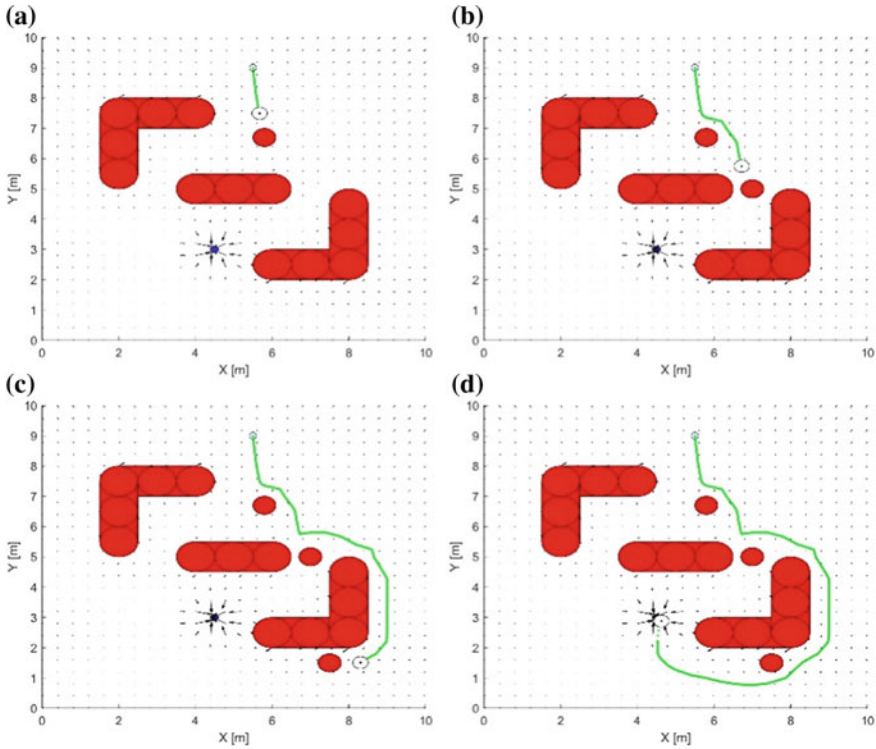


Fig. 9 Online path planning considering new static obstacles in test Environment 3

the third new obstacle; it calculates the obstacle position to update the environment layout, as shown in Fig. 9c.

Finally, the mobile robot follows the new path to reach the goal; the complete path is shown in Fig. 9d. The total path length from the original start point to the goal is $1.519 + 2.210 + 6.181 + 5.434 = 15.346$ m.

Table 3 shows a summary of the results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new static obstacles. In Table 3, the results show that the mobile robot path-planning algorithm found the best paths with a population of 512 chromosomes. It can be observed that the quality of the solutions (shorter paths) increase as the population of chromosomes grew up. Figure 10 shows the best resultant path for the online path planning considering a new static obstacle in the test Environment 1, 2, and 4, respectively.

Table 4 shows a summary of the performance results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new static obstacles. Table 4 shows that the speedup is increased in both cases, parallel CPU and parallel GPU, when the population size grows. Besides, it can be observed the advantages of the parallel CPU implementation for small population size, and the advantage is more significant for the GPU implementation with a large population

Table 3 Results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new static obstacles

	Population size	Path length (m)			
		Best	Worst	Average	Std. Dev.
Environment 1	16	9.823	10.260	12.505	0.323
	32	9.807	9.991	10.653	0.107
	64	9.788	9.909	10.509	0.092
	128	9.768	9.836	10.009	0.042
	256	9.764	9.791	9.850	0.021
	512	9.754	9.754	9.754	0.012
Environment 2	16	7.481	14.052	16.920	1.973
	32	7.261	14.672	16.734	1.712
	64	7.200	8.766	15.757	1.666
	128	7.193	9.880	15.942	2.056
	256	7.105	7.174	7.255	0.107
	512	7.070	7.070	7.070	0.073
Environment 3	16	16.066	18.333	16.912	0.250
	32	15.622	17.364	16.211	0.180
	64	15.536	16.753	16.058	0.133
	128	15.456	16.380	15.976	0.096
	256	15.454	15.916	15.710	0.081
	512	15.346	15.657	15.460	0.058
Environment 4	16	9.602	9.971	10.344	0.166
	32	9.472	9.553	9.503	0.052
	64	9.139	9.241	9.226	0.062
	128	8.975	9.096	9.039	0.052
	256	8.943	9.027	8.899	0.040
	512	8.910	8.910	8.910	0.041

size, where the best speedup was 12.718 in the test Environment 4 for a population size of 512 chromosomes.

7.3 Online Path Planning Considering New Dynamic Obstacles

These experiments consider the case of new dynamic obstacles that can be persons or other robots moving through the test environment. Differently to the case of random static obstacle placement, here the new obstacle will not remain static, it will

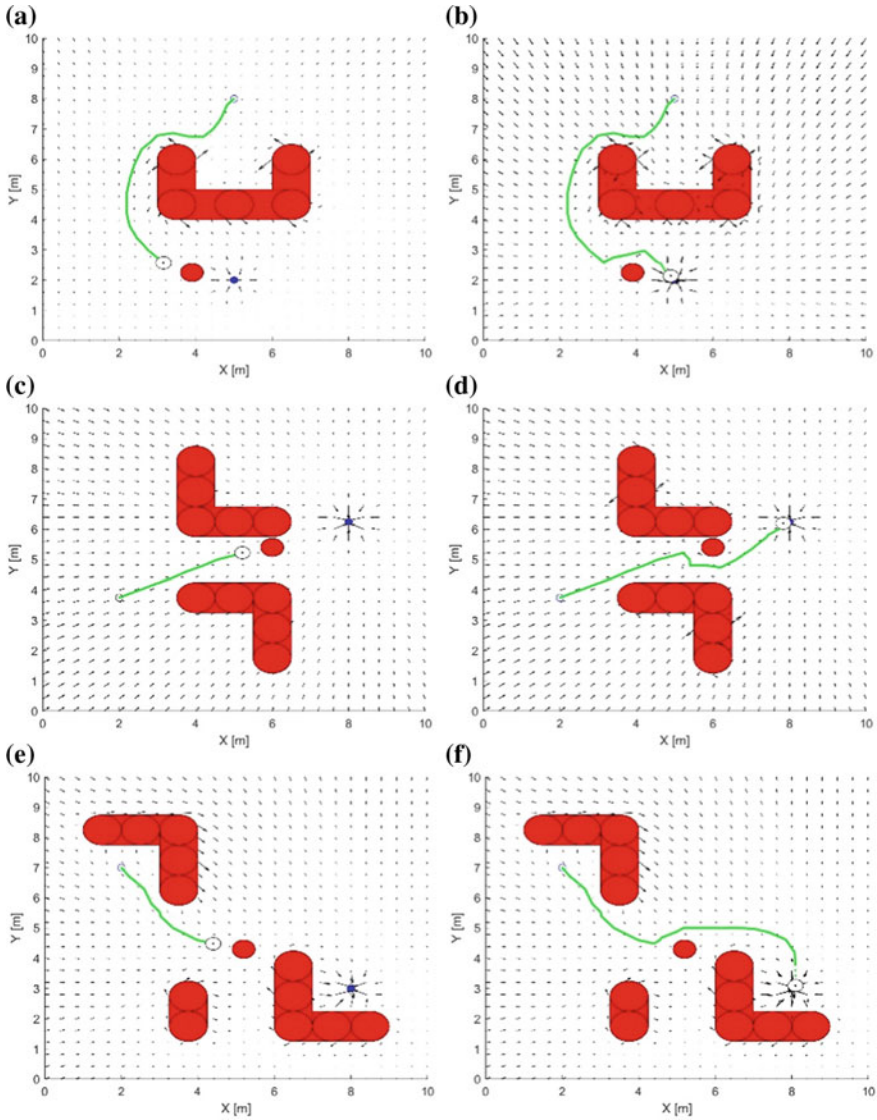


Fig. 10 Online path planning considering a new static obstacle in the test Environment 1, 2, and 4, respectively

be moving with a defined trajectory that is unknown to the mobile robot. This problem is more complicated than the scenarios considered in the online path planning considering new static obstacles because it faces the path-planning algorithm to find a feasible path in a noisy surface because at every iteration the searching zone is different [30].

Table 4 Performance results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new static obstacles

	Population size	Sequential CPU (s)	Parallel CPU (s)	Speedup	Parallel GPU (s)	Speedup
Environment 1	16	0.900	0.200	4.500	2.400	0.375
	32	1.700	0.400	4.250	2.400	0.708
	64	3.300	0.600	5.500	2.500	1.320
	128	7.100	1.100	6.455	2.600	2.731
	256	13.700	2.000	6.850	2.500	5.480
	512	28.400	3.500	8.114	2.900	9.793
Environment 2	16	1.500	0.300	5.000	2.600	0.577
	32	3.000	0.700	4.286	2.600	1.154
	64	5.800	1.300	4.462	2.500	2.320
	128	11.300	2.000	5.650	2.500	4.520
	256	22.800	4.100	5.561	2.700	8.444
	512	46.900	7.900	5.937	4.200	11.167
Environment 3	16	1.200	0.300	4.000	2.467	0.486
	32	2.767	0.633	4.368	2.433	1.137
	64	5.633	1.000	5.633	2.533	2.224
	128	11.300	1.900	5.947	2.567	4.403
	256	23.033	3.600	6.398	2.667	8.638
	512	46.767	7.033	6.649	3.933	11.890
Environment 4	16	1.600	0.300	5.333	2.600	0.615
	32	3.100	0.600	5.167	2.500	1.240
	64	5.800	1.200	4.833	2.600	2.231
	128	12.400	2.100	5.905	2.600	4.769
	256	23.500	3.900	6.026	2.700	8.704
	512	49.600	7.500	6.613	3.900	12.718

For this experiment, the mobile robot path-planning algorithm based on the sequential/parallel EAPF of Fig. 6. It is going to perform the online path planning with new random dynamic obstacles. We start with a known environment; therefore, the trajectories of the dynamic obstacles were established in advance, in the test environment. Figure 11 illustrates the online path planning experiment considering the new dynamic obstacles. For this experiment, test Environment 1 (Fig. 7a) was employed, which is found in many real-life scenarios.

First, the path planning is performed in offline mode because we know the environment information described by the original test Environment 1, see Fig. 8a. The minimum path length found to reach the goal is 9.330 m, using the best set of gain parameters (5.886, 0.992). As it is described in Table 1. After a while, when the mobile robot has traveled 3.903 m, it reaches the position (2.396, 5.813). The mobile

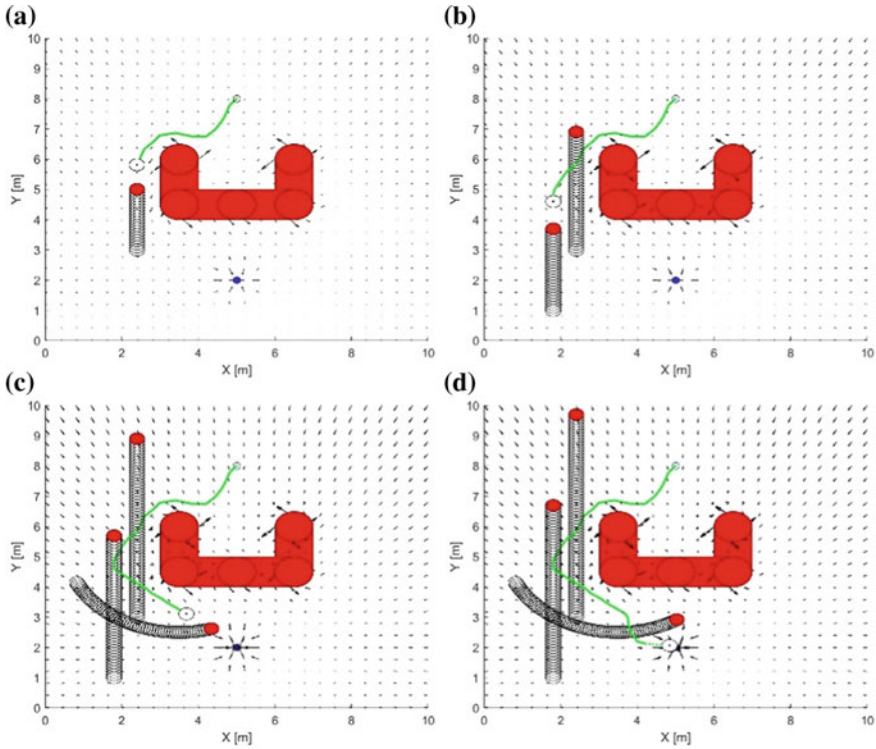


Fig. 11 Online path planning considering new dynamic obstacles in test Environment 1

robot senses a new obstacle, which is located at (2.400, 5.000) at that moment, the mobile robot calculates the obstacle position to update the environment layout map, as shown in Fig. 11a.

The mobile robot path-planning algorithm based on the EAPF (Fig. 6) now has a different environment layout; hence, it is necessary to update the path by recalculating the set of gain parameters to update the path to reach the goal point. After a while, when the mobile robot has traveled 1.413 m, it reaches the position (1.801, 4.598). The mobile robot senses the second new dynamic obstacle that is located at (1.800, 3.700) at that moment. The mobile robot calculates the obstacle position to update the environment layout, as shown in Fig. 11b.

Then, when the mobile robot has traveled 2.408 m, it reaches the position (3.684, 3.110). The mobile robot senses a third new dynamic obstacle, located at (4.337, 2.619) at that moment. The mobile robot calculates the obstacle position to update the environment layout, as shown in Fig. 11c.

Finally, the mobile robot follows the new path to reach the goal; the complete path is shown in Fig. 11d. The total path length from the original start point to the goal is $3.903 + 1.413 + 2.408 + 1.841 = 9.567$ m.

Table 5 shows a summary of the results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new dynamic obstacles. In Table 5, the results also show that the mobile robot path-planning algorithm found the best paths with a population of 512 chromosomes. Figure 12 shows the best resultant path for the online path planning considering a new dynamic obstacle in the test Environment 2, 3, and 4, respectively.

Table 6 shows a summary of the performance results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new dynamic obstacles. In addition, it can be observed the advantages of the parallel CPU implementation for small population size, and the advantage is more significant for

Table 5 Results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new dynamic obstacles

	Population size	Path length (m)			
		Best	Worst	Average	Std. Dev.
Environment 1	16	9.682	12.173	10.068	0.149
	32	9.649	10.541	9.857	0.058
	64	9.616	10.383	9.750	0.051
	128	9.612	9.855	9.681	0.022
	256	9.590	9.690	9.628	0.014
	512	9.567	9.587	9.577	0.009
Environment 2	16	7.138	7.195	7.319	0.045
	32	6.984	7.094	7.135	0.043
	64	6.951	7.013	7.111	0.048
	128	6.945	6.969	7.090	0.035
	256	6.930	6.925	6.939	0.018
	512	6.901	6.901	6.901	0.012
Environment 3	16	7.852	7.958	8.885	0.130
	32	7.831	7.987	8.874	0.161
	64	7.790	7.805	7.978	0.033
	128	7.757	7.759	7.905	0.030
	256	7.742	7.730	7.717	0.013
	512	7.720	7.720	7.720	0.014
Environment 4	16	10.199	10.702	11.823	0.321
	32	10.052	10.260	10.563	0.103
	64	9.920	9.984	9.999	0.043
	128	9.908	9.889	9.871	0.028
	256	9.888	9.913	9.962	0.038
	512	9.865	9.865	9.865	0.032

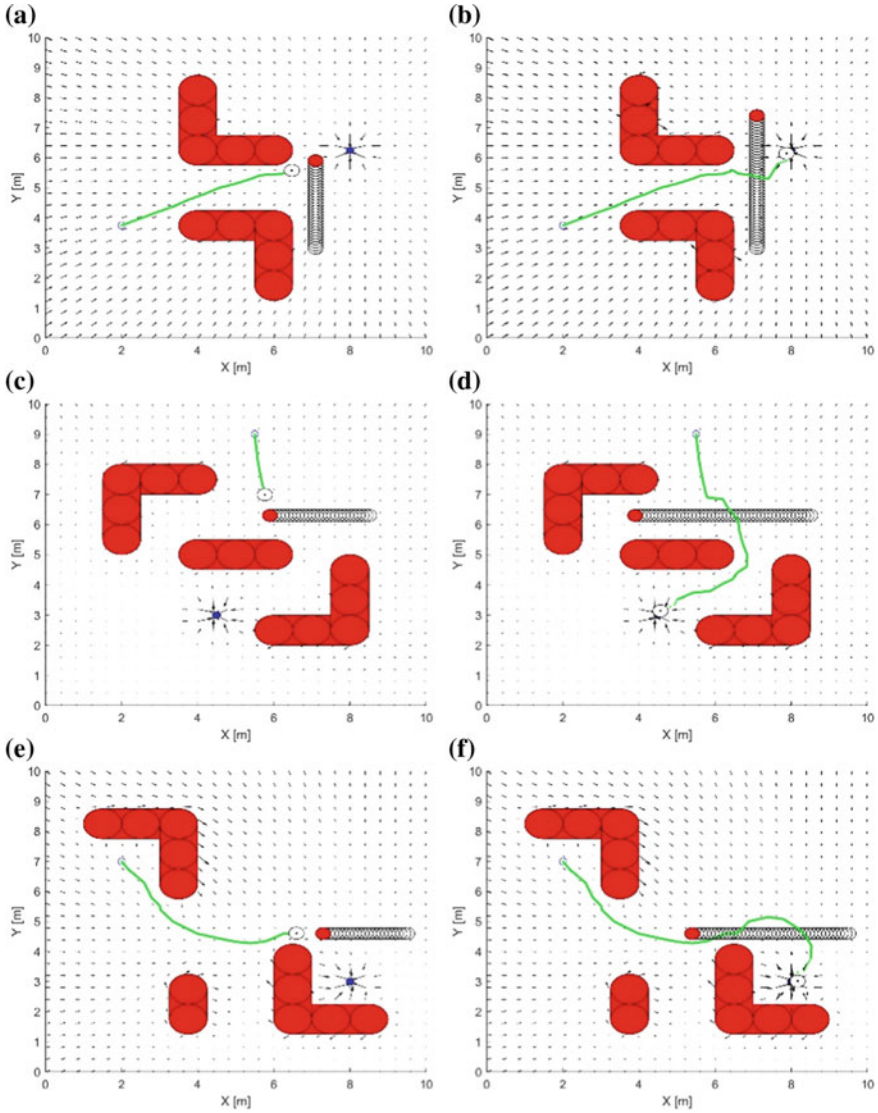


Fig. 12 Online path planning considering a new dynamic obstacle in the test Environment 2, 3, and 4, respectively

the GPU implementation with a large population size, where the best speedup was 11.277 in the test Environment 4 for a population size of 512 chromosomes.

Table 6 Performance results of thirty independent run test for the mobile robot path-planning algorithm in online mode considering new dynamic obstacles

	Population size	Sequential CPU (s)	Parallel CPU (s)	Speedup	Parallel GPU (s)	Speedup
Environment 1	16	0.633	0.200	3.167	2.367	0.268
	32	1.800	0.400	4.500	2.267	0.794
	64	3.333	0.600	5.556	2.400	1.389
	128	6.667	1.067	6.250	2.467	2.703
	256	14.000	2.033	6.885	2.500	5.600
	512	27.400	3.833	7.148	2.900	9.448
Environment 2	16	0.600	0.100	6.000	2.200	0.273
	32	1.000	0.300	3.333	2.500	0.400
	64	2.300	0.500	4.600	2.500	0.920
	128	4.500	0.700	6.429	2.600	1.731
	256	9.100	1.400	6.500	2.600	3.500
	512	18.000	2.700	6.667	2.700	6.667
Environment 3	16	0.700	0.300	2.333	2.100	0.333
	32	2.200	0.400	5.500	2.400	0.917
	64	3.500	0.800	4.375	2.500	1.400
	128	7.500	1.600	4.688	2.600	2.885
	256	16.300	2.700	6.037	2.700	6.037
	512	32.100	5.400	5.944	3.300	9.727
Environment 4	16	1.700	0.400	4.250	2.700	0.630
	32	3.200	0.700	4.571	2.600	1.231
	64	6.400	1.100	5.818	2.600	2.462
	128	12.700	2.100	6.048	2.600	4.885
	256	25.100	4.100	6.122	2.800	8.964
	512	53.000	8.300	6.386	4.700	11.277

8 Conclusions

In this work, we have seen how the mobile robot path-planning algorithm based on EAPF for non-static environments solves the offline and online path planning problem in all the different test environments and situations exposed. The results demonstrated that the parallel implementation is very powerful to accelerate the evaluation of the solutions, for the experiments evaluated in this work, we have reached the maximum acceleration of 8.114 for the parallel implementation in CPU, and a maximum acceleration of 12.718 for the parallel implementation in GPU.

Population sizing has been one of the most critical issues in evolutionary computation. Researchers usually argue that a small population size could guide the algorithm

to poor solutions and that a large population size could make the algorithm expend more computation time in finding a solution.

In this work, we have observed how the mobile robot path-planning algorithm based on EAPF for non-static environments solves the offline and online path planning problem with different population sizes. The results demonstrated that the parallel CPU and GPU implementations are powerful to accelerate the generation and evaluation of the solutions because the problem is stated as a data-parallel problem.

Due to the experimental results, we can conclude that the mobile robot path-planning algorithm based on EAPF for non-static environments can be capable of facing more complex and bigger offline and online planning problems, making the implementation suitable for real-world applications in complex environments.

Acknowledgments This research was supported in part by the Coordinación Institucional de Investigación of CETYS Universidad, and in part by the Consejo Nacional de Ciencia y Tecnología (CONACYT, Mexico).

References

1. Montiel, O., Sepúlveda, R., Orozco-Rosas, U.: Optimal path planning generation for mobile robots using parallel evolutionary artificial potential field. *J. Intell. Rob. Syst.* **79**(2), 237–257 (2015)
2. Devaurs, D., Siméon, T., Cortés, J.: Optimal path planning in complex cost spaces with sampling-based algorithms. *IEEE Trans. Autom. Sci. Eng.* **13**(2), 415–424 (2016)
3. Masehian, E., Amin-Naseri, M.R.: Sensor-based robot motion planning—a tabu search approach. *IEEE Robot. Autom. Mag.* **15**(2), 48–57 (2008)
4. LaValle, S.M.: *Planning Algorithms*. Cambridge University Press, New York (2006)
5. Masehian, E., Sedighzadeh, D.: Classic and heuristic approaches in robot motion planning a chronological review. *Int. J. Mech. Aerosp. Ind. Mechatron. Manuf. Eng.* **1**(5), 228–233 (2007)
6. Park, M.G., Lee, M.C.: A new technique to escape local minimum in artificial potential field based path planning. *KSME Int. J.* **17**(12), 1876–1885 (2003)
7. Khatib, O.: Real-time obstacle avoidance for manipulators and mobile robots. In: *Proceedings of the IEEE International Conference on Robotics and Automation* (1985)
8. Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: Parallel evolutionary artificial potential field for path planning—an implementation on GPU. In *Design of Intelligent Systems Based on Fuzzy Logic, Neural Networks and Nature-Inspired Optimization. Studies in Computational Intelligence*, vol. 601, pp. 319–332 (2015)
9. Montiel, O., Sepúlveda, R., Castillo, O., Melin, P.: Ant colony test center for planning autonomous mobile robot navigation. *Comput. Appl. Eng. Educ.* **21**(2), 214–229 (2013)
10. Zhang, Q., Chen, D., Chen, T.: An obstacle avoidance method of soccer robot based on evolutionary artificial potential field. *Energy Procedia* 16-C, 1792–1798 (2012)
11. Vadakkepat, P., Lee, T.H., Xin, L.: Application of evolutionary artificial potential field in robot soccer system. In: *Proceedings Joint 9th IFSA World Congress and 20th NAFIPS International Conference* (Cat. No. 01TH8569), Vancouver, BC, Canada (2001)
12. Vadakkepat, P., Kay, C.T., Wang, M.-L.: Evolutionary artificial potential fields and their application in real time robot path planning. In: *Proceedings of the 2000 Congress on Evolutionary Computation* (2000)
13. Sivanandam, S.N., Deepa, S.N.: *Introduction to Genetic Algorithms*. Springer, Heidelberg (2008)

14. Mitchell, M.: *An Introduction to Genetic Algorithms*, Cambridge, Massachusetts. Bradford, USA (2001)
15. Fogel, D.B.: An introduction to evolutionary computation. In: *Evolutionary Computation: The Fossil Record*, pp. 1–28. Wiley-IEEE Press (1998)
16. Orozco-Rosas, U., Picos, K., Montiel, O., Sepúlveda, R., Díaz-Ramírez, V.: Obstacle recognition for path planning in autonomus mobile robots. In: *Optics and Photonics for Information Processing X* (2016)
17. Tsai, C., Huang, H.C., Chan, C.K.: Parallel elite genetic algorithm and its application to global path planning for autonomous robot navigation. *IEEE Trans. Industr. Electron.* **58**(10), 4813–4821 (2011)
18. Fogel, D.B.: An introduction to simulated evolutionary optimization. *IEEE Trans. Neural Netw.* **5**(1), 3–14 (1994)
19. Holland, J.H.: *Adaptation in natural and artificial systems*, 2nd edn. MIT Press. Cambridge (1992). (First edition, Ann Arbor: University of Michigan Press, 1975)
20. Picos, K., Orozco-Rosas, U., Díaz-Ramírez, V.H., Montiel, O.: Pose estimation in non-continuous video sequence using evolutionary correlation filtering. *Math. Prob. Eng.* 1–14 (2018)
21. Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: High-performance navigation system for mobile robots. In: *High Performance Programmig for Soft Computing Application*, pp 258–281. CRC Press (2014)
22. Cheng, J., Grossman, M., McKercher, T.: *Professional CUDA C Programming*. Wrox-Wiley, Indianapolis, Indiana (2014)
23. Shin, D.H., Singh, S.: *Path Generation for Robot Vehicle Using Composite Clothoid Segments*. The Robotic Institute. Carnegie-Mellon University, Pittsburgh, Pennsylvania (1990)
24. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: *Introduction to Autonomous Mobile Robots*, 2nd edn. The MIT Press, Cambridge (2011)
25. Sariff, N., Buniyamin, N.: An overview of autonomous mobile robot path planning algorithms. In: *4th Student Conference on Research and Development SCORED 2006* (2006)
26. Orozco-Rosas, U., Montiel, O., Sepúlveda, R.: Mobile robot path planning using membrane evolutionary artificial potential field. *Appl. Soft Comput.* **77**, 236–251 (2019)
27. Ge, S.S., Cui, Y.J.: New potential functions for mobile robot path planning. *IEEE Trans. Robot. Autom.* **16**(5), 615–620 (2000)
28. Koren, Y., Borenstein, J.: Potential field methods and their inherent limitations for mobile robot navigation. In: *IEEE International Conference on Robotics and Automation* (1991)
29. Aghababa, M.P.: 3D path planning for underwater vehicles using five evolutionary optimization algorithms avoiding static and energetic obstacles. *Appl. Ocean Res.* **38**, 48–62 (2012)
30. Morales, N., Toledo, J., Acosta, L.: Path planning using a multiclass support vector machine. *Appl. Soft Comput.* **43**, 498–509 (2016)

Multi-objective Evaluation of Deep Learning Based Semantic Segmentation for Autonomous Driving Systems



Cynthia Olvera, Yoshio Rubio and Oscar Montiel

Abstract Recent applications of deep learning (DL) architectures for semantic segmentation had led to a significant development in autonomous driving systems (ADS). Most of the semantic segmentation applications for ADS consider a plethora of classes. Nevertheless, we believe that focusing only on the segmentation of drivable roads, sidewalks, traffic signs, and cars, can drive the improvement of navigation and control techniques in autonomous vehicles. In this study, some state-of-the-art topologies are analyzed to find a strategy that can achieve a uniform performance for the four classes. We propose a multiple objective evaluation method with the purpose of finding the non-dominated solutions in different DL architectures. Numerical results are shown using CityScapes, SYNTHIA, and CamVid datasets.

Keywords Semantic segmentation · Deep learning · Autonomous vehicles · Road detection

1 Introduction

Autonomous driving systems have attracted special attention in the last years, they are considered as one of the technologies that will have a significant impact on lives of citizens of industrialized countries in the near future [1]. Advanced driver assistance systems (ADAS) are a crucial technology in autonomous vehicles that has proved to be effective in reducing the risk of crashes and minimizing driver injuries. ADAS are primarily vision based, however, technologies such as light detection and

C. Olvera · Y. Rubio · O. Montiel (✉)

Instituto Politécnico Nacional, Centro de Investigación y Desarrollo de Tecnología Digital (IPN-CITEDI), Av. Instituto Politécnico Nacional no. 1310, Nueva Tijuana, 22435 Tijuana, B.C., Mexico

e-mail: oross@ipn.mx

C. Olvera

e-mail: colvera@citedi.mx

Y. Rubio

e-mail: rubio@citedi.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_23

ranging (LiDAR), radio detection and ranging (RADAR), ultrasonic ranging, and others advanced-sensing techniques are also popular [2].

Semantic segmentation in urban scenarios plays a key role in understanding driving environments and traffic conditions. Semantic segmentation assigns a categorical label to every pixel in an image and segments the whole image into different semantic parts [3]. It is a powerful tool for a wide range of applications such as robotic navigation [4], medical imaging [5, 6], remote sensing [7], human-machine interaction [8], and autonomous driving [9–12] among others.

The rapid development of deep learning and the advances in GPU technologies have driven different works on automated driving to emerge. Deep learning tools have obtained better results than traditional techniques for semantic segmentation and even achieved state-of-the-art performance [13–15]. Many recent works in segmentation techniques for autonomous vision grant the same level of importance to all the classes in the dataset [9, 10, 12]. We strongly believe that in autonomous driving, not all the classes are relevant [16]. The semantic segmentation algorithm should focus on obtaining high accuracy on the classes of interest (road, sidewalk, car, traffic sign) and not worrying much about the accuracy of the other classes where only their distance to the vehicle is of importance (obstacles in the road, pedestrians).

From this point of view, not all existing semantic segmentation methods can be suitable for dealing with autonomous driving and obtaining real-time performance. It is necessary to decide which of these techniques could be possibly implemented on a real-time system and to adapt these methods for the task at hand. In this work, we propose a method to evaluate the performance of different state-of-the-art architectures of convolutional neural networks (CNN or ConvNet) for semantic segmentation, using a variety of datasets of driving scenarios. The goal of the evaluation is to aid in the selection of the most useful classes, and the highest performing CNNs for semantic segmentation focused on the control of autonomous driving systems.

2 Related Work

Recent progress in semantic segmentation is mainly due to the active development of deep learning techniques, especially convolutional neural networks. Deep convolutional architectures have been successfully applied to many computer vision applications, including image classification [17–19], object detection [20, 21], scene labeling [22, 23], and semantic segmentation [13, 14].

The state-of-the-art methods for semantic segmentation are generally trained end-to-end for pixelwise prediction. Fully convolutional network (FCN) is the first-pixel level segmentation network trained end-to-end. In this approach, AlexNet [17], VGG [18], and GoogleLeNet [19] classifiers were converted to dense prediction FCN. Segmentation is achieved training by fine-tuning, fusing different layer representations through skip architecture, and learning end-to-end on whole images.

In [24] a detailed survey of semantic segmentation for autonomous driving systems is presented. DeconvNet [15] is a semantic segmentation architecture in which a

multilayer deconvolution network is placed on top of the convolution network. This decoder network generates dense and precise object segmentation masks through a sequence of unpooling, deconvolution and rectification operations. SegNet [14] is deep encoder-decoder architecture for semantic segmentation specifically designed to be efficient in terms of memory and computation time for scene understanding applications. The decoder network uses the max pooling indices of the corresponding encoder to perform non-linear upsampling.

Bayesian SegNet [25] arises from this idea. It is the first probabilistic framework for deep semantic segmentation with a measure of model uncertainty. U-net [26] is also an encoder-decoder network where the entire encoder feature map is concatenated to the corresponding upsampled decoder feature map. This architecture is highly popular in biomedical applications.

Siam et al. [27] presents a real-time segmentation framework and a study of several algorithms for autonomous driving. The availability of large-scale and diverse annotated datasets of urban scenarios is a major contributing factor to the success of DL based semantic segmentation systems for autonomous driving.

The popular semantic segmentation datasets for this application are KITTI [9], CamVid [10], Cityscapes [11] and SYNTHIA [12]. Each dataset has different characteristics and therefore produces different results. Nevertheless, the performance of the networks trained with these datasets is not satisfactory for a commercial autonomous driving system. Even though there is a growing effort in building larger datasets, and more efficient architectures, our interest is to find which combination of datasets and architectures achieves better performance.

3 Fully Convolutional Networks

Each layer output in a CNN is a three-dimensional array of size $h \times w \times d$, which refers to height, width and depth respectively. Each neuron is connected only to a local region of the input volume, which is called the receptive field. ConvNets are translation invariant, and their essential components depend only on relative spatial coordinates [13].

Typical classifiers [17–19] take fixed-size input images and output non-spatial data. The fully connected (FC) layers of these models have fixed dimensions and cause loss of spatial coordinates. However, FC layers can be expressed as $1 \times 1 \times k$ convolutions, where k is the number of kernels that cover the entire input region, and hence can cast these nets into FCNs able to take input images of any size and generate coarse output maps. Dense predictions can be obtained from these coarse outputs by applying shifted kernels and stitching together outputs in the direction of the rotation.

Another method to produce dense predictions is by upsampling the coarse outputs. Upsampling can be viewed as a convolution with a fractional input stride of $1/f$ where f is an integral factor. It is performed in-network through transposed convolutions by

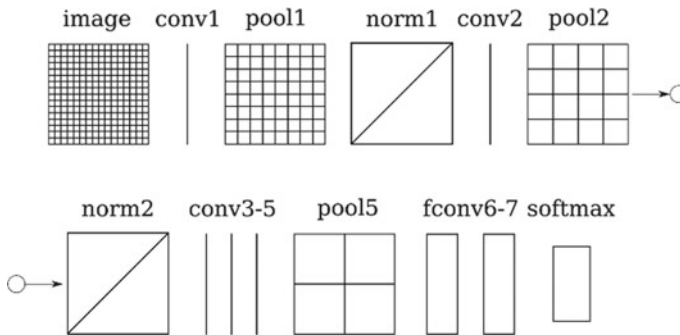


Fig. 1 FCN-Alexnet architecture. Pooling layers are represented as grids, convolutional layers as vertical lines, and fully convolutional layers as rectangles

reversing the forward and backward passes of a convolution for end-to-end learning from pixelwise loss.

Patchwise training and fully convolutional training can produce any distribution of inputs and their efficiency depends on overlap and batch size. For dense prediction, whole image fully convolutional training is preferred since it is significantly faster than patch sampling due to the large number of images to be considered per batch.

These convolutionalized networks are appropriate for dense problems such as semantic segmentation. The forward and backward passes are more efficient when computed layer-by-layer over an entire image since they take full advantage of the intrinsic computational efficiency of convolution.

3.1 FCN-Alexnet

Alexnet [17] is the pioneering deep CNN and consists of five convolutional layers, max pooling layers, rectified linear units (ReLUs), three FC layers, and a dropout. FCN-Alexnet is a convolutionalized version of this network, where the FC layers are converted to fully convolutional layers as shown on Fig. 1.

3.2 FCN-8s

FCN-8 [13] is an improved version of the FCN-VGG16. It consists of a downsampling path used to capture the semantic information, and an upsampling path, used to recover spatial information. The downsampling is built from convolutional, pooling, and fully convolutional layers; the upsampling path is built from backward convolution and a skip architecture that combines coarse and fine layers to produce accurately

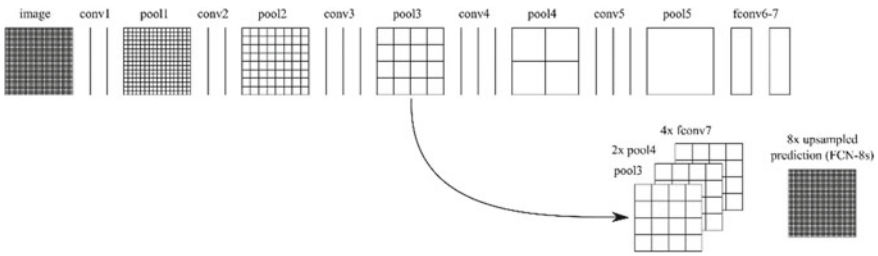


Fig. 2 FCN-8s architecture. Pooling and prediction layers are presented as grids that indicate the spatial coarseness, convolutional layers are represented as vertical lines, fully convolutional layers as rectangles, and the arrow represents the skips

detailed segmentations. It combines predictions from the final layer, the pool4 layer, and the pool3 layer to make stride 8 predictions as shown in Fig. 2.

3.3 SegNet

SegNet [14] is illustrated in Fig. 3 and consists of an encoder network, a decoder network, and a soft-max pixelwise classification layer. The encoder network comprises the first 13 convolutional layers and the 5 pooling layers of the VGG16 [18] network. The FC layers are discarded to retain higher resolution feature maps. Since each encoder layer has a corresponding decoder layer, the decoder network also has 13 convolutional layers and 5 upsampling layers. Max pooling indices are stored for each encoder and are used to upsample the feature maps in each decoder.

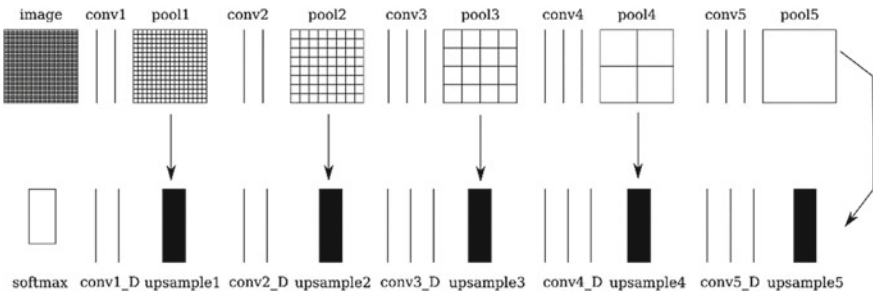


Fig. 3 SegNet Architecture. The upper network is the encoder and the bottom network is the decoder. The decoder uses the pooling indices as indicated by the arrows, to upsample its input. Filled rectangles represent upsampling, vertical lines represent convolution, and grids represent pooling

4 Methodology

The CNN architectures are trained using each dataset, with pre-trained VGG16 weights [18] and parameters according to [13]. The networks are trained ignoring the void class, except for SegNet, which is trained with the void class and without it. Each ConvNet is tested with the corresponding validation dataset, and the metrics are calculated. We compute metrics commonly used in semantic segmentation and scene parsing [13]: true positive rate, pixel accuracy, mean accuracy, mean IU, frequency weighted IU (see Eqs. 1–5),

$$n_{ii} / \left(\sum_j n_{ji} - n_{ii} \right), \quad (1)$$

$$\sum_i n_{ii} / \sum_i t_i, \quad (2)$$

$$(1/n_{cl}) \sum_i n_{ii} / t_i, \quad (3)$$

$$(1/n_{cl}) \sum_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii} \right), \quad (4)$$

$$\left(\sum_k t_k \right)^{-1} \sum_i t_i n_{ii} / \left(t_i + \sum_j n_{ji} - n_{ii} \right) \quad (5)$$

where n_{ii} is the number of true positives for class i , n_{ji} is the number of pixels of class j predicted that belong to class i , n_{cl} is the total number of classes, and t_i is the total number of pixels of class i .

Hereafter, we obtain the Pareto fronts from five different sets of objectives. These different fronts consider global (all the classes), local (road, sidewalk, car, traffic sign, called *relevant classes*), and mixed classes metrics. The first front considers all the metrics mentioned above. The second front considers the four-individual metrics of the relevant classes. The third front considers the individual metrics, the local mean accuracy, the local and global mean IU, and the frequency weighted IU. The fourth front considers all the metrics except the individual metrics for the relevant classes. And the fifth fronts consist of the local mean accuracy, the local and global mean IU and the frequency weighted IU.

Finally, the obtained solutions in the Pareto fronts are scored using the proposed method. For each solution in the first Pareto front, the ratio between the amount of dominating solutions and dominated solutions is calculated. The higher ratios within a difference of 0.1 between them are given a score of 3, the next ranking elements a score of 2, the third ranking elements of the first front a score of 1, and the rest of the solutions of the first front (if any) a score of 0. If the elements of the first front are

exhausted, the scoring system proceeds to the second front, and so on. The process is repeated for the five Pareto fronts evaluations. The same approach is used to evaluate the average forward pass.

After obtaining the points for each solution, a level is assigned. The A ranked solutions have the best performance, the B ranked solutions have average performance, and the C ranked solutions are the less efficient solutions.

5 Experiments and Results

This section presents details about the experimental settings, including the employed datasets and parametric configuration. Likewise, the experimental results are reported in terms of the metrics and Pareto fronts mentioned in Sect. 4. Finally, the solutions ranked according to the proposed scoring systems are analyzed.

5.1 Experimental Setting

We propose a method that evaluates and ranks the performance of FCN-Alexnet, FCN-8s, and SegNet architectures for semantic segmentation using three urban scenes datasets. All the models run on an NVIDIA Titan X of 12 GB of memory. FCN-Alexnet and FCN-8s are trained on DIGITS, and SegNet is trained on its Caffe implementation.

5.1.1 Datasets

We used three popular semantic segmentation datasets of urban scenes that have very different characteristics between them: Cambridge-driving Labeled Video Database (CamVid) [10], Cityscapes [11], and SYNTHetic collection of Imagery and Annotations (SYNTHIA) [12].

CamVid is a ground truth database with 32 semantic classes. It comprises of 600 RGB images with 480×360 pixel resolution. For this study, only 12 classes were used: sky, building, pole, road, pavement, tree, sign symbol, fence, car, pedestrian, bicyclist, and unlabeled.

Cityscapes contains 5000 images with a resolution of 2048×1024 pixels. The images were taken from many different cities from spring to fall. Only 20 of the 30 classes were used: void, road, sidewalk, building, fence, wall, pole, traffic sign, traffic light, vegetation, terrain, sky, person, rider, car, truck, bus, train, motorcycle, and bicycle.

SYNTHIA is a synthetic dataset consisting of photo-realistic frames of a virtual city with semantic annotations for 13 classes: void, sky, building, road, sidewalk,

Table 1 Training parameters

Architecture	Dataset	Training epochs	Solver type	Learning rate	Step size	Gamma
FCN-Alexnet	CamVid	10	SGD	0.0001	33	0.1
	Cityscapes	50	SGD	0.0001	25	0.1
	SYNTIA	10	SGD	0.0001	33	0.1
FCN-8s	CamVid	100	SGD	0.00001	25	0.1
	Cityscapes	30	SGD	0.0001	25	0.1
	SYNTIA	10	SGD	0.00001	25	0.1
Segnet	CamVid	200	SGD	0.01	40	0.1
	Cityscapes	50	SGD	0.01	33.6	0.1
	SYNTIA	5	SGD	0.01	33.5	0.1

fence, vegetation, pole, car, sign, pedestrian, cyclist, and marking. The frames simulate the four seasons and different lighting conditions. It contains 13407 images with a 960×720 pixel resolution.

To speed-up the training process and employ less memory, the resolution of SYNTIA and Cityscapes is downsampled to 360×480 pixels and 256×512 pixels respectively. The RGB label images are also converted to one-channel images where the value of each pixel is equal to the class to which it belongs.

5.1.2 Parametric Configuration

As mentioned above, the networks were trained using each dataset, with pre-trained VGG16 weights [18] and parameters according to Table 1.

5.2 Experimental Results

The architectures were tested with the validation dataset of the corresponding dataset and the metrics were calculated by comparing the results of the tests and the one-channel label images.

We defined a solution as a combination of an architecture and a dataset. Figure 4 displays an example of the test results of each solution. Initially, a confusion matrix was computed for obtaining the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for each class and in total. With this information, the metric results were obtained and are shown in Tables 2, 3. The inference times were also calculated for each solution (see Table 4). In the scoring system, we only considered the average forward pass.

It is possible to observe that FCN-Alexnet had difficulties predicting small classes, such as pole, traffic sign, and even pedestrian. On average, FCN-8s has a higher

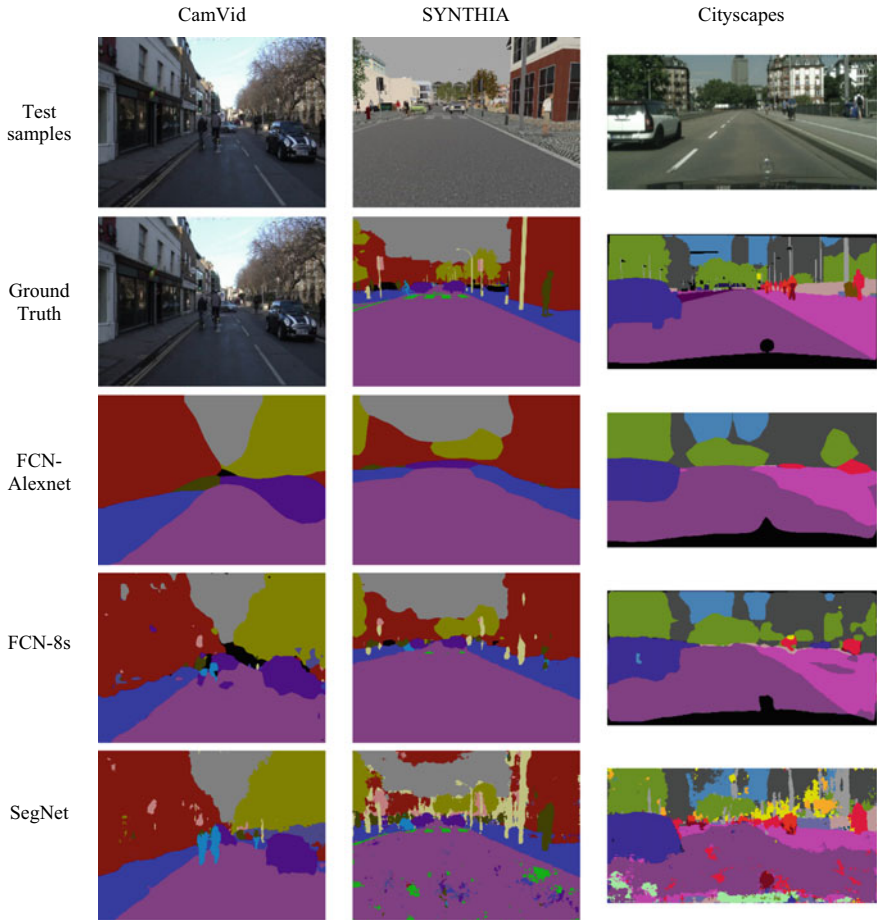


Fig. 4 Representative segmentation results of FCN-Alex, FCN-8s, and SegNet architectures for CamVid, SYNTHIA, and Cityscapes datasets

prediction rate if we combine the outputs of all datasets for the relevant classes (except for the sign class in Cityscapes due to the large number of classes). Although a qualitative analysis of the inferences obtained by SegNet seems to favor its outputs, especially for the small classes, the results show a higher appearance of FN.

The Pareto fronts were calculated as explained in Sect. 4, see Table 5. For the first, second, and third Pareto fronts the individual metrics play an important role. The fourth and fifth fronts depend only on global metrics.

The solutions are ranked as follows:

- Score A: SFIII, SSI.
- Score B: SAIII, SSIII, SSV.
- Score C: SAI, SAIL, SFI, SFII, SSII, SSIV.

Table 2 Individual metrics

Solution	Architecture	Dataset	TPR road	TPR sidewalk	TPR car	TPR sign
SAI	FCN-Alexnet	CamVid	0.957052	0.67154404	0.65232974	0.03933341
SAII		Cityscapes	0.95225934	0.68282967	0.8601143	0.05174555
SAIII		SYNTHIA	0.95620835	0.92779717	0.90909551	0.15047869
SFI	FCN-8s	CamVid	0.95088566	0.72559723	0.87353355	0.32568248
SFII		Cityscapes	0.95478848	0.64252052	0.82779516	0.05435011
SFIII		SYNTHIA	0.96470307	0.94415263	0.92839816	0.4918346
SSI	Segnet	CamVid	0.96703423	0.9113198	0.85756261	0.53535609
SSII		Cityscapes	0.84535636	0.67406984	0.77348856	0.69924832
SSIII		SYNTHIA	0.90503844	0.73730198	0.69358013	0.84313699
SSIV		SYNTHIA void	0.95718419	0.81955991	0.69059726	0.48203103
SSV		Cityscapes void	0.94303683	0.76036994	0.87674601	0.51598683

Table 3 Global and individual metrics

Solution	Pixel acc global	Pixel acc local	Mean acc global	Mean acc local	Mean IU global	Mean IU local	Frequency weighted IU
SAI	0.81897	0.81034	0.57900	0.67037	0.40038	0.46532	0.69988
SAII	0.84380	0.89872	0.56941	0.71848	0.34612	0.54309	0.74118
SAIII	0.90659	0.92724	0.73291	0.81087	0.57550	0.67681	0.83493
SFI	0.85258	0.83876	0.65931	0.68330	0.47785	0.55917	0.75344
SFII	0.83809	0.87948	0.41702	0.70000	0.25792	0.51595	0.72949
SFIII	0.91576	0.92638	0.76682	0.84154	0.61913	0.74719	0.85047
SSI	0.90883	0.93058	0.78166	0.83431	0.67029	0.72712	0.84081
SSII	0.70165	0.82683	0.35985	0.60155	0.27138	0.50233	0.60610
SSIII	0.74915	0.81133	0.54013	0.62330	0.44369	0.51731	0.63763
SSIV	0.85170	0.85763	0.65722	0.70489	0.53141	0.59391	0.75799
SSV	0.81768	0.88998	0.44306	0.68313	0.33548	0.60255	0.72588

The experiments show that the best combinations of nets and datasets were SFIII, and SSI. One solution used SYNTHIA (SFIII), and another used CamVid (SSI), each one with a different net architecture. The second-best combinations were SAIII, SSII, and SSV. Most of the experiments that included the use of the Cityscapes (SAII, SFII, SSII) scored the third tier, mainly due to a low value of mean IU, and low scores in the sensitivity of sidewalk and sign classes. The highest values of mean IU (higher than 0.60 for global, and higher than 0.7 for local), and the highest values for pixel accuracy (higher than 0.9) were obtained by A tier combinations.

Table 4 Inference times

Solution	Time in ms			
	Average forward pass	Average backward pass	Average forward-backward	Total time
SAI	11.8344	8.3525	20.2299	1011.5
SAII	10.7678	7.45536	18.2685	913.423
SAIII	11.6876	8.22718	19.9571	997.854
SFI	75.4818	87.8937	163.424	8171.21
SFII	67.4294	72.7735	140.248	7012.41
SFIII	75.6433	88.092	163.801	8190.06
SSI	71.3896	91.9225	163.386	8169.3
SSII	76.5265	96.2096	172.811	8640.54
SSIII	100.156	129.736	229.958	11497.9
SSIV	100.156	129.736	229.958	11497.9
SSV	76.5265	96.2096	172.811	8640.54

Table 5 Pareto fronts obtained

PF 1	PF 2	PF 3	PF 4	PF 5
SAIII	SFIII	SFIII	SFIII	SFIII
SFIII	SSI	SSI	SSI	SSI
SSI	SSII	SSII		
SSII	SSIII	SSIII		
SSIII	SSV	SSV		
SSV				

6 Conclusions

The analysis of the proposed methods shows that the current deep learning nets for semantic segmentation autonomous vehicles scenarios have variant accuracy, depending on the database and the difficulty of the classes in the databases. In semantic segmentation for autonomous vehicles, the amount and the type of classes to segment by the system is critical. Although the results show that recent architectures can segment a high number of classes with high accuracy, small classes like traffic signs and fences are very hard to detect in the databases of real scenarios. For this type problems, synthetic databases, such as SYNTHIA, can enhance the correct segmentation of small classes or those that are near classes that have a high number of pixels in the images (such as the roads and buildings).

Further work would include the enhancement of the detection results of the architectures used; the exploration of more state of the art architectures; the mix of synthetic databases with real scenarios databases in the training and validation stages; and real-time implementations and evaluations.

References

1. Woensel, L., Archer, G.: Ten Technologies Which Could Change Our Lives. EPRS—European Parliamentary Research Services (2015)
2. Kukkala, V., Tunnell, J., Pasricha, S., et al.: Advanced driver-assistance systems: a path toward autonomous vehicles. *IEEE Consum. Electron. Mag.* **7**(5), 18–25 (2018)
3. Zhang, X., Chen, Z., Wu, Q., et al.: Fast semantic segmentation for scene perception. *IEEE Trans. Industr. Inf.* **15**(2), 1183–1192 (2019)
4. Zhang, Y., Chen, H., He, Y., et al.: Road segmentation for all-day outdoor robot navigation. *Neurocomputing* **314**, 316–325 (2018)
5. Çiçek, Ö., Abdulkadir, A., Lienkamp, S. et al.: 3D U-Net: learning dense volumetric segmentation from sparse annotation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, 424–432 (2016)
6. Jiang, F., Grigorev, A., Rho, S., et al.: Medical image semantic segmentation based on deep learning. *NeuralComput. Appl.* **29**(5), 1257–1265 (2018)
7. Kemker, R., Salvaggio, C., Kanan, C.: Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. *ISPRS J. Photogramm. Remote Sens.* (2018)
8. Oberweger, M., Wohlhart, P., Lepetit, V.: Hands deep in deep learning for hand pose estimation. [arXiv:1502.06807](https://arxiv.org/abs/1502.06807) (2015)
9. Geiger, A., Lenz, P., Stiller, C., et al.: Vision meets robotics: the KITTI dataset. In: *Int. J. Robot. Res.* **32**(11), 1231 (2013)
10. Brostow, G., Fauqueur, J., Cipolla, R.: Semantic object classes in video: a high-definition ground truth database. *Pattern Recognit. Lett.* **30**(2), 88–97 (2009)
11. Cordts, M., Omran, M., Ramos, S., et al.: The Cityscapes dataset for semantic urban scene understanding. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016)
12. Ros, G., Sellart, L., Materzynska, J., et al.: The SYNTHIA dataset: a large collection of synthetic images for semantic segmentation of urban scenes. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4321–4330 (2016)
13. Shelhamer, E., Long, J., Darrell, T.: Fully convolutional networks for semantic segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 640–651 (2017)
14. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: a deep convolutional encoder-decoder architecture for image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **39**, 2481–2495 (2017)
15. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp 1520–1528 (2015)
16. Chen, B., Gong, C., Yang, J.: Importance-aware semantic segmentation for autonomous vehicles. *IEEE Trans. Intell. Transp. Syst.* **20**(1), 137–148 (2018)
17. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: *Proceedings of the Neural Information Processing Systems*, pp 1106–1114 (2012)
18. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: *Proceedings of the International Conference Learning Representation* (2015)
19. Szegedy, C., Wei, L., Jia, Y., et al.: Going deeper with convolutions. In: *Proceedings of the Computer Vision Pattern Recognition*, pp. 1–9 (2015)

20. Girshick, R., Donahue, J., Darrell, T., et al.: Region-based convolutional networks for accurate object detection and segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(1), 142–158 (2015)
21. Sermanet, P., Eigen, D., Zhang, X., et al.: OverFeat: integrated recognition, localization and detection using convolutional networks. [arXiv:1312.6229](https://arxiv.org/abs/1312.6229) (2013)
22. Farabet, C., Couprie, C., Najman, L., et al.: Learning hierarchical features for scene labeling. *IEEE Trans. Pattern Anal. Mach. Intell.* **35**(8), 1915–1929 (2013)
23. Chen, L., Papandreou, G., Kokkinos, I., et al.: Semantic image segmentation with deep convolutional nets and fully connected crfs. [arXiv:1412.7062](https://arxiv.org/abs/1412.7062) (2014)
24. Siam, M., Elkerdawy, S., Jagersand, M., et al.: Deep semantic segmentation for automated driving: taxonomy, roadmap and challenges. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), pp 1–8 (2017)
25. Kendall, A., Badrinarayanan, V., Cipolla, R.: Bayesian segnet: model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *CoRR abs/1511.02680* (2015)
26. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-Assisted Intervention, pp. 234–241 (2015)
27. Siam, M., Gamal, M., Abdel-Razek, M., et al.: A comparative study of real-time semantic segmentation for autonomous driving. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp 700–710. Salt Lake City, UT (2018)

Towards Tracking Trajectory of Planar Quadrotor Models



Prometeo Cortés-Antonio, Fevrier Valdez, Oscar Castillo and Patricia Melin

Abstract The paper introduces the basis to control a simple planar quadrotor model in the tracking trajectory problem. The dynamics of the model is developed and a control system is designed and implemented to tracking two different trajectories without obstacles. General control system contains a PD controller to drive altitude (motion in z direction), and a cascade control scheme to drive y position by controlling orientation of roll angle. Results of the error position and command variables on two different trajectories are analyzed.

Keywords Tracking trajectory · PD and cascade control · Quadrotor

1 Introduction

Nowadays, several applications with Unmanned Autonomous Vehicle (UAV) can be found in the literature, for example, applications for aerial video, surveillance, packages delivering, etc. The cost of use an UAV can be expensive and have several limitations as the battery life, air resistance, weight of drone, number of rotors, accuracy of GPS, etc. For this reason, many researchers try to improve some features such as, the cost of sensors, actuators, and the development of electronic devices to these UAVs of good quality and the best cost [1, 2]. Several approaches have been proposed with tracking trajectories, in [3] the authors proposed the emulation of a physical standard decentralized control of a multi-agent system composed of several

P. Cortés-Antonio (✉) · F. Valdez · O. Castillo · P. Melin
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: prometeo.cortes@tectijuana.edu.mx

F. Valdez
e-mail: fevrier@tectijuana.mx

O. Castillo
e-mail: ocastillo@tectijuana.mx

P. Melin
e-mail: pmelin@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_24

differentially driven mobile robots. Also, in [4] was proposed an approach with three trajectory tracking control strategies for unicycle-type robots based on a leader-followers scheme. Also, to improve the performance achieved with the traditional control systems some researchers use fuzzy systems for parameter adaptation to achieve the best performance in the control system, for example, in [5, 6] the authors proposed a dynamic fuzzy logic parameter tuning for Ant Colony Optimization (ACO) [7] of an autonomous mobile robot and the Traveling Salesman Problem (TSP) [8, 9]. By other hand, in [10] was proposed a Bat Algorithm [11, 12] in optimizing the trajectory of a unicycle mobile robot, based on two wheels mounted on the same axis and a front wheel and the algorithm is responsible for building the best Type-1 fuzzy system once selected the best applied to the mobile robot model with the objective of following an established path with the least margin of error. As can be seen, the use of metaheuristics methods is very popular in the last years to optimize some important parameters, optimize the best path, as in [13] the authors developed a Cuckoo search algorithm [14] with bat algorithm to find the optimal path planning of mobile robot in the unknown or partially known environment. However, our approach is different because the main idea in this research is develop a control model to make autonomous flight with tracking trajectories in the space without obstacles of quadrotors using a PD controller. In the future, this work can be extended with optimization of important parameters to achieve the best performance. The structure of the paper is as follows: in Sect. 2, it is introduced the basis to model the dynamics of quadrotor; Sect. 3 presents the design and implementation of the control system for tracking trajectory; the results of the controller model for tracking of two different trajectories are presented in Sect. 4. Finally, Sect. 5 states the conclusion and future works.

2 Dynamics of Planar Quadrotor

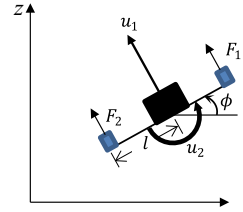
In general, a quadrotor consists of four independently controlled rotors. A very simple model cannot consider the effects due to flapping on hinges or blades. In a quadrotor there are a pair of rotors spinning in opposite direction to the other one to avoid a constant rotation. Through the angular velocity of a rotor, a thrust force is generated. The four thrust forces of a quadrotor point upward of it. Using different magnitudes of these forces, it is possible to control the position and orientation of the quadrotor, that is, to drive its six degree of freedom.

The position of a quadrotor can be specified by its translational motions over x , y , z coordinates and its orientation by three angular motions called roll, pitch and yaw to describe rotational motions on x , y and z axis.

In this paper, we reduce the complexity of the problem to a planar quadrotor model, specifically, to control the quadrotor in y - z plane. Figure 1 shows the parameters and dynamics of planar quadrotor model. The planar model considers only two motors, which generate thrust forces F_1 and F_2 .

Newton equations describe the translational and rotational motion as follows

Fig. 1 Dynamics of a planar quadrotor model



$$\ddot{y} = -\frac{u_1 \sin(\phi)}{m} \quad (1)$$

$$\ddot{z} = \frac{u_1 \cos(\phi)}{m} - g \quad (2)$$

Such that

$$u_1 = F_1 + F_2 \quad (4)$$

$$u_2 = (F_1 - F_2)l \quad (5)$$

where

m is mass of the quadrotor

l is length from center of mass of robot to location of the rotor.

g is the gravity constant

I_{xx} is inertia moment over x axis

u_1 is sum of thrust forces

u_2 is attitude control variable

$\ddot{\phi}$ is rotational acceleration or roll acceleration of the quadrotor

\ddot{y} and \ddot{z} are translational acceleration on y and z axis.

The newton equations can be linearized in equilibrium or hover configuration. In hover state, the quadrotor has the following values: $y_e = z_e = \phi_e = u_{2,e} = 0$ and $u_{1,e} = mg$. The linearized planar quadrotor model is described as follows:

$$\ddot{y} = g\phi \quad (6)$$

$$\ddot{z} = \frac{u_1}{m} - g \quad (7)$$

$$\ddot{\phi} = \frac{u_2}{I_{xx}} \quad (8)$$

The linearized plant will be controller as is described in next section.

3 Design and Implementation of Control System

Figure 2 shows the general feedforward control scheme to control a planar quadrotor model. x_d is the desired position with y_d and z_d components. $x = [y, z, \phi]$ are positions and orientation states of the quadrotor, and u defines the command variables u_1 and u_2 of the plant.

The controller block contains three embedded controllers, a PD controller to control the altitude z of the quadrotor. Whereas a cascade control configuration, with two embedded PD controllers, to control y position.

Altitude is controlled directly by driving the control variable u_1 , using error and derivative of error of the desired height and the height of the robot. The command variable u_1 is computed by a PD control as follows:

$$u_1 = k_{v,z}\dot{e}_z + k_{p,z}e_z \tag{9}$$

To control y position, it is necessary to drive u_2 by assignation of a command angle ϕ_c as a function of error in y position. The set of equations to calculate u_2 is as follows:

$$\phi_c = k_{v,y}\dot{e}_y + k_{p,y}e_y \tag{10}$$

$$u_2 = k_{v,\phi}\dot{e}_\phi + k_{p,\phi}e_\phi \tag{11}$$

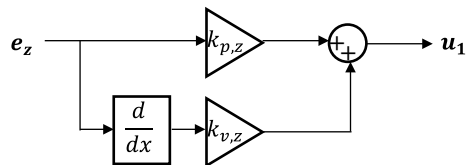
The embedded control schemes of the controller block shown in Fig. 2 are shown in Figs. 3 and 4.

In Fig. 3, u_1 is composed by a proportional term plus a derivative term of the e_z error. Whereas in Fig. 4, a cascade scheme is designed for controlling u_2 command variable. First ϕ_c is estimated using error of y position, after that, u_2 is computed using error of ϕ_c and actual angle ϕ of the robot.

Fig. 2 General feedback control scheme of a quadrotor



Fig. 3 PD control scheme for controlling altitude of the quadrotor



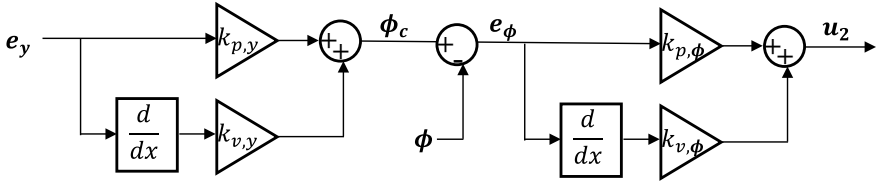


Fig. 4 Cascade control scheme for controlling attitude of the quadrotor

The unique constraints put on the command variables are: (a) $0 < u_1 < 7$ and (b) $-7 < u_1 < 7$.

4 Experimental Results

In this section, we present simulations of the designed controller system on two trajectories. The planar quadrotor model under consideration has the physical parameters values showed in Table 1.

The parameter values of the control system designed in Sect. 3 were tuning by trial and error. The final values, which give best results in the two trajectories are listed in Table 2.

The desired trajectories studied in this paper are defined by the next equations. For simulation 1, the trajectory is as follows:

$$y_d = 2 \cos(t) + 4 \cos(0.5t) - 2 \tag{12}$$

Table 1 Physical parameters of quadrotor planar model

Parameter	Value
m	0.2 kg
l	0.15 m
I_{xx}	0.1 kg m ²
g	9.81 m/s ²

Table 2 Control parameter values

Parameter	Value
$k_{p,z}$	50
$k_{v,z}$	3
$k_{p,y}$	0.8
$k_{v,y}$	1
$k_{p,\phi}$	20
$k_{v,\phi}$	10

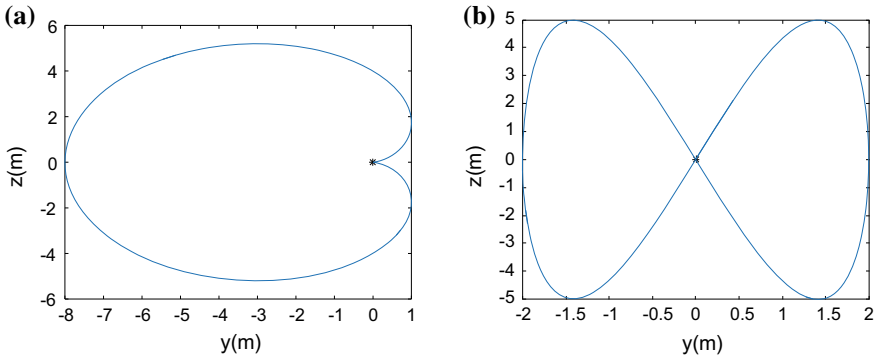


Fig. 5 Desired trajectories for **a** simulation 1 and **b** simulation 2

$$z_d = 2 \sin(t) - 4 \sin(0.5t) \tag{13}$$

Whereas for simulation 2, the trajectory is given as follows:

$$z_d = 5 * \sin(t) \tag{14}$$

$$y_d = 2 * \sin(0.5 * t) \tag{15}$$

Figure 5 depicts the shapes of the trajectories of above equations in $y - z$ plane. The execution time to complete a cycle in both trajectories is $4 * \pi$ (approximately 12.57) seconds. In simulation 1, the desired trajectory is into a region of $[-8,1]$ meters in y direction and $[-5,5]$ meters in z direction. The second trajectory is into a region of $[-2,2]$ meters in y direction and $[-5,5]$ meters in z direction. The symbol * indicates the initial and final position of the trajectory.

Trajectory 1

Figure 6 shows the target versus actual trajectories of the quadrotor. In Fig. 6a, b are shown the displacements in y direction and z direction during a time of 8π seconds. Figure 6c shows shape curve in a $y-z$ plane of the trajectory. Additionally, Fig. 6d depicts the roll angle of the robot during the test.

The command variables assigned by the control system to follow the trajectory 1 are shown in Fig. 7. It can see u_1 has only positive magnitudes and it is into $[0,2.6]$ range of values. It can also see that u_2 is into $[-1.5,4.3]$ range of values.

Finally, Fig. 8 shows the error curves in y and z positions. It can observe maximum error in y position is less than 27 cm, whereas in z , it is less than 5 cm. However, it can observe that in some parts of path (peaks on curve), the robot had difficulties in follow the desired trajectory.

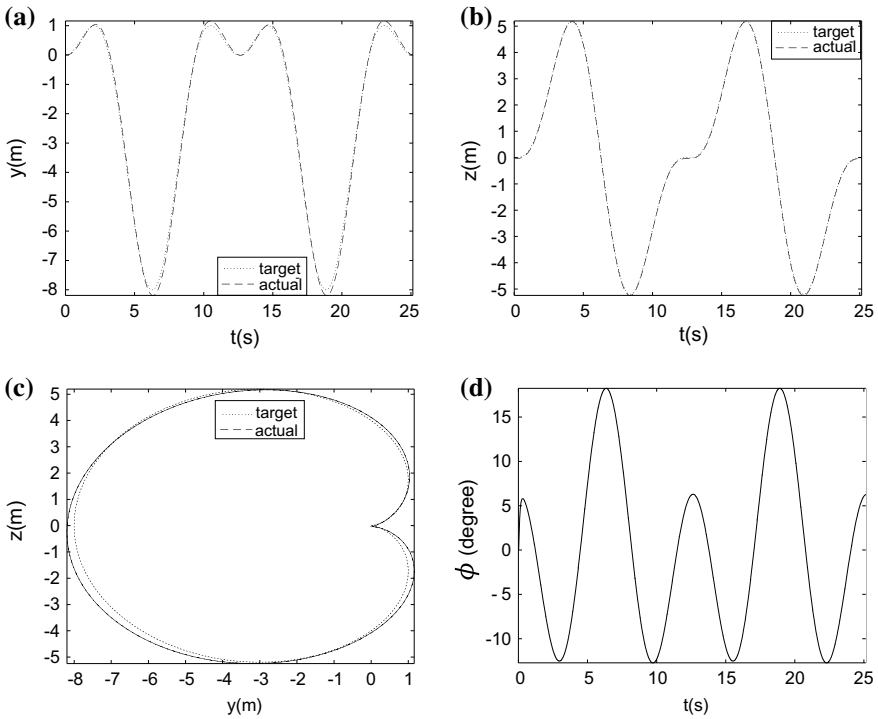


Fig. 6 Target and actual positions of the control system in trajectory 1

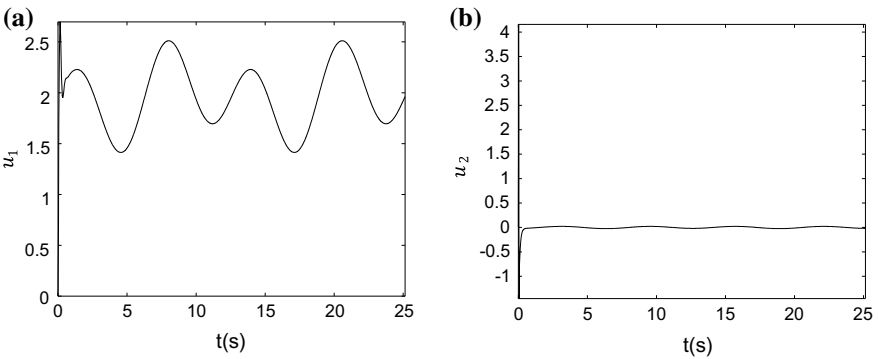


Fig. 7 Command variables in trajectory 1

Trajectory 2

Figure 9 shows the target versus actual trajectories of the quadrotor for trajectory 2. Picture shows the displacements in y and z directions in the time, the trajectory formed in the $y-z$ plane and roll angle of the robot during the test. The execution time

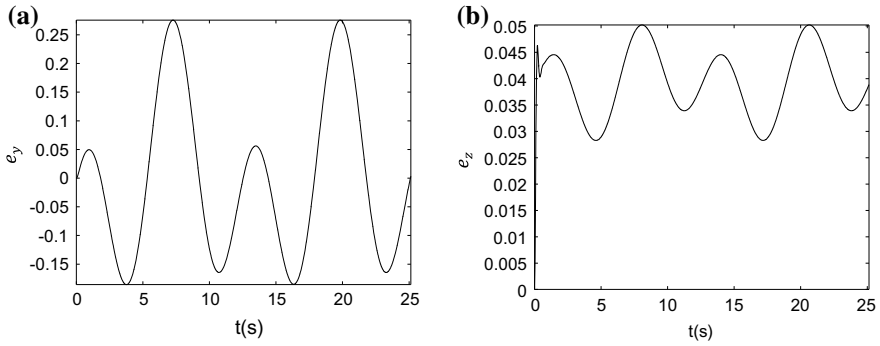


Fig. 8 Error curves in simulation 1

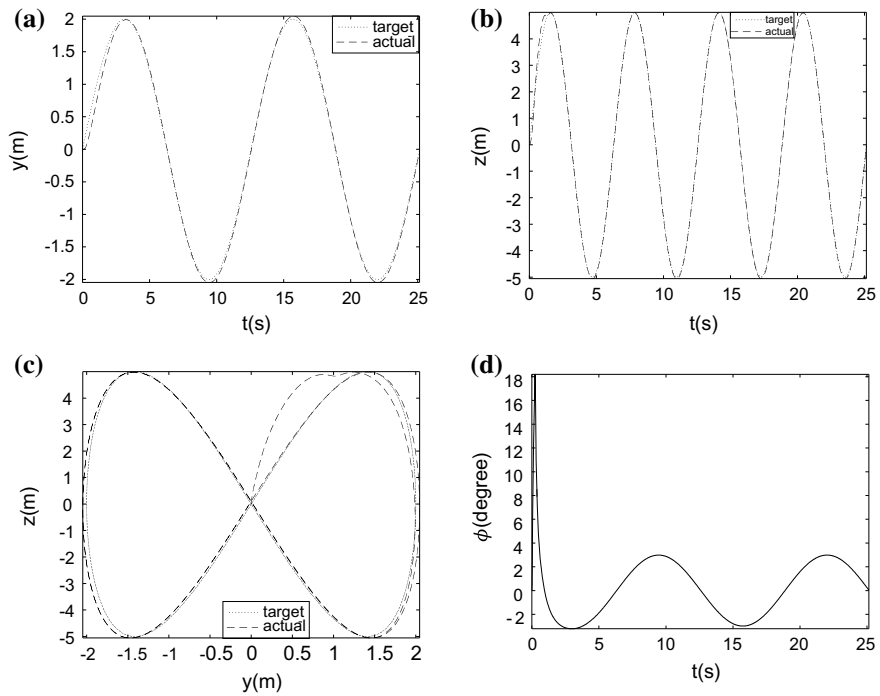


Fig. 9 Target and actual positions for the control system in trajectory 2

is also 8π seconds. In Fig. 9c, it is more evident how the quadrotor had problems at the beginning of the path, but in second round the problem disappeared.

The command variables assigned by the control system to follow the trajectory 2 are shown in Fig. 10. It can see, u_1 has positive magnitudes into $[0, 7]$ range of values. It can also see that u_2 is into $[-7, 7]$ range of values. So, we can state that trajectory

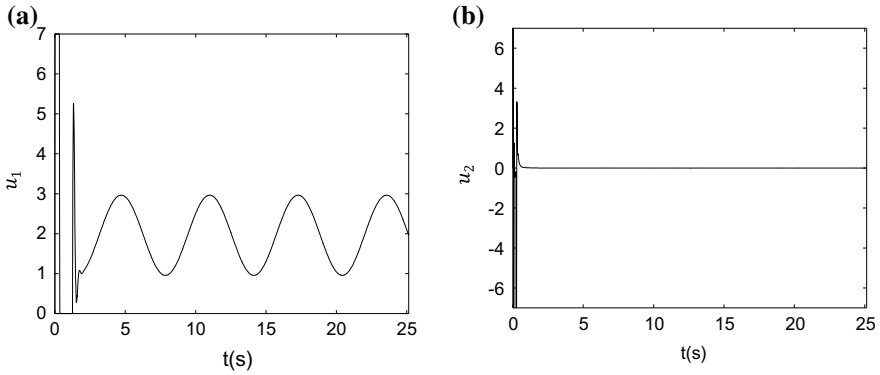


Fig. 10 Command variables in trajectory 2

2 was more complicate to follow than trajectory 1 and the quadrotor spends more energy.

Figure 11 shows the error curves in y and z positions for trajectory 2. It can observe maximum error in y is less than 28 cm, whereas in z , it is less than 50 cm. However, as it was pointed out, the biggest error occurred at startup of the system.

Table 3 lists five different metrics of errors, that can be used in future works to performance comparison with other control designs. Metric presented are: (a) root mean square error (RMSE); (b) integral square error (ISE); (c) integral of absolute

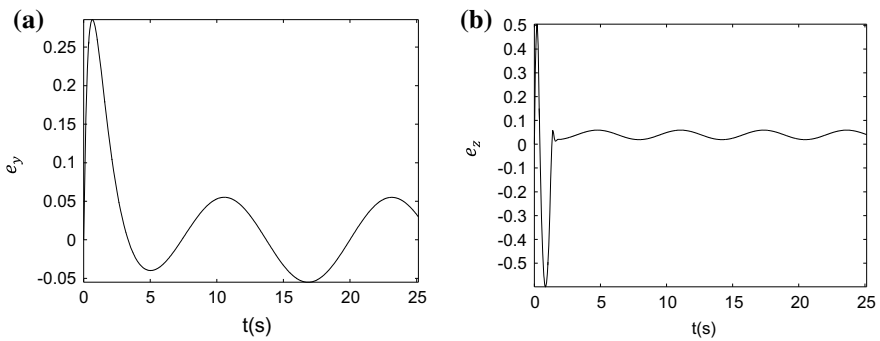


Fig. 11 Error curves in simulation 2

Table 3 Different metrics errors for trajectories

Simulation	RMSE	ISE	IAE	ITSE	ITAE
Trajectory 1	0.15	0.54	3.94	7.10	51.19
Trajectory 2	0.12	0.40	2.68	1.34	25.04

error (IAE); (d) integral of time multiply square error (ISE); and (e) integral of time multiply absolute error (IAE).

5 Conclusions and Future Works

In this paper, we designed a control system composed by three embedded PD controllers. A PD controller was implemented for estimating the z position, whereas, a cascade control configuration of two PD controller was implemented for estimating the y position of the robot. The controller parameters were tuning by trial and error until we obtained a good configuration that could follow two desired trajectories. With the simulations presented, it was shown that the proposed control design obtained good performance with small errors to follow two different trajectories.

One important observation, is that the command variables considered u_1 and u_1 are considered as independent variables, however these variables in a real system are depended of thrust forces F_1 and F_2 , for this reason, in future works, we will work with a control scheme for F_1 and F_2 inputs of the plant using intelligent methods to tune both the controller parameters and F_1 and F_2 values. For example, a Genetic Algorithm [15, 16], Gravitational Search Algorithm [17, 18], Particle Swarm Optimization [19, 20], Differential Evolution [21, 22], Harmony Search Algorithm [23, 24] are optimization methods which help us to add dynamically parameters to the control system.

Acknowledgements We would like thankful to Tijuana Institute of Technology and CONACYT by their support to this research work.

References

1. Lim, H., Park, J., Lee, D., Kim, H.J.: Build your own quadrotor. *IEEE Robot. Autom. Mag.* **19**(3), 33–45 (2012)
2. Ha, Q., Deville, U., Pham, Q., Hà, M.: On the min-cost traveling salesman problem with drone. *Trans. Res. Part C: Emerg. Technol.* **86**(2018), 597–621 (2018)
3. González-Sierra, J., Aranda-Bricaire, E., Hernández-Mendoza, D.E., Santiaguillo-Salinas, J.: Emulation of n-trailer systems through differentially driven multi-agent systems: continuous- and discrete- time approaches. *J. Intell. Rob. Syst.* **75**(1), 129–146 (2014)
4. González-Sierra, J., Aranda-Bricaire, E., Hernández-Martínez, E.G.: Robotic systems formation tracking with orientation convergence for groups of unicycles. *Regular Paper*
5. Castillo, O., Neyoy, H., Soria, J., García, M., Valdez, F.: Dynamic fuzzy logic parameter tuning for ACO and its application in the fuzzy logic control of an autonomous mobile robot. *Int. J. Adv. Robotic Syst.* **10**(1), 51 (2013)
6. Neyoy, H., Castillo, O., Soria, J.: Dynamic fuzzy logic parameter tuning for ACO and its application in TSP problems. In: Castillo, O., Melin, P., Kacprzyk, J. (eds.) *Recent Advances on Hybrid Intelligent Systems. Studies in Computational Intelligence*, vol. 451. Springer, Berlin, Heidelberg (2013)

7. Olivas, F., Valdez, F., Castillo, O.: Fuzzy system for parameter adaptation in ant colony optimization. In: 2014 IEEE Symposium on Swarm Intelligence (SIS), pp. 84–89 (2014)
8. Reinelt, G.: TSP_LIB-a traveling Salesman Problem Library. *ORSA J. Comput.* **3**, 376–384 (1991)
9. Perez, J., Melin, P., Castillo, O., Valdez, F., Gonzalez, C., Martinez, G.: Trajectory optimization for an autonomous mobile robot using the bat algorithm. In: Melin, P., Castillo, O., Kacprzyk, J., Reformat, M., Melek, W. (eds.) *Fuzzy Logic in Intelligent System Design. NAFIPS 2017. Advances in Intelligent Systems and Computing*, vol. 648. Springer, Cham (2018)
10. Wang, Z., Guo, J., Zheng, M., Wang, Y.: Uncertain multiobjective traveling salesman problem. *Eur. J. Oper. Res.* **241**(2), 478–489 (2015). ISSN 0377-221
11. Yang, X.-S.: A new metaheuristic bat-inspired algorithm. In: Gonzalez, J. R., et al. (eds.) *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, vol. 284, pp. 65–74 (2010)
12. Pérez, J., Valdez, F., Castillo, O., Roeva, O.: Bat algorithm with parameter adaptation using Interval Type-2 fuzzy logic for benchmark mathematical functions. In: 2016 IEEE 8th International Conference on Intelligent Systems (IS), pp. 120–127 (2016)
13. Saraswathi, M., Bala, G., Murali, B., Deepak, B.: Optimal path planning of mobile robot using hybrid cuckoo search-bat algorithm. *Procedia Comput. Sci.* **133**, 510–517 (2018). ISSN 1877-0509
14. Mareli, M., Twala, B.: An adaptive Cuckoo search algorithm for optimization. *Appl. Comput. Inf.* **14**(2), 107–115 (2018). ISSN 2210-8327
15. Machado, A., et al.: Real time pathfinding with genetic algorithm. In: 2011 Brazilian Symposium on Games and Digital Entertainment, pp. 215–221. Salvador (2011)
16. Haupt, R., Haupt, S.: *Practical Genetic Algorithms 2nd edn.* A Wiley-Interscience Publication (2004)
17. Olivas, F., Valdez, F., Melin, P., Sombra, A., Castillo, O.: Interval type-2 fuzzy logic for dynamic parameter adaptation in a modified gravitational search algorithm. *Inf. Sci.* **476**, 159–175 (2019). ISSN 0020-0255
18. González, B., Valdez, F., Melin, P., Prado-Arechiga, G.: Fuzzy logic in the gravitational search algorithm enhanced using fuzzy logic with dynamic alpha parameter value adaptation for the optimization of modular neural networks in echocardiogram recognition. *Appl. Soft Comput.* **37**, 245–254 (2015)
19. Olivas, F., Valdez, F., Castillo, O., et al.: Dynamic parameter adaptation in particle swarm optimization using interval type-2 fuzzy logic. *Soft. Comput.* **20**, 1057 (2016). <https://doi.org/10.1007/s00500-014-1567-3>
20. Saska, M., Macas, M., Preucil, L., Lhotska, L.: Robot path planning using particle swarm optimization of Ferguson splines. In: 2006 IEEE Conference on Emerging Technologies and Factory Automation, Prague, pp. 833–839 (2006)
21. Zhang, X., Chen, J., Xin, B., Fang, H.: Online path planning for UAV using an improved differential evolution algorithm. In: *IFAC Proceedings*, vol. 44, Issue No. 1, pp. 6349–6354 (2011). ISSN 1474-6670
22. Zamuda, H., Hernández, J.: Success history applied to expert system for underwater glider path planning using differential evolution. *Expert Syst. Appl.* **119**, 155–170 (2019). ISSN 0957-4174
23. Peraza, C., Valdez, F., Castillo, O.: Study on the use of Type-1 and Interval Type-2 fuzzy systems applied to benchmark functions using the fuzzy harmony search algorithm. In: Melin, P., Castillo, O., Kacprzyk, J., Reformat, M., Melek, W. (eds.) *Fuzzy Logic in Intelligent System Design*, vol. 648, pp. 94–103. Springer International Publishing: Cham, Switzerland (2018)
24. Peraza, C., Valdez, F., Garcia, M., Melin, P., Castillo, O.: A new fuzzy harmony search algorithm using fuzzy logic for dynamic parameter adaptation. *Algorithms* **9**, 69 (2016)

Autonomous Garage Parking of a Car-Like Robot Using a Fuzzy PD + I Controller



Enrique Ballinas, Oscar Montiel and Yoshio Rubio

Abstract In this work a path following algorithm for autonomous garage parking was designed. The main objective of the algorithm is to park a vehicle avoiding collisions. We designed a fuzzy PD + I controller to decrease the error generated between the real position and the previously generated objective position. We present simulations results to validate the analysis and to demonstrate how the fuzzy controller solved this tracking problem.

Keywords Autonomous parking · Fuzzy controller · Path planning · Mobile robot

1 Introduction

In the last decade, the number of vehicles in the cities has increased enormously. Looking for a parking space has become a difficult task for the drivers, generating a loss of time and fuel, besides causing accidents due to the lack of experience, age or possible illnesses that decrease the skills to perform parking maneuvers.

Many proposals have been made with the aim of solving this problem. In general, we can mention two approaches. The first is based on implementing algorithms that allow the parking place to be intelligent, that is, the user leaves the vehicle in the main entrance, and a trolley picks up the car and takes it to a parking spot. For example, in [1] Thomas and Kooor used the idea mentioned above but they designed a genetic algorithm to decrease the waiting time and traffic. In [2] Xiao and Zuo developed an algorithm that uses RFID technology to detect which vehicles can enter the parking

E. Ballinas · O. Montiel (✉) · Y. Rubio

Instituto Politécnico Nacional, Centro de Investigación y Desarrollo de Tecnología Digital (IPN-CITEDI), Av. Instituto Politécnico Nacional No. 1310, Colonia Nueva Tijuana, Tijuana 22435, Baja California, Mexico
e-mail: oross@ipn.mx

E. Ballinas

e-mail: lballinas@citedi.mx

Y. Rubio

e-mail: rubio@citedi.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_25

325

lot, once detected, it sends a message to the driver by an LCD to let him know in which space it can be parked. In a similar way Cho, Park, Kim et al. [3] propose a machine learning-based object detection algorithm to classify empty or occupied parking spaces in a parking lot. Qian and Hongyan [4] developed a Parking Guidance and Information System based on intelligent mobile phone terminal; the system uses an improved Dijkstra algorithm to generate the shortest route to the parking space.

The second approach attacks the problem by designing algorithms that allow the vehicle to perform the parking maneuvers autonomously. In this branch, we can find three methods: geometric-based methods, heuristic-based methods and methods based on control theory.

The geometric-based methods are characterized by the design of the desired route and then compared with the real path. In [5] Yi, Xin, Lu and Xing-he developed an algorithm for the garage parking path of Ackerman steering vehicles. The starting point is known previously while the end-point is calculated using the coordinates of the parking space.

In some cases, this method is also combined with heuristic and control theory methods as in [6] Huang and Hsu; they designed a backward auto-parking path for parallel and garage parking. To reduce the error between the actual trajectory and the desired trajectory, they implemented a PI control. They also developed a Self-Organizing Fuzzy Control that can generate or modify its knowledge base and fuzzy rules continuously, based on a learning strategy to generate the real-time steering angle control commands during reversed parking process.

The characteristic of the heuristic-based methods is that they are based on the imitation of human expertise, such as fuzzy logic [7–11], or in search based-methods. In [12] Zips, Bock and Kugi designed an optimization algorithm where the orientation angle, the movement of the vehicle (back or forward), the start position and the end position of the parking path are the objective function. They used the convex decomposition method to detect obstacles in the environment. The algorithm consists of two phases; phase A takes the vehicle to the starting point where the parking process will begin, and phase B consists of applying the necessary maneuvers to park the vehicle properly.

Some works consider obstacles in motion [13, 14]. In these cases, the parking problem becomes an optimization problem that is generally composed of the mechanical restrictions of the vehicle, the kinematic model, and the collision-free restrictions, while others consider the spaces of parking as cells, which can also be vertices of a graph and thus implement algorithms such as A* or similar [15, 16].

An example of methods based on control theory is the work of [17], where it uses a robust controller for parallel parking which does not require the vehicle to follow a previously calculated route. The route is determined by polytopes which contain the initial state of the vehicle and the final state where the parking maneuvers will end.

In this work, a fuzzy PD + I controller was designed to decrease the error between the real path and the desired path. The shape of the membership functions was manually tuned as well as the fuzzy controller rules. The main contribution of this work is the mathematical development to find the key points where the car-like robot makes the turns to be able to perform parking maneuvers.

This paper is organized as follows: Sect. 2 explains the kinematic model of a car-like robot; in Sect. 3, the path for the garage parking task is designed; Sect. 4 describes the proposed fuzzy controller; Sect. 5 presents the control diagrams that describes how to reduce the error in trajectories; the performance of the fuzzy controller is presented in Sect. 6; finally, in Sect. 7, the conclusions and future work are discussed.

2 Kinematic Model of a Car-like Robot

The garage parking algorithm is designed for a car-like robot with nonholonomic restrictions. Mathematically, a nonholonomic system is one that expresses its restrictions with derivatives of the form $f(q, t) = 0$ where, q are the constraints of the system. On the other hand, if the system is expressed without derivatives in the form $f(q, \dot{q}, \ddot{q}, t) = 0$ the system is holonomic [18].

To design the parking path is necessary to know the kinematic model of the mobile robot, which moves in a two-dimensional plane (x, y) . Due to the nonholonomic nature of the robot that has three degrees of freedom x, y and, the orientation angle θ , which is normally the angle formed with respect to the x axis.

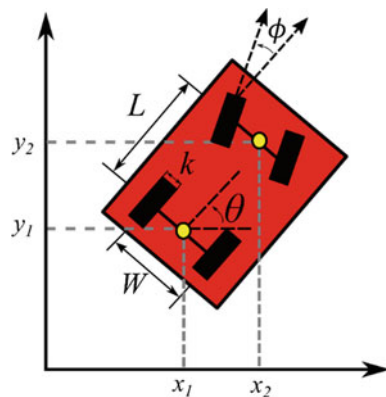
The parking trajectory is calculated using the kinematic model as mentioned above by taking as a reference point the middle of the rear axle. At any given time, the system has an instantaneous linear velocity v and an angular velocity ω (see Fig. 1).

The scheme of the kinematic model of the robot is represented by the wheelbase L , the width of the axle W and the width of the rim k . The restrictions of the robot are given by the steering angle limiting both position and orientation. The restrictions of a front-steered robot are described as follows [19, 20]:

$$\dot{y}_1 \cos \theta - \dot{x}_1 \sin \theta = 0, \tag{1}$$

$$y_2 \cos(\theta + \phi) - x_2 \sin(\theta + \phi) = 0, \tag{2}$$

Fig. 1 Kinematic model of a car-like robot



where $x_2 = x_1 + L \cos \theta$ and $y_2 = y_1 + L \sin \theta$. With this information it is possible to determine the total restrictions of our system, these are presented in Eq. (2).

$$q = \begin{bmatrix} x_1 \\ y_1 \\ \theta \\ \phi \end{bmatrix} \quad (3)$$

It is important to bear in mind that for the parking process the velocity of the robot is low as well as its mass, for that reason the influence of slippage is not considered [21]. Thus, the instantaneous curvature g is calculated as follows:

$$g = \frac{\tan \phi}{L} \quad (4)$$

Now it is necessary to determine the relationship between the elements of q to obtain the kinematic equations of the system. Using the system shown in Fig. 1 and the instantaneous curvature and linear velocities equations it is possible to obtain the kinematic model of the car-like robot.

$$\dot{q} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 \\ \sin \theta & 0 \\ \frac{\tan \phi}{L} & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} v \\ \omega \end{bmatrix} \quad (5)$$

The linear and angular velocities, as well as the steering angle, are bounded as we can see in Eq. (6).

$$\begin{cases} |v| \leq v_{\max} \\ |\phi| \leq \phi_{\max} \\ |\omega| \leq \omega_{\max} \end{cases} \quad (6)$$

3 Path Planning for Garage Parking

For the design of the parking trajectory, the geometric approach made by Ballinas et al. [22] was taken as a basis. The garage parking path is made by five points, forming two straight lines and two curves as shown in Fig. 2.

The initial coordinates are (X_i, Y_i) . The distance from (X_i, Y_i) to (X_{c1}, Y_{c1}) is heuristically defined as $2L$, and is the first segment of the path. To obtain the value of the initial coordinates, we need to determine the origin point and calculate (X_{c1}, Y_{c1}) . The origin point is defined as C_1 (see Fig. 3).

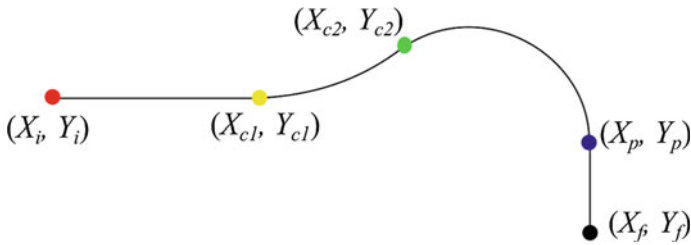


Fig. 2 Complete garage parking path

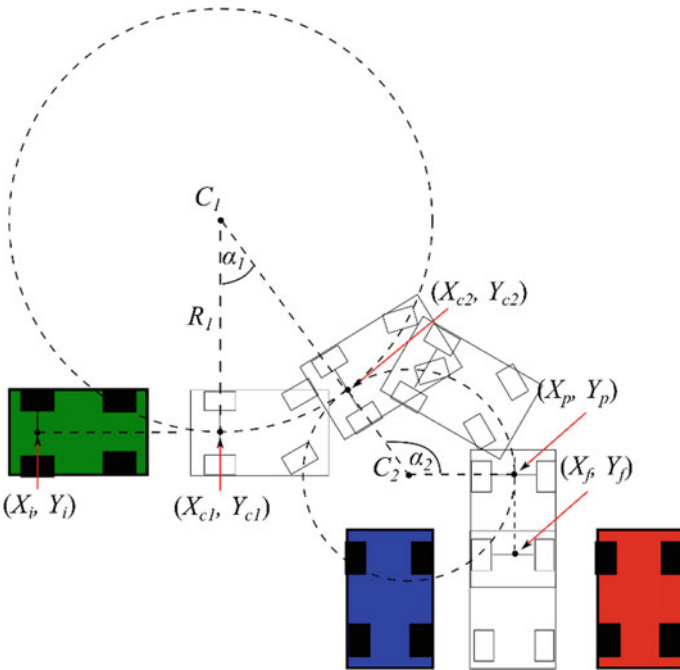


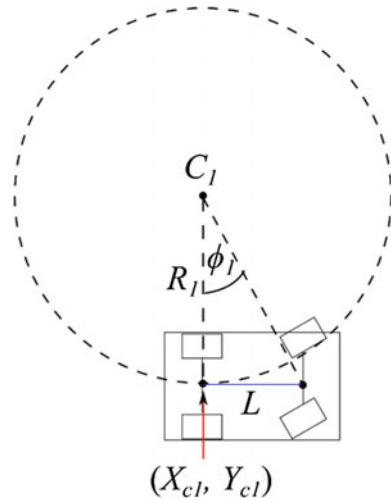
Fig. 3 The geometric approach of the garage parking

As $C_1(0, 0)$ it is possible to say that the values of $X_{c1} = 0$ and $Y_{c1} = R_1$. With this information, we can calculate the origin point (X_i, Y_i) using the Eq. (7).

$$\begin{aligned} X_i &= -2L \\ Y_i &= Y_{c1} \end{aligned} \tag{7}$$

Before calculating the next parking point, it is necessary to know the radius R_1 of the circle that forms the first arc. This arc begins in (X_{c1}, Y_{c1}) until (X_{c2}, Y_{c2}) (see Fig. 2). Using trigonometric identities, it is possible to find out the value of this radius as can be seen in Fig. 4.

Fig. 4 A trigonometric approach to calculate R_1



The tangent of the angle is equal to the opposite divided by the adjacent, therefore $R_1 = \frac{L}{\tan \phi_1}$. The angle of direction influences the size of the circle, the larger the angle the smaller the radius of the circle and vice versa, in this case, the angle is $\phi_1 = \phi_{max}$.

The next points are (X_{c2}, Y_{c2}) , these coordinates indicate the change of direction to park the vehicle.

$$Y_{c2} = R_1 - \frac{W}{2} \tag{8}$$

$$X_{c2} = R_1 \cos \alpha_1 \tag{9}$$

Equation (9) indicates that it is possible to determine the value of R_1 if we know the value of the opening angle α_1 of the first arc. To calculate the value of the opening angle α_1 we can substitute Eq. (8) for the following equation $Y_{c2} = R_1 \sin \alpha_1$. Since the value of Y_{c2} is already known, it is possible to obtain the value of the opening angle α_1 as shown in the Eq. (10).

$$\alpha_1 = \sin^{-1} \left(1 - \frac{W}{2R_1} \right) \tag{10}$$

The coordinates that indicate the end of the second arc and the start of the straight segment to enter the parking space (X_p, Y_p) can be calculated using the parametric equations as we did in Eq. (9), the main difference is that the generated circle is outside of the origin point, see Eq. 11.

$$\begin{aligned} X_p &= C_{21} + R_2 \cos \alpha_2 \\ Y_p &= C_{22} + R_2 \sin \alpha_2 \end{aligned} \quad (11)$$

where, C_{21} and C_{22} represent the coordinates of the center of the circle and they can be calculated using:

$$\begin{aligned} C_{21} &= X_{c2} + W - k \\ C_{22} &= Y_{c1} + \frac{W}{2} \end{aligned} \quad (12)$$

The radius R_2 is calculated in the same way as the radius R_1 was calculated, the difference is the steering angle used. The radius R_2 uses the maximum steering angle $\phi_2 = \phi_{\max}$ generating a circle smaller than the circle generated with the radius R_1 . To calculate the opening angle α_2 it is necessary to know at least one of the coordinates (X_p, Y_p) , if we look at Fig. 3 it is possible to calculate Y_p as $Y_p = R_1 + \frac{W}{2}$. As we already know the value of Y_p , from Eq. (11) we can calculate the opening angle α_2 as we can see in Eq. (13). With all this information it is already possible to calculate the value of X_p .

$$\alpha_2 = \sin^{-1} \left(\frac{2R_1 + W - 2C_{22}}{2R_2} \right) \quad (13)$$

The last coordinates are (X_f, Y_f) , since X_f is at the same level as X_p we can define the following equation $X_f = X_p$, while Y_f is calculated as $Y_f = Y_p + L$.

4 Fuzzy Inference System

The fuzzy inference system (FIS) used for the development of this controller is of the Mamdani type. The main objective is to generate the correct steering angle to allow the mobile robot to follow the desired path with the least possible error. The Membership Functions (MFs) of the FIS are shown in Fig. 5, 6 and 7. Table 1 shows the type and parameter values of the MFs. Table 2 shows the rule matrix. The system has two inputs: the error between the current and the desired direction, and error change. Every input has five MFs: NB, N, Z, P, PB. The inputs are provided in radians.

The output variable has five MFs: BD, D, Z, I, BI (see Fig. 7). The universe of discourse of the output is based on the maximum steering angle of the robot and is given in radians.

Table 1 summarizes the fuzzy rules that determine the behavior of the fuzzy controller. Table 2 shows the parameters and types of membership functions that each of the inputs and outputs of the system have.

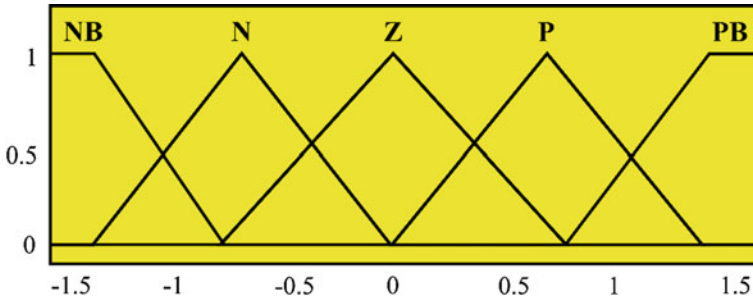


Fig. 5 Input membership functions for the error E_p

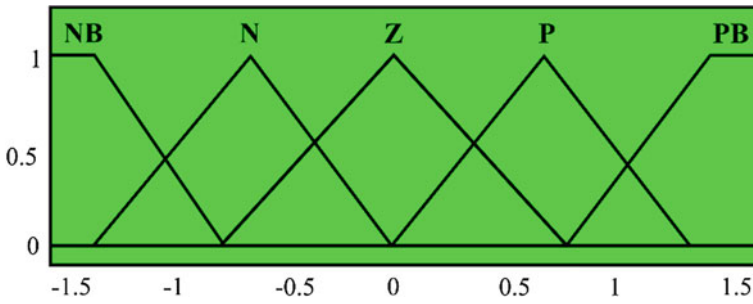


Fig. 6 Input membership functions for the error change ΔE_D

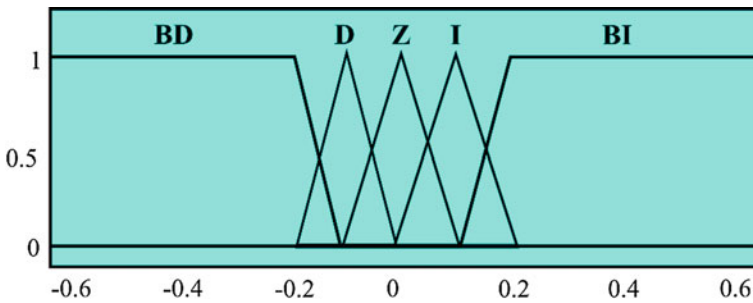


Fig. 7 Membership functions of the output o

Table 1 Fuzzy rule matrix of the FIS

	NB	N	Z	P	PB
NB	BI	BI	BI	BI	BI
N	BI	I	I	Z	D
Z	BI	I	Z	D	BD
P	I	Z	D	D	BD
PB	BD	BD	BD	BD	BD

Table 2 Linguistic inputs and output variables for the FIS

Variable name	Term name	Type	Parameters
E_p	NB	Trapezoidal	$[-1.745 -1.745 -1.374 -0.7851]$
E_p	N	Triangular	$[-1.374 -0.7067 0]$
E_p	Z	Triangular	$[-0.7851 0 0.7851]$
E_p	P	Triangular	$[0 0.7067 1.374]$
E_p	PB	Trapezoidal	$[0.7851 1.374 1.745 1.745]$
ΔE_D	NB	Trapezoidal	$[-1.745 -1.745 -1.374 -0.7851]$
ΔE_D	N	Triangular	$[-1.374 -0.6867 0]$
ΔE_D	Z	Triangular	$[-0.7851 0 0.7851]$
ΔE_D	P	Triangular	$[0 0.687 1.37]$
ΔE_D	PB	Trapezoidal	$[0.7851 1.374 1.745 1.745]$
O	BD	Trapezoidal	$[-0.8 -0.6283 -0.1885 -0.06283]$
O	D	Triangular	$[-0.1885 -0.09425 0]$
O	Z	Triangular	$[-0.06283 0 0.06283]$
O	I	Triangular	$[0 0.09425 0.1885]$
O	BI	Trapezoidal	$[0.06283 0.1885 0.6283 0.8]$

5 Control Strategy

To reduce the error between the desired trajectory and the actual trajectory, a fuzzy PD + I control was used (see Fig. 8). The proportional and derivative are part of the fuzzy controller while the integral is added to the output of the controller.

As mentioned before, the controller has two inputs: the error and the error change. The error e is caused by the difference between the orientation angle and the steering angle of the robot. The error is the number of radians that the steering angle of the robot must rotate to equal the orientation angle and thus follow the desired path.

The error change Δe is the difference between the current error and the previous error, while the sum of the error Σe as the name indicates is the sum of the current error and the previous error. Each error is multiplied by its corresponding gain as can

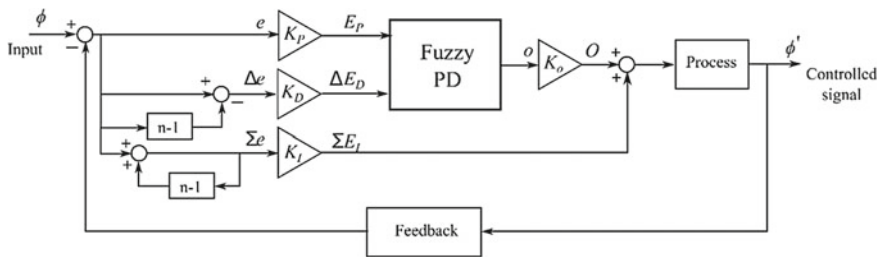


Fig. 8 Fuzzy PD + I controller

be seen in Eq. 14.

$$\begin{aligned}
 E_p &= K_p e \\
 \Delta E_D &= K_D \Delta e \\
 \Sigma E_I &= K_I \Sigma e
 \end{aligned}
 \tag{14}$$

The output of the fuzzy controller o is multiplied by its gain to then add the value of the integral error as can be seen below:

$$O(\phi) = K_o o + \Sigma E_I
 \tag{15}$$

Finally, the corrected steering angle ϕ' is defined as the value of the current steering angle plus the final output of the fuzzy controller $O(\phi)$ (see Eq. (16)).

$$\phi' = \phi + O(\phi)
 \tag{16}$$

6 Results

The simulations for the garage parking path and the fuzzy inference system were developed in Matlab. The garage parking path was calculated using a car-like robot with 0.26 m of wheelbase, with this value and the steering angle is possible to calculate the radius of the first and second circle. A steering angle of 0.314 radians (18°) was used to calculate the first radius giving as a result 0.8 m. The value of the second radius is 0.35 m. The steering angles and the constants K_p, K_D, K_I, K_o were found experimentally. The values of the constants are $K_p = 1, K_D = 1, K_I = 0.2, K_o = -\pi$. The results of the fuzzy controller are shown in Table 3.

To calculate the mean squared error (MSE), we use the l_2 -norm defined as $\|e\|_2 = \sqrt{\sum_{i=1}^n |e_i|^2}$ and for the maximum error we used the $\|e\|_\infty$ norm defined as $\|e\|_\infty = \max_{1 \leq i \leq n} |e_i|$. For the MSE the number of samples is 60 and for the maximum error is 298 samples.

In the first curve (see Fig. 9), the error has a big overshoot this is because the car-like robot is get into the curve, then after 10 s the controller apply a correction and decrease the error until 0.022 radians, the error keeps stable but after approximately

Table 3 Error comparison in the fuzzy controller

	Steady state error for the first curve (rad)	Steady state error for the second curve (rad)	Maximum error (rad)	Mean square error (rad)
Fuzzy PD + I	0.022	-0.00066	0.6916	0.1007

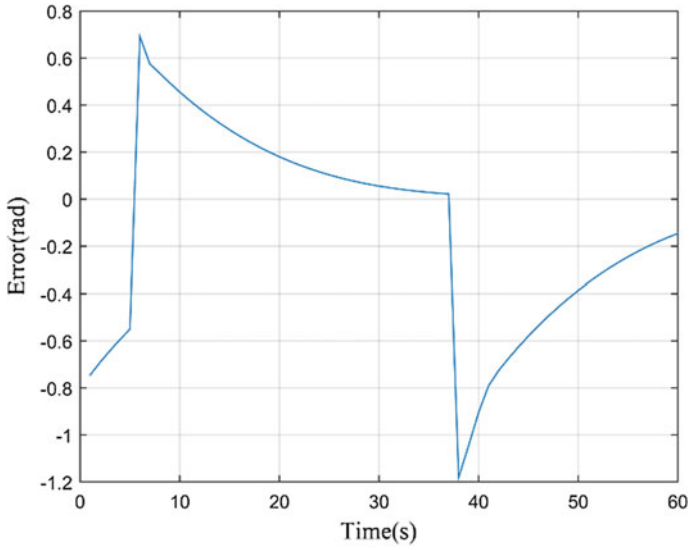


Fig. 9 Error behavior in the first arc

37 s a negative increment occurred, the reason is because the robot is coming out of the first curve and is detecting the second curve.

For the second curve, (see Fig. 10) the controller takes approximately 20 s to take the error to a value very close to zero (0.00066 radians) and remains stable until

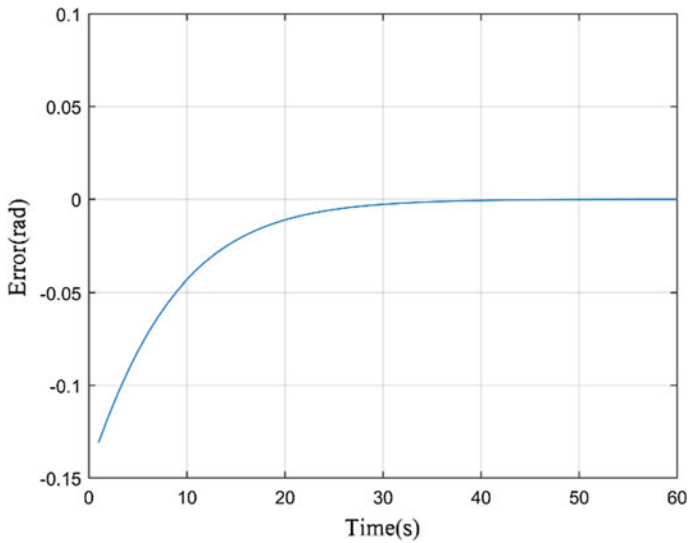


Fig. 10 Error behavior in the second arc

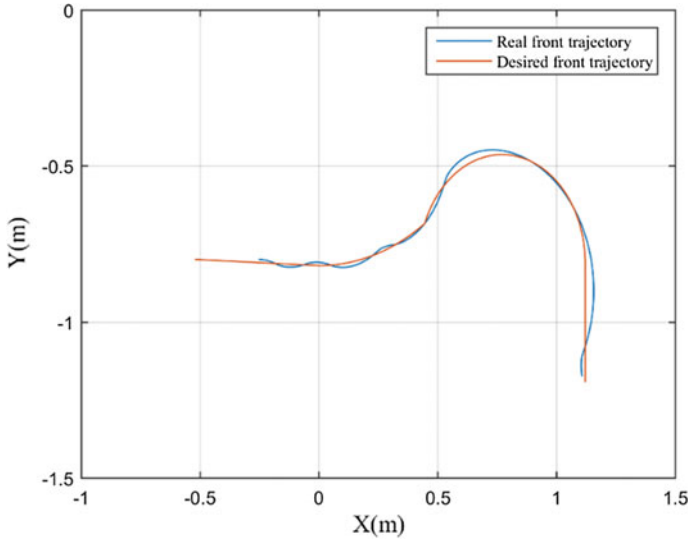


Fig. 11 Real front trajectory vs desired front trajectory

the end of the curve, this happens because the designed of the second curve was calculated with a steering angles of 36° (0.628 rad), it means the maximum steering angle, and for that reason the car-like robot follows the curve easier than the first curve.

Since the problem is front garage parking, the control system was designed to minimize the error from the front part of the robot and the desired path. The front part of the car-like robot follows the desired trajectory with a maximum error distance of 0.1871 m and an MSE of 0.0105 m, in contrast, the rear axle of the car-like robot has a maximum error of 0.4441 m and a MSE of 0.0938 (see Fig. 11).

One of the important aspects to consider is the movement of the wheels for the change of direction. Usually, in a robot, the element in charge is a servomotor; this having mechanical parts is prone to damage due to sudden movements. Therefore, it is important to keep in mind that the fuzzy controller apart from decreasing the error must be able not to cause sudden movements in the steering mechanism. Figure 12 shows the changes of direction made by the steering mechanism; the changes are not abrupt, furthermore, after 10 s the fuzzy control stabilizes the steering mechanism. Although after 20 s a peak is generated, but this is due to the end of the second curve and the beginning of the last segment of parking.

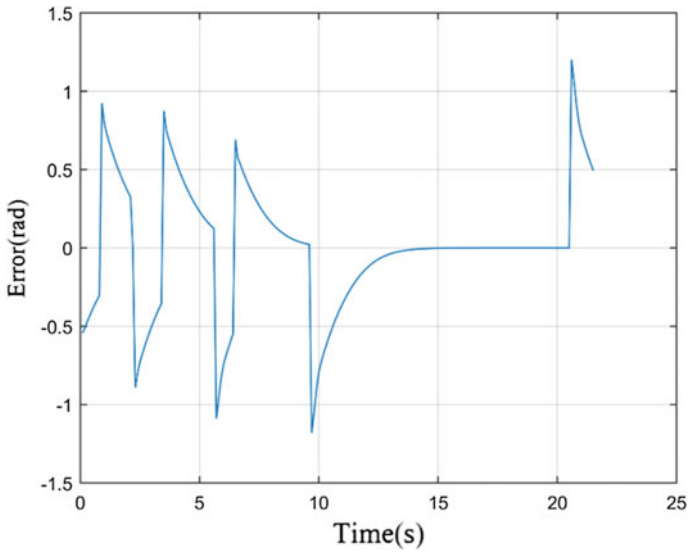


Fig. 12 The behavior of the error in the steering angle

7 Conclusions

Different techniques have been used to solve the problem of garage parking, some more effective than others. For the development of this work, we opted for the design of a path following using a fuzzy PD + I control to reduce the error generated between trajectories.

The results showed that the designed fuzzy PD + I controller has the capacity to do the task of garage parking for a car-like robot. As future work, it is intended to design an evolutionary algorithm to adjust the membership functions of the FIS and, thereby, reduce the sudden changes that the element responsible for the steering of the car-like robot may suffer, in addition to implementing the path following algorithm in a scale vehicle.

References

1. Thomas, D., Kooor, B.C.: A genetic algorithm approach to autonomous smart vehicle parking system. *Procedia Comput. Sci.* **125**, 68–76 (2018)
2. Xiao, X., Zou, Q.: The architecture of the RFID-based intelligent parking system. *Adv. Intell. Syst. Comput.* **686**, 119–124 (2018)
3. Cho, W., Park, S., Kim, M.-J., Han, S., Kim, M., Kim, T., Kim, J., Paik, J.: Robust parking occupancy monitoring system using random forests. In: *International Conference on Electronics, Information and Communication, ICEIC 2018*, 2018 January, pp. 1–4 (2018)

4. Qian, Y., Hongyan, G.: Study on parking guidance and information system based on intelligent mobile phone terminal. In: Proceedings—8th International Conference on Intelligent Computation Technology and Automation, pp. 871–874 (2016)
5. Yang, Y., Qu, X., Zhu, H., Zhang, L., Li, X.-H.: Design and implementation of path planning algorithm for vehicle parking. *J. Beijing Inst. Technol. (English Edition)* **25**, 502–511 (2016)
6. Huang, S.-J., Hsu, Y.-S.: Parking space detection and trajectory tracking control for vehicle auto-parking. *World academy of science, engineering and technology. Int. J. Mech. Mechatron. Eng.* **11**, 1712–1718 (2017)
7. Ryu, Y.-W., Oh, S.-Y., Kim, S.-Y.: Robust automatic parking without odometry using an evolutionary fuzzy logic controller. *Int. J. Control Autom. Syst.* **6**, 434–443 (2008)
8. Chaparro-Fonseca, J.M., Avilés-Sánchez, O.F.: Diseño De Un Controlador Difuso Para El Estacionamiento De Un Automóvil En Reversa. *Sci. Technica Año* **18**(1), 25–31 (2013)
9. Aye, Y.Y., Watanabe, K., Maeyama, S., Nagai, I.: An automatic parking system using an optimized image-based fuzzy controller by genetic algorithms. *Artif. Life Robot.* **22**, 139–144 (2017)
10. Nakrani, N., Joshi, M.: An intelligent fuzzy based hybrid approach for parallel parking in dynamic environment. *Procedia Comput. Sci.* **133**, 82–91 (2018)
11. Mitrović, S.T., Durović, Ž.M.: Fuzzy logic controller for bidirectional garaging of a differential drive mobile robot. *Adv. Robot.* **24**, 1291–1311 (2010)
12. Zips, P., Bock, M., Kugi, A.: A fast motion planning algorithm for car parking based on static optimization. In: *IEEE International Conference on Intelligent Robots and Systems*, pp. 2392 (2013)
13. Li, B., Shao, Z.: A unified motion planning method for parking an autonomous vehicle in the presence of irregularly placed obstacles. *Knowl. Based Syst.* **86**, 11–20 (2015)
14. Hu, X., Chen, L., Tang, B., Cao, D., He, H.: Dynamic path planning for autonomous driving on various roads with avoidance of static and moving obstacles. *Mech. Syst. Signal Process.* **100**, 482–500 (2018)
15. Serpen, G., Dou, C.: Automated robotic parking systems: real-time, concurrent and multi-robot path planning in dynamic environments. *Appl. Intell.* **42**, 231–251 (2015)
16. Li, B., Yang, L., Xiao, J., Valde, R., Wrenn, M., Leflar, J.: Collaborative mapping and autonomous parking for multi-story parking garage. *IEEE Trans. Intell. Transp. Syst.* **19**, 1629–1639 (2018)
17. Ornik, M., Moarref, M., Broucke, M.E.: An automated parallel parking strategy using reach control theory. *IFAC-PapersOnLine* **50**, 9089–9094 (2017)
18. Li, B., Shao, Z.: Simultaneous dynamic optimization: a trajectory planning method for nonholonomic car-like robots. *Adv. Eng. Softw.* **87**, 30–42 (2015)
19. Güzey, H.M., Dierks, T., Jagannathan, S., Acar, L.: Hybrid consensus-based control of nonholonomic mobile robot formation. *J. Intell. Rob. Syst. Theor. Appl.* **88**(1), 181–200 (2017)
20. Ni, J., Hu, J.B.: Dynamic control of autonomous vehicle at driving limits and experiment on an autonomous formula racing car. *Mech. Syst. Signal Process.* **90**, 154–174 (2017)
21. Rosolia, U., De Bruyne, S., Alleyne, A.G.: Autonomous vehicle control: a nonconvex approach for obstacle avoidance. *IEEE Trans. Control Syst. Technol.* **25**(2), 469–484 (2017)
22. Ballinas, E., Montiel, O., Castillo, O., Rubio, Y., Aguilar, L.T.: Automatic parallel parking algorithm for a car-like robot using fuzzy PD + I control. *Eng. Lett.* **26**(4), 447–454 (2018)

Analysis of P, PI, Fuzzy and Fuzzy PI Controllers for Control Position in Omnidirectional Robots



Leticia Luna-Lobano, Prometeo Cortés-Antonio, Oscar Castillo and Patricia Melin

Abstract In this paper, it will be presented the analysis of different controllers schemes for an omnidirectional robot. The models studied are *P*, *PI*, *Fuzzy* and *Fuzzy PI* controllers. These were implemented using Simulink Tool of Matlab and tested in Robotino View. The objective is to control the command velocities to drive two linear directions of the robot, from an initial position to a final position without obstacles. The analysis compares time and distance to determine the performance of each them.

Keywords *P* control · *PI* control · Fuzzy control · Omnidirectional robot

1 Introduction

At present, manufacturing companies are working on the implementation of technologies that improve their operations through automation. The use of robots is one of them [1–3]. Robots are programmable devices, they have a wide application in manufacturing industry, with a precondition steps that automate their operation [4, 5].

Many publications have based their study on kinematics of mobile robots, which are controlled by the velocity input [6]. Based on nonlinearities and uncertainty various control systems have been designed [7]. Fuzzy Logic in intelligent computing is strongly applied to robotic solutions to improvement automation processes [8–10]. The feedback information about the robot pose can be estimated from encoder optical

L. Luna-Lobano · P. Cortés-Antonio (✉) · O. Castillo · P. Melin
Tijuana Institute of Technology, Tijuana, Baja California, Mexico
e-mail: prometeo.cortes@tectijuana.edu.mx

L. Luna-Lobano
e-mail: LLunaLobano@gmail.com

O. Castillo
e-mail: ocastillo@tectijuana.mx

P. Melin
e-mail: pmelin@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_26

sensor [11], and this is used to computed command linear and angular velocities obtained.

In [12], a simple adaptive fuzzy logic controller is proposed, which utilizes a fuzzy logic system for estimating the unknown robot parameters for tracking control of wheel mobile robot and it required only position measurement. Fuzzy logic theory is a powerful soft computing technique for complex and non-linear control systems based on human expert knowledge [13, 14].

Omnidirectional mobile robots have as a main feature, they can move simultaneously and independently in rotational and two translational motions on a flat surface [15]. Fuzzy logic is a good alternative for increasing the capabilities of autonomous mobile robots in an unknown dynamics environment by integrating human experience [16]. Fuzzy theory has shown advantage in dealing with uncertainty and nonlinear problems [17]. Fuzzy Logic can be applied in several solutions such obstacle avoidance [18], Path Navigation [13] in robotic solutions, and autonomous Humanoid Robot Control [19]. Other works using Intelligent Control applied autonomous navigation robots can be found in [20, 21].

In this article the navigation of omnidirectional robot is addressed using different types of controllers, P , PI , Fuzzy and Fuzzy-PI controller. P and PI controllers are conventional controllers, but the major drawback is the estimation of its parameters in nonlinear systems. Fuzzy control helps to compute command velocities based on fuzzy rules taken from human experience. Last, Fuzzy-PI computes the parameter values for k_p and k_i of a PI controller with dynamical parameters.

The paper has the following structure: Sect. 2 introduces the kinematics of Omnidirectional Robot. Section 3 explain the different control schemes, Sect. 4, includes an implementation of control schemes, and the Sect. 5 includes results obtained on test execution. Finally, conclusions are about this paper are presented.

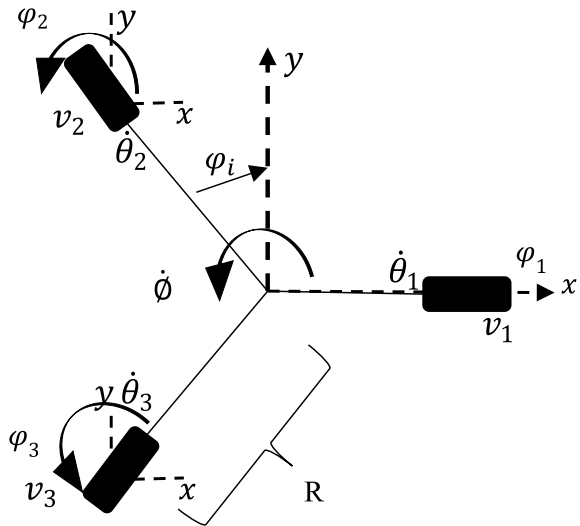
2 Kinematics of Omnidirectional Robot

Main characteristic of omnidirectional robots are they can move in three directions, that is, they can move on x, y translational directions, and additionally, they can perform rotational motions. The omnidirectional robot used in this work is the Robotino of Festo [22]. Robotino is a holonomic mobile robot with three omnidirectional drive units, meaning that the controllable degrees of freedom equal the total degree of freedom of the robot [16].

Robot odometer uses data state from the motions of the three actuators to estimate the robot position in time. In the current design problem, we use odometer module (incorporated into Robotino tools) to estimate the mobile robot position relative to the known starting location.

Figure 1 illustrates the kinematics of an omnidirectional robot that contains 3 wheels, each one can be identify by their angular location φ_i and their angular velocity $\dot{\theta}_i$, with $i = 1, 2, 3$. The robot position with respect to reference frame is defined as $\mathbf{x} = [x, y, \theta]^T$.

Fig. 1 Kinematics of an omnidirectional robot



According to [13], the kinematics model of Robotino is given by next equation

$$\begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} V_1 \\ V_2 \\ V_3 \end{bmatrix} = \frac{1}{r} \begin{bmatrix} -\sin(\vartheta + \varphi_1) \cos(\vartheta + \varphi_1) R \\ -\sin(\vartheta + \varphi_2) \cos(\vartheta + \varphi_2) R \\ -\sin(\vartheta + \varphi_3) \cos(\vartheta + \varphi_3) R \end{bmatrix} \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\vartheta} \end{bmatrix}$$

Or simpler, in matrix notation

$$\dot{\theta} = \frac{1}{r} V = \frac{1}{r} P(\vartheta) \dot{x}$$

With

- V_i Linear velocity of the wheel i
- R Distance between a wheel and the center of the robot
- $\dot{\vartheta}$ Angular velocity of the robot
- φ_i Angular location of the wheel i
- r Radius of the wheel
- $P(\vartheta)$ Transformation matrix between the angular velocities of the wheels and the robot velocity vector $(\dot{x}, \dot{y})^T$.

3 Control Schemes

Main objective of this work is to analysis the behavior of Robotino to reach a desired position by computing the command velocities v_x and v_y of the robot using different

control schemes. In this paper, we study four controllers: (a) proportional, P ; (b) proportional-integral, PI ; (c) Fuzzy, F and; (d) Fuzzy proportional-integral, FPI . Figure 2 shows general scheme of the feedback control implemented in this work. Robotino tools provide an odometer module to estimate the pose, x , of the robot. In the controller block of Fig. 2, it will be implemented the different mentioned controllers.

Table 1 provides the equations to compute the command velocities, v_x and v_y for P and PI controllers.

To compute v_x and v_y in F and FPI controllers, it is accomplished through fuzzy logic controllers as is shown in Fig. 3. F controller is composed by 2 parallel Mamdani Fuzzy systems with 1-input and 1-Output, 5 fuzzy rules and 5 fuzzy sets per variable. FPI is unlike F controller as its parallel fuzzy systems have 2 Outputs and 3 FS per output variable, also, it can note the outputs are control parameters k_{px} ,

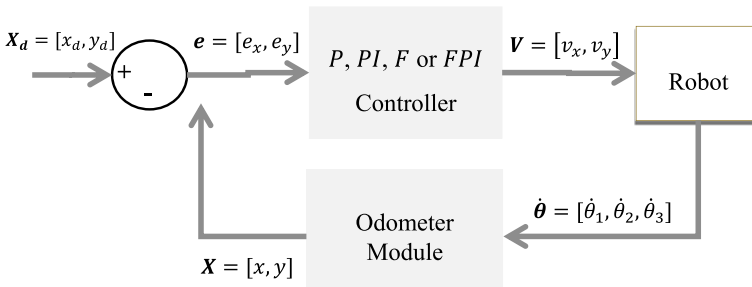


Fig. 2 General feedback control scheme

Table 1 Equations of command velocities for P and PI controllers

P	PI
$v_x = k_{px}e_x$	$v_x = k_{py}e_x + k_{ix} \int e_x$
$v_y = k_{py}e_y$	$v_y = k_{py}e_y + k_{iy} \int e_y$

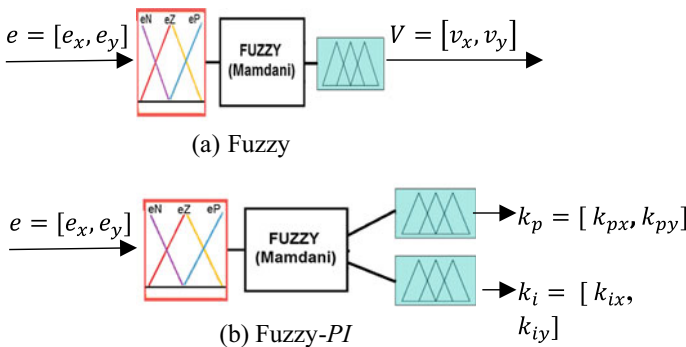


Fig. 3 Schemes of a F and b FPI controllers

k_{py} , k_{ix} and k_{iy} instead of v_x or v_y . For better visualization, Fig. 3 only shows one of two embedded controllers in each scheme.

4 Implementation of Control Schemes

In this section, we introduce the Simulink implementation of the different controller systems for tracking trajectory of Robotino. For better visualization, in each system the controller block will show only one of two parallel schemes (to calculate command velocities v_x and v_y from e_x and e_y respectively). Figure 4 depicts the P controller, its input variable is error and its output is velocity. Where, k_p is a constant value and x_d is the desired position.

Figure 5 depicts the PI controller, which its input variable is error and its output is velocity. Where, k_p and k_i are the constants for proportional and integral terms respectively.

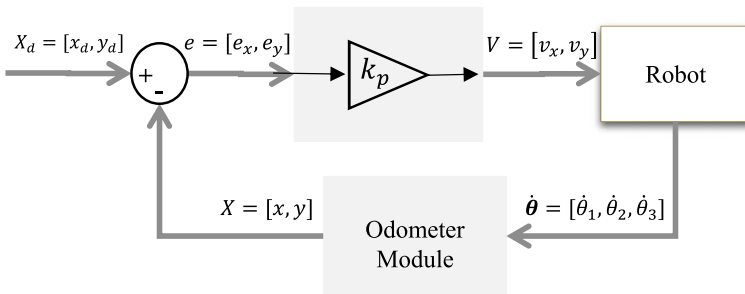


Fig. 4 Scheme of P controller

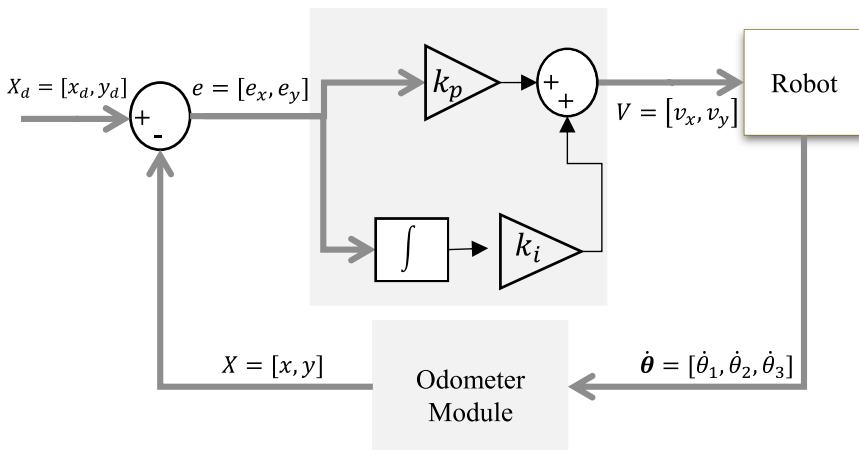


Fig. 5 Scheme of PI controller

Figure 6 depicts fuzzy controller, its input is error and its output is the velocity.

Figure 7 shows the fuzzy variables error and velocity, which represent input and output the fuzzy model with values equally spaced along its range. Table 2 lists the linguistic values of the fuzzy variables.

The number of Membership Functions (MF) assigned to each input-output variable was chosen empirically. Finally, the rule base created is listed as follows.

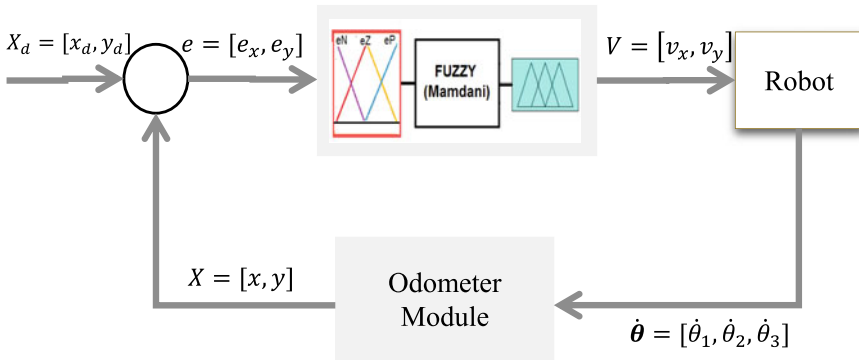


Fig. 6 Error and velocity variables of Fuzzy controller

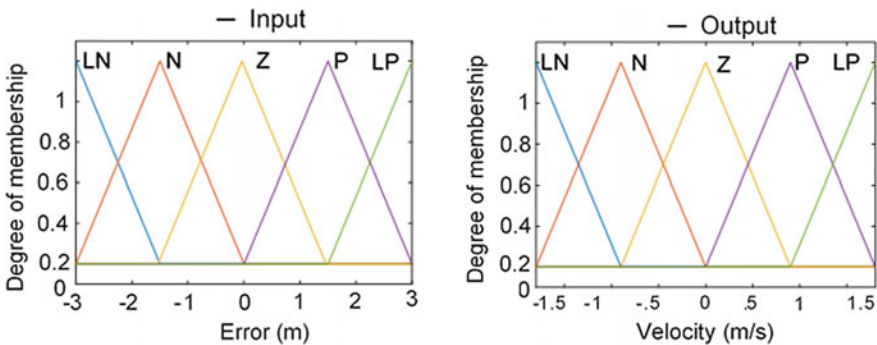


Fig. 7 Error and velocity variables of Fuzzy controller

Table 2 Linguistic values of fuzzy variables of F controller

Error	Velocity
LN: Negative	LN: Large negative
N: Negative	N: Negative
Z: Zero	Z: Zero
P: Positive	P: Positive
LP: Large positive	LP: Large positive

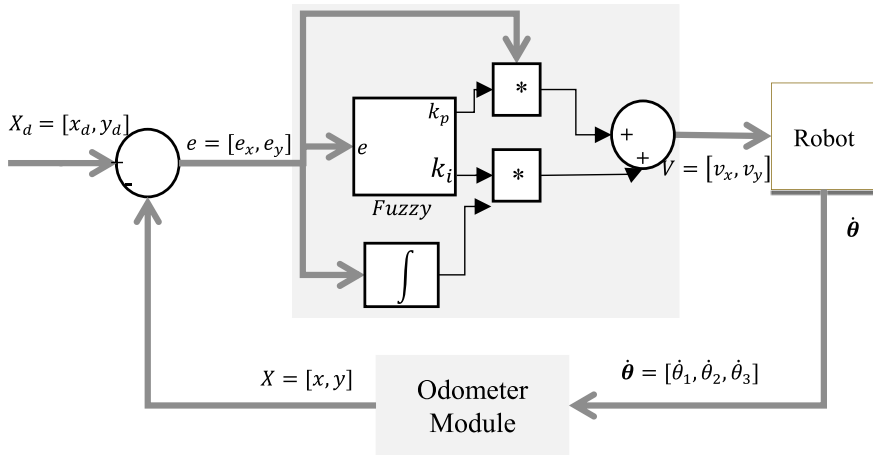


Fig. 8 Schema of FPI controller

- $R_1 = \text{if error is Negative Large then Velocity is Negative Large}$
- $R_2 = \text{if error is Negative then Velocity is Negative}$
- $R_3 = \text{if error is Zero then Velocity is Zero}$
- $R_4 = \text{if error is Positive then Velocity is Positive}$
- $R_5 = \text{if error is Negative Large then Velocity is Negative Large.}$

Figure 8 depicts *FPI* controller which each input is the error of the position and outputs are variables k_p and k_i . Then *FPI* controller computes the dynamic parameters k_p and k_i to be used by a *PI* scheme to calculate command velocities v_x and v_y .

Figure 9 depicts the membership functions assigned to input error, and to output k_p and k_i of fuzzy system. Table 3 lists the names of fuzzy values of the variables. The rule base created for *FPI* controller is listed as follows.

- $R_1 = \text{if error is Negative Large then } k_i \text{ is large and } k_p \text{ is large}$

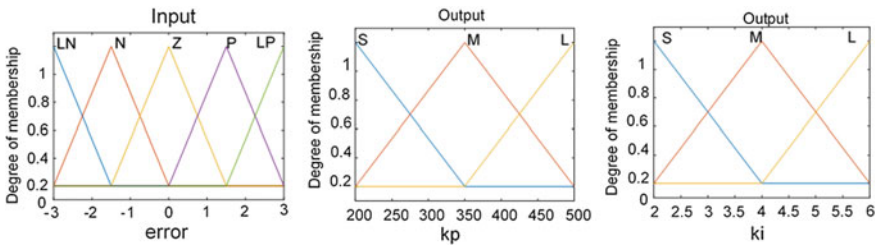


Fig. 9 Error, k_i and k_p variables of *Fuzzy – PI* controller

Table 3 Linguistic values of fuzzy variables of *FPI* controller

Error	K_p	K_i
LN: Large negative	S: Small	S: Small
N: Negative	M: Medium	M: Medium
Z: Zero.	L: Large	L: Large
P: Positive		
LP: Large positive		

$R_2 = \text{if error is Negative then } k_i \text{ is Medium and } k_p \text{ is Medium}$

$R_3 = \text{if error is Zero then } k_i \text{ is Small and } k_p \text{ is Small}$

$R_4 = \text{if error is Positive then } k_i \text{ is Medium and } k_p \text{ is Medium}$

$R_5 = \text{if error is Positive Large then } k_i \text{ is Large and } k_p \text{ is Large.}$

5 Results

To analyze the different control schemes introduced above, we perform three test in which, the target position is $x_{d1} = [3, 3]$, $x_{d2} = [5, 4]$ and $x_{d3} = [5, 2]$ respectively. In first test, run time is 10 s, the gain parameter values for *P* and *PI* controllers are listed in Table 4.

To compare the different controllers, it is calculated the Integral Square Error (ISE) as follows

$$ISE = \int_0^T e^2(t)dt = \int_0^T e_x^2(t)dt + \int_0^T e_y^2(t)dt$$

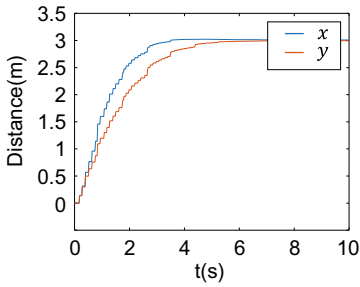
where e_x and e_y are error of each variable

Table 5 contains the integral square errors and settling time (t_s) obtained from the execution in test 1 of different controllers to go from initial position [0,0] to final position [3,3]. Additionally, it shows if the trajectory had overshoot.

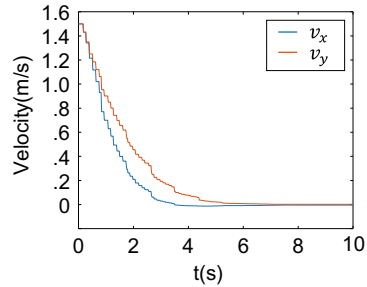
Figure 10 shows the behavior of the Robotino over the x and y position under the different schemes. It can see how Robotino converges to desired position. Additionally, it is shown the command velocities v_x and v_y applied during execution time for each controller.

For second 2, the desired position is $x_{d2} = [5, 4]$. Run time is also 10 s, the gain parameter values for *P* and *PI* controllers are listed in Table 6.

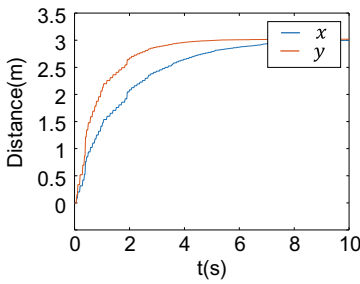
Table 7 contains the integral square errors and t_s time obtained from execution of the different controllers to go from initial position [0,0] to final position [5,4]. Additionally, it shows if the trajectory had overshoot.



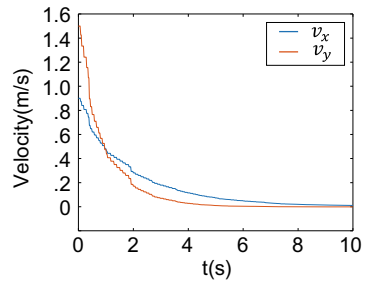
(a) Positions with P controller



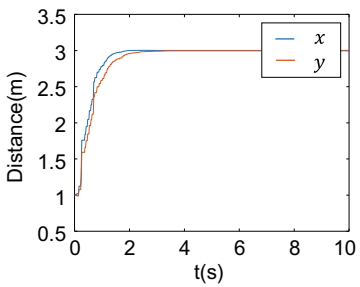
(b) Velocities with P controller



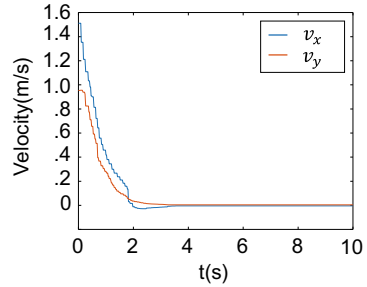
(c) Positions with PI controller



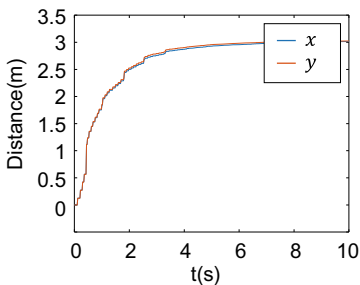
(d) Velocities with PI controller



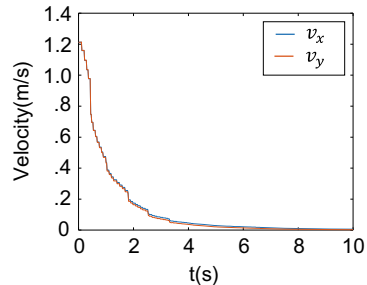
(e) Positions with F controller



(f) Velocities with F controller



(g) Positions with FPI controller



(h) Velocities with FPI controller

Fig. 10 Positions and command velocities for the different control schemes in test 1

Table 4 Gains of P and PI controllers for test 1

Parameters	P	PI
k_{px}, k_{py}	[500,500]	[300,500]
k_{ix}, k_{iy}	–	[2,3]

Table 5 Results of the different control schemes for test 1

Results	P	PI	F	FPI
ISE_x, ISE_y	[65.69,86.34]	[76.90,41.94]	[14.24,17.20]	[54.31,53.57]
ISE	152.03	118.84	31.44	107.88
t_{sx}, t_{sy}	[3.5,7.14]	[8.36,4.12]	[1.84,3.26]	[7.60,7.00]
Overshoot	No	No	Yes	No

Table 6 Gains of P and PI controllers for test 2

Parameters	P	PI
k_{px}, k_{py}	[300,300]	[300,500]
k_{ix}, k_{iy}	–	[2,3]

Table 7 Results of the different control schemes for test 2

Results	P	PI	F	FPI
ISE_x, ISE_y	[267.79,28.57]	[158.01,161.51]	[252.15,145.11]	[33.23,48.90]
ISE	296.36	319.52	397.26	82.13
$TS_x TS_y$	[5.08,6.44]	[2.38,5.08]	[9.98,7.66]	[2.28,6.00]
Overshoot	Yes	Yes	No	Yes

Figure 11 shows the behavior of the robot on x and y position under the different schemes. It can see how Robotino converges to desired position. Additionally, it is shown the command velocities v_x and v_y applied during execution time for each controller.

Finally, in third test, $x_{d3} = [5, 2]$. the run time is also 10 s, the gain parameters for P and PI controllers are listed in Table 8.

Table 9 contains the integral square errors, the t_s time and overshoot obtained from the execution of different controllers to go from initial position $[0,0]$ to final position $[5, 2]$. It can note that P controller had the best performance.

Figure 12 shows the behavior of the robot on the x and y position under different schemes studied, and it can see how Robotino converges to desired position. Additionally, it is shown the command velocities v_x and v_y applied during execution time for each controller.

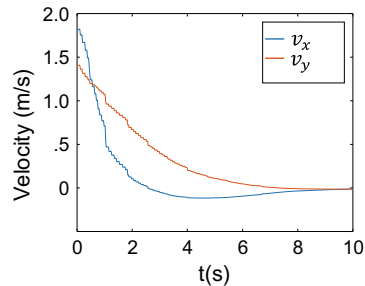
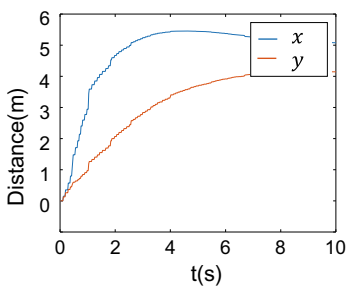
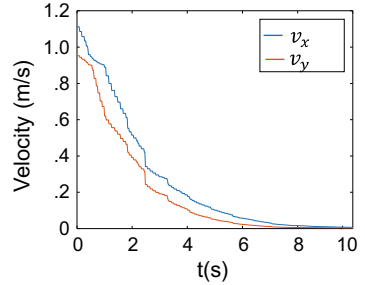
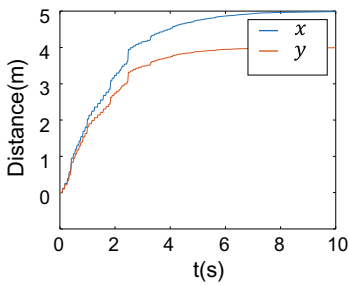
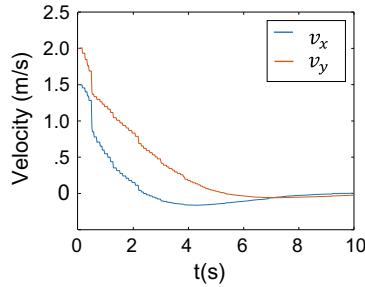
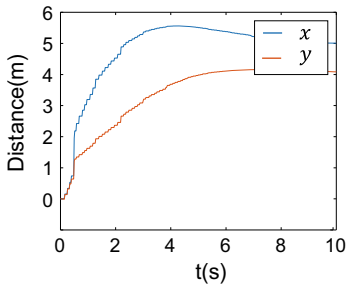
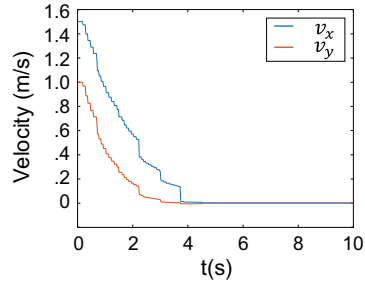
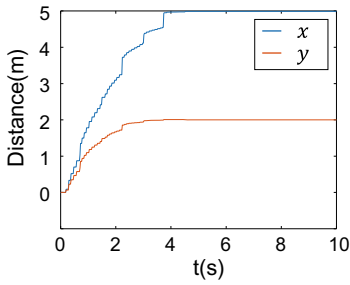


Fig. 11 Robot positions and command velocities for the different control schemes in test 2

Table 8 Gains of P and PI controllers for test 3

Parameters	P	PI
k_{px}, k_{py}	[300,500]	[500,300]
k_{ix}, k_{iy}	–	[3,2]

Table 9 Results of the different control schemes for test 3

Results	P	PI	F	FPI
ISE_x, ISE_y	[52.47, 1.62]	[184.77, 27.43]	[207.07, 26.49]	[144.41, 58.86]
ISE	54.09	212.2	233.56	203.27
$TS_x TS_y$	[4.52, 3.74]	[7.68, 6.55]	[10.00, 5.66]	[3.42, 3.44]
Overshoot	No	No	No	Yes

6 Conclusions

In paper, it can see that all models designed could arrive to desired position. Analyzing the results of the first test, it is identified that the Proportional controller presents the best stable time but it has a light error position. By the other hand, the fuzzy controller has the worst stable time but it was more accurate. But second test using same parameters Fuzzy controller was the most stable because there was not an overshoot, but timing was worst, and the fastest was FPI , the rest of the controllers reflected a very large overshoot. third test, trying to prove a more extreme scenario, the desired position in x_d and y_d was very different from each other, this generated results finding that the FPI controller obtains the best settling time with overshoot in its results, and the worst time was obtained by Fuzzy controller. In the P , PI and FPI controllers the times are mutually adjusted to achieve the required destination, while in the Fuzzy controller.

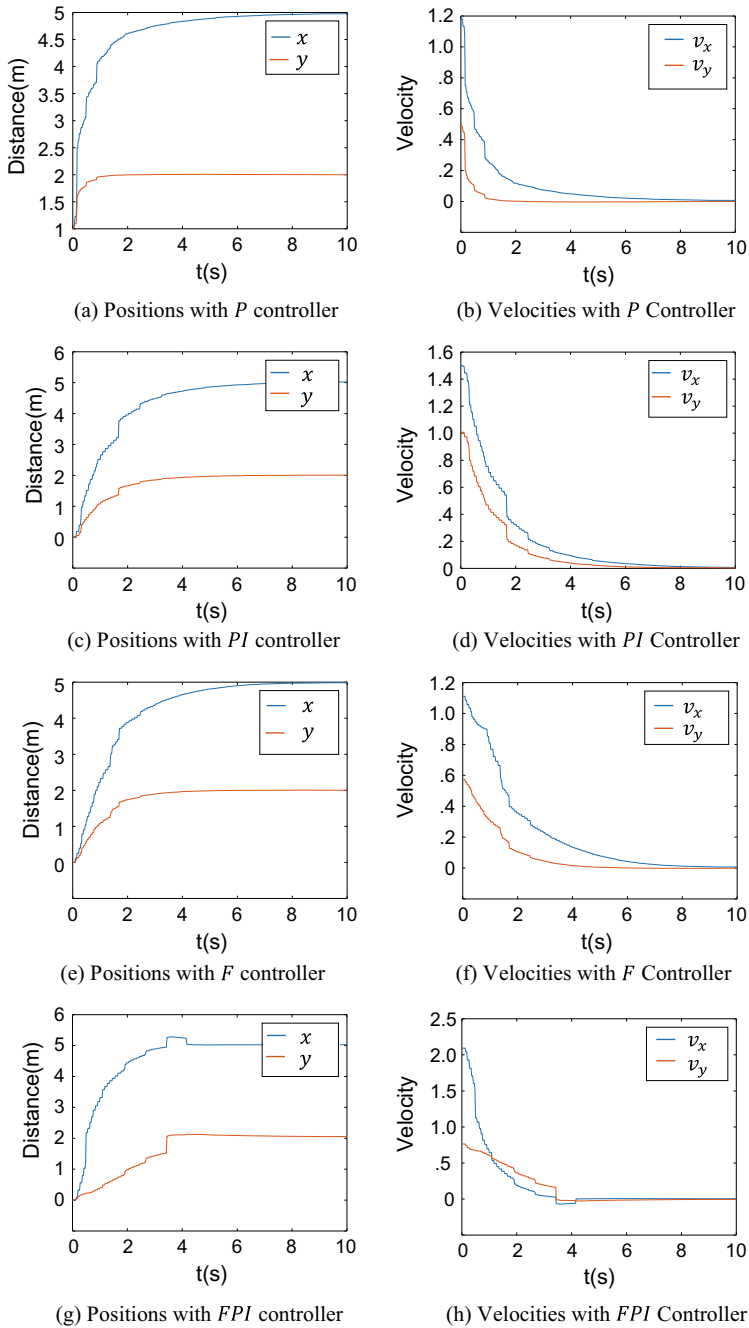


Fig. 12 Positions and command velocities for the different control schemes in test 3

Acknowledgements The authors would like to express our thankful to CONACYT, CIC-IPN and Tijuana Institute of Technology by their support to this research work.

References

1. Kermorgant, O.: A magnetic climbing robot to perform autonomous welding in the shipbuilding industry. *Robot. Comput.-Integr. Manuf.* **53**, 178–186 (2018)
2. Garnier, S., Subrin, K., Arevalo-Siles, P., Caverot, G., Furet, B.: Mobile robot stability for complex tasks in naval industries. *Proc. CIRP* **72**, 297–302 (2018)
3. Landscheidt, S., Kans, M., Winroth, M.: Opportunities for robotic automation in wood product industries: the supplier and system integrators' perspective. *Proc. Manuf.* **11**, 233–240 (2017)
4. Phuluwa, H.S., Mpofu, K.: Human-robot collaboration in a small scale rail industry: demanufacturing operations. *Proc. Manuf.* **17**, 230–237 (2018)
5. Aaltonen, I., Salmi, T., Marstio, I.: Refining levels of collaboration to support the design and evaluation of human-robot interaction in the manufacturing industry. *Proc. CIRP* **72**, 93–98 (2018)
6. Martinez, R., Castillo, O., Aguilar, L.T.: Optimization of interval type-2 fuzzy logic controllers for a perturbed autonomous wheeled mobile robot using genetic algorithms. *Inf. Sci.* (2009)
7. Abiyev, R.H., Günsela, I.S., Akkayaa, N., Aytaca, E., Çağmana, A., Abizadaa, S.: Fuzzy control of omnidirectional robot. *Proc. Comput. Sci.* **120**, 608–616 (2017)
8. Krishnaa, S., Vasub, S.: Fuzzy PID based adaptive control on industrial robot system. *Mater. Today: Proc.* **5**, 13055–13060 (2018)
9. Aly, A., Griffiths, S., Stramandinoli, F.: *Towards Intelligent Social Robots: Current Advances in Cognitive Robotics* (2015)
10. Raun, D., Zhou, C., Gupta, M.M.: Preface: fuzzy set techniques for intelligent robotic systems. *Fuzzy Sets Syst.* **134**, 1–4 (2003)
11. Forte, M.D., Correia, W.B., Nogueira, F.G., Torrico, B.C.: Reference Tracking of a Nonholonomic Mobile Robot using Sensor Fusion Techniques and Linear Control. *IFAC-PapersOnLine* (2018)
12. Singh, N.H., Thongam, K.: Mobile robot navigation using fuzzy logic in static environments. *Proc. Comput. Sci.* (2017)
13. Slim, M., Krichen, N., Masmoudi, M., Derbel, N.: Fuzzy logic controllers design for omnidirectional mobile robot navigation. *Appl. Soft Comput.* (2016)
14. Castillo, O.: *Type-2 Fuzzy Logic in Intelligent Control Applications*. Springer, Heidelberg (2012)
15. Sheikhlar, A., Fakharian, A., Adhami-Mirhosseini, A.: Fuzzy adaptive PI control of omnidirectional mobile robot. In: 13th Iranian Conference on Fuzzy Systems (IFSC) (2013)
16. Oltean, S.E., Dulău, M., Puskas, R.: Position control of Robotino mobile robot using fuzzy logic. In: IEEE International Conference on Automation (2010)
17. Huang, H.-C., Wu, T.-F., Yu, C.-H., Hsu, H.-S.: Intelligent fuzzy motion control of three-wheeled omnidirectional mobile robots for trajectory tracking and stabilization. In: 2012 International Conference on Fuzzy Theory and its Applications (iFUZZY2012) (2012)
18. Abiyeva, R.H., Günsela, I., Akkayaa, N., Aytaca, E., Çamana, A., Abizadaa, S.: Robot soccer control using behaviour trees and fuzzy logic. *Proc. Comput. Sci.* **102**, 477–484 (2016)
19. Fujita, M., Kitano, H.: *Development of an autonomous quadruped robot for robot entertainment*. *Auton. Robots* (1998)
20. Melin, P., Astudillo, L., Castillo, O., Valdez, F., Garcia, M.: Optimal design of type-2 and type-1 fuzzy tracking controllers for autonomous mobile robots under perturbed torques using a new chemical optimization paradigm. *Expert Syst. Appl.* (2013)

21. Montiel, O., Sepulveda, R., Melin, P., Castillo, O., Porta, M.A., Meza, I.M.: Performance of a simple tuned fuzzy controller and a PID controller on a DC motor. In: IEEE Symposium on Foundations of Computational Intelligence (2007)
22. Festo, Robotino. Robotino® Manual (2016). <http://www.festo-didactic.com>

Fuzzy Logic Controller with Fuzzylab Python Library and the Robot Operating System for Autonomous Robot Navigation: A Practical Approach



Eduardo Avelar, Oscar Castillo and José Soria

Abstract The navigation system of a robot requires sensors to perceive its environment to get a representation. Based on this perception and the state of the robot, it needs to take an action to make a desired behavior in the environment. The actions are defined by a system that processes the obtained information and which can be based on decision rules defined by an expert or obtained by a training or optimization process. Fuzzy logic controllers are based on fuzzy logic on which degrees of truth are used on system variables and has a rulebase that stores the knowledge about the operation of the system. In this paper a fuzzy logic controller is made with the Python fuzzylab library which is based on the Octave Fuzzy Logic Toolkit, and with the Robot Operating System (ROS) for autonomous navigation of the TurtleBot3 robot on a simulated and a real environment using a LIDAR sensor to get the distance of the objects around the robot.

Keywords Fuzzy controller · Mobile robot navigation · Obstacle avoidance

1 Introduction

The goal of autonomous mobile robotics is to build physical systems that can move without human intervention in real world environments [1]. One of the most important tasks of an autonomous system of any kind is to acquire knowledge about its environment. This is done by taking measurements using sensors and then extracting meaningful information from those measurements [2].

The controllers of the robots are the mechanism to handle the actuators based on what received by the sensors. There are many kinds of controllers for autonomous robot navigation but in this paper we use fuzzy logic controllers (FLCs) for this task. There are many works that use FLCs for robot navigation where in most of them the controller is implemented only in a simulated way [3–5], leaving the uncertainty

E. Avelar · O. Castillo (✉) · J. Soria
Tijuana Institute of Technology, Tijuana, BC, Mexico
e-mail: ocastillo@tectijuana.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_27

355

about the behavior that controller could have in a real environment, but also there are works where physical robots are used [6]. In this paper a FLC is created working in a simulate and real environment. This work pretend to be a starting point to use the fuzzylab library for creating fuzzy logic controllers in Python language for ROS, showing that it is possible create FLCs that operate successfully in real environments. In the next sections we will talk about why use fuzzy logic in controllers and how to create a basic controller for the TurtleBot3 robot.

1.1 *Many-Valued Logic*

Imagine a sensor that can detect the presence of an object up to a distance of 3 meters (m), in the absence of an object the sensor sends a voltage of 0 volts and in the presence of an object within the range it sends a voltage of 5. This is an example of two-valued or bivalent logic because only two values are obtained from the sensor. Now imagine a sensor that cannot only detect the presence of an object, but also measure the distance from the sensor, the sensor sends a voltage depending on the distance to the object. This is an example of many-valued logic where the number of values depends of the sensor resolution.

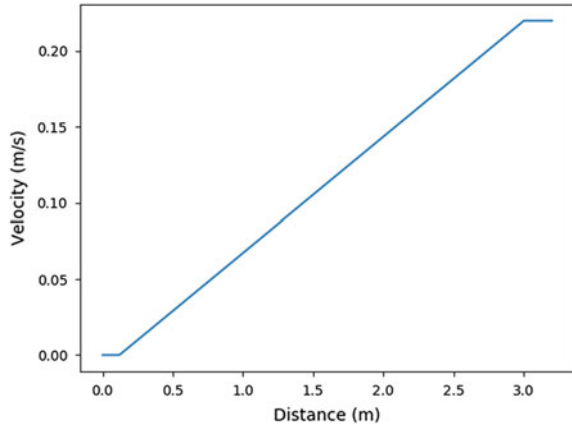
Usually, when a person talks about the distance to an object using terms such as *near* or *far*, it does not mention exactly the distance that it has to the object because it does not really know it. With the visual distance perception to an object we can determine if the object is near or far depending of our own criteria, therefore, it is relative for each person. Going back to the previous example, for a person, *near* could be a distance around and less than 0.75 m and *far* could be a distance around and more than 2.25 m, but, what happens between those values? A distance of 1.5 m is near or far? or is half near and half far?. This uncertainty can be processed and interpreted by fuzzy logic that is a form of many-valued logic.

Considering the distance sensor as the eyes of a mobile robot, we can make the robot interpret the distance as a linguistic variable with the linguistic values *near* and *far* [7] and not only numerical, bringing it closer to a more human rationing. In the “Fuzzy Logic” section we will see how to handle this linguistic variables with fuzzy logic.

1.2 *Linear and Nonlinear Systems*

Suppose that we want to control the linear velocity of a mobile robot depending of the distance to an object in front, simulating a breaking system. The robot has a maximum linear velocity of 0.22 m/s which it can go to if it is above the maximum distance that can be detect the distance sensor that is 3 m and as it gets closer to an object its speed will decrease.

Fig. 1 Velocity control with distance linear relation



The velocity (v) can be determined with a linear correspondence to the distance (d) described in the Eq. 1.

$$v = \frac{0.22}{3}d \tag{1}$$

The sensor can sense distances above 0.12m, this detects when it is out of range and the distance variable gets the value of 0 or 3 when this happens as shown in Fig. 1.

The linear behavior of the velocity in respect to the distance is a simple model of how we can determine the velocity, but, what if we want the velocity to have a different behavior in respect to the distance, maybe a smoother transition when the distance changes from 3.1 to 2.9m (see Fig. 1) for example, it is necessary to implement nonlinear models when the systems behavior cannot be modeled mathematically. Fuzzy logic allows us to create nonlinear systems depending on how the system designer wants the system to behave. In the “Fuzzy Logic” section we will see the behavior of a nonlinear system.

1.3 Fuzzy Logic

Fuzzy logic and fuzzy sets were introduced by Lofti Zadeh [8] in 1965, these are usefully to model the behavior of nonlinear systems. In fuzzy logic the linguistic values are not entirely true or false, they have some degree of membership defined by membership functions (MFs).

If we section the distance perceived by a robot using a distance sensor in the regions *near* and *far*, we can define partitions with a full membership of each linguistic value. This kind of partitions are called crisp partitions that are a zero-order uncertainty

Fig. 2 Crisp distance partitions

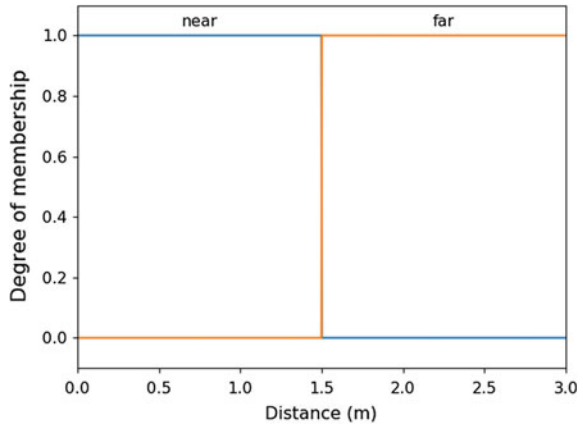
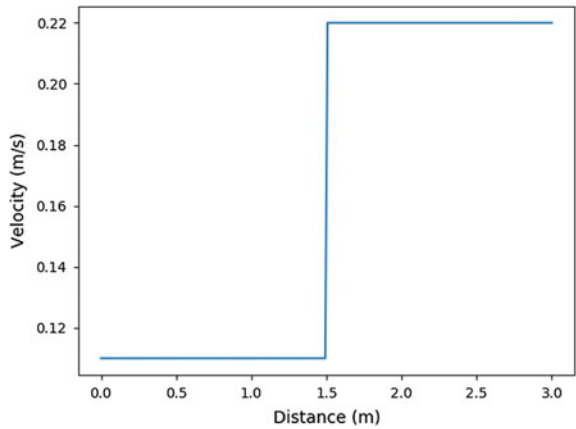


Fig. 3 Sharp transition between linguistic values



partition [9] where the degree of membership in each region is 1 as shown in Fig. 2, therefore, they do not allow any uncertainty between *near* and *far* linguistic values.

This generates a sharp transition from one term to the next. Figure 3 shows the transition from *near* to *far*.

With fuzzy logic we can define regions where the degree of membership of the regions are not always 1, this is done using MFs for each linguistic value. There are many different kinds of membership functions, in control tasks, the most used are the trapezoidal-shaped and triangular-shaped MFs. The gaussian MF generates smoother transitions but requires more computational resources but for a better smoother transition visualization we will use it for the example of velocity control. The gaussian function depends on two parameters, σ for standard deviation and center (mean) value.

Fig. 4 MFs of d (distance) = $\{near, far\}$

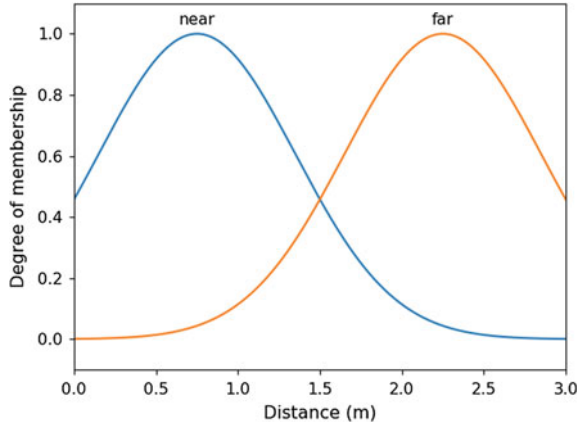
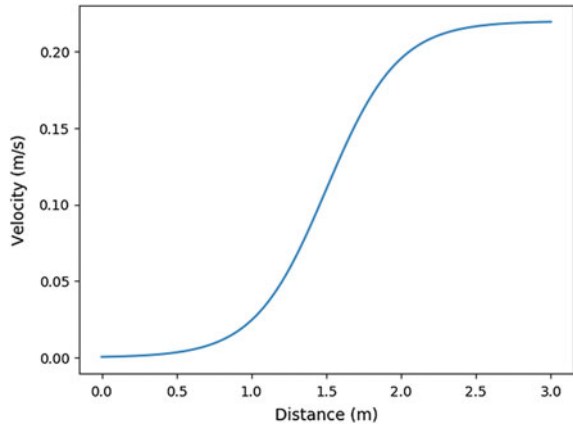


Fig. 5 Smooth transition between linguistic values



A rule-based fuzzy system contains rules, fuzzifier, inference, and output processor components. Rule antecedents are in terms of variables that can be observed or measured [9], therefore, in the velocity control example the distance is an antecedent variable.

In Fig. 4 two gaussian MFs are defined for *distance* linguistic value that form a part of the inference process and in Fig. 5 the output of the fuzzy inference system (FIS) that defines the value of the velocity is shown.

In “Design of the Fuzzy Logic Controller” section we explain how to create fuzzy systems with Python language.

2 TurtleBot3 Robot and Robot Operating System

2.1 Robot Operating System

ROS is a platform for robot software development, it is helpful to test algorithms for robot tasks like mobile robot navigation. With Gazebo software we can simulate a robot and create virtual realistic stages, that can be based on real stages. The advantage of working in a simulated environment is that we can avoid damage to the robot due to improper behaviors when testing, in our case, controllers for the autonomous navigation of a mobile robot.

2.2 TurtleBot3 Mobile Robot

TurtleBot is consider the ROS standard platform robot and we used the TurtleBot3 robot burger version that is a differential drive mobile robot, its components are shown in Fig. 6. This has a maximum linear velocity of 0.22 m/s and a maximum angular velocity of 2.84 rad/s.

The key sensor that we used is the LIDAR sensor with which we can acquire 360 distance readings (d_i for $i = 0, \dots, 359$) to the objects around it with a range of 0.12–3.5m. Based on the work of Boubertakh [10] we created 3 groups of readings to reduce the number of inputs for the robot controller designed in the next section. The first reading starts in front of the robot and taken counterclockwise. The left group SL consists of d_i for $i = 43, \dots, 48$, d_i for $i = 368, 359, 0, 1, 2$ for the front group SF and d_i for $i = 313, \dots, 318$ for the right group SR as shown in Fig. 7.

Fig. 6 TurtleBot3 burger components

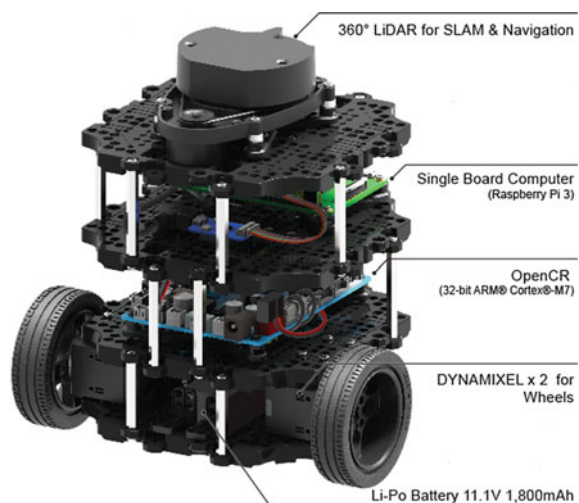
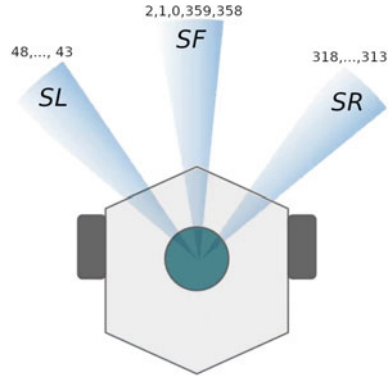


Fig. 7 Groups of sensor readings



The distances measured by the the three groups SL , SF and SR denoted by dL , dF and dR respectively are expressed as follows:

$$\begin{cases} dL = \text{mean}(d_{i=43,\dots,48}) \\ dF = \text{mean}(d_{i=2,1,0,359,358}) \\ dR = \text{mean}(d_{i=313,\dots,318}) \end{cases} \quad (2)$$

3 Design of the Fuzzy Logic Controller

There are many software tools for working with fuzzy logic. The MATLAB Fuzzy Logic Toolbox is one of the most used but this is a proprietary software so it is necessary to buy a license in order to use it. The disadvantage of sharing codes developed with proprietary software is that not everyone can replicate your work, it is necessary that the other person has a valid license of the software used and as we want for anyone to replicate those made in this paper we opted to use free software tools. The Python programming language is a interpreted language similar to MATLAB. Scikit-fuzzy is a fuzzy logic toolkit written on Python but it was not used in the experiments because it does not implement the creation of Sugeno-type fuzzy inference systems, for this reason a Python library called fuzzylab [11] based on the source code of the Octave Fuzzy Logic Toolkit [12] was developed.

3.1 Creating the FIS of the FLC

In this section we will explain step by step the creation of a fuzzy controller using the fuzzylab library for the TurtleBot3 robot. The Goal of the controller is for the robot

to navigate in a stage without hitting the walls. First a `sugfis` object is defined with the `fuzzylib` library:

```
fis = sugfis()
```

The controller has the task to determine the angular velocity depending of dL , dF and dR distances obtained by Eq. 2 that are the antecedent variables and the inputs of the FIS. To say that an object is *near* or *far* to the robot we need to define the range that the object's distance can be expected to vary. The minimum LIDAR sensor range is 0.12 m and has a considered distance reading error of 0.01 m. The input range setting for the LIDAR sensor is from 0.13 to the maximum range that is 3.5. With this information we can add an input to `fis`:

```
minr = 0.13
maxr = 3.5
fis.addInput([minr, maxr],Name='dF')
```

Now we define the membership functions for the fuzzy variable dF . Based on the work of Boubertakh [10] we use trapezoidal membership functions and define d_m , the minimum permitted distance to an obstacle and d_s , the safety distance beyond with which the robot can move at high speed. Those values were defined by experimentation, choosing those that generated an appropriate movement according to our own consideration.

```
dm = 0.3
ds = 0.7
fis.addMF('dF','trapmf',[minr, minr, dm, ds],Name='N')
fis.addMF('dF','trapmf',[dm, ds, maxr, maxr],Name='F')
```

The MF with the name `N` is the MF for *near* linguistic value and the MF with the name `F` is the MF for *far* linguistic value. Figure 8 shows the plot of the membership functions of dF using the `plotmf` function:

```
plotmf(fis,'input',0)
```

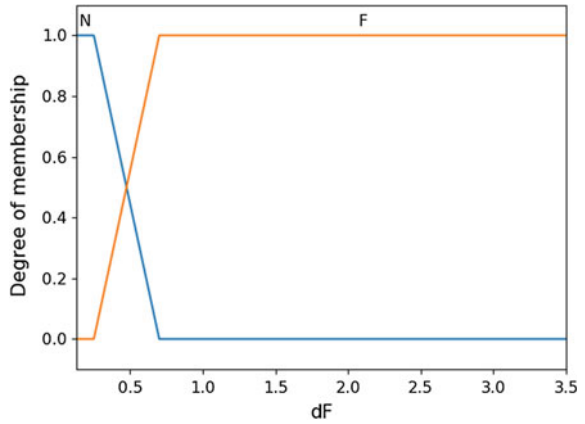
In the same way, we add dL , dR variables and the membership functions with the same parameters:

```
fis.addInput([minr, maxr],Name='dL')
fis.addMF('dL','trapmf',[minr, minr, dm, ds],Name='N')
fis.addMF('dL','trapmf',[dm, ds, maxr, maxr],Name='F')

fis.addInput([minr, maxr],Name='dR')
fis.addMF('dR','trapmf',[minr, minr, dm, ds],Name='N')
fis.addMF('dR','trapmf',[dm, ds, maxr, maxr],Name='F')
```

It is necessary to set some parameters for the robot and others for the controller for the navigation task. For the robot, we fixed the linear velocity ($0 < v \leq 0.22$)

Fig. 8 Membership functions of dF variable



with the value of 0.15 (m/s) and we define the maximum angular velocity ($0 < \omega \leq 2.84$) at which the robot can rotate with the value of 1.5 (rad/s). Those values were chosen based on the TurtleBot3 Machine Learning tutorial [13]. We add the angular velocity consequent variable to `fis` with values *NB* (*Negative Big*), *ZR* (*Zero*) and *PB* (*Positive Big*):

```
lin_vel = 0.15
min_ang_vel = 0
max_ang_vel = 1.5
```

```
fis.addOutput([min_ang_vel, max_ang_vel],Name='ang_vel')
fis.addMF('ang_vel','constant',-max_ang_vel,Name='NB')
fis.addMF('ang_vel','constant',min_ang_vel,Name='ZR')
fis.addMF('ang_vel','constant',max_ang_vel,Name='PB')
```

Suppose a simple FIS where the task is to avoid only objects in front, the robot needs to rotate to left (*PB*) or right (*NB*). If we define two simple rules saying that “If dF is *N* then ang_vel is *PB*” and “If dF is *F* then ang_vel is *ZR*”, the FIS will have the behavior to increase the angular velocity as the distance decreases as shown in Fig. 9, depending on the d_m and d_s values.

In this case there is a linear relation between dF and the angular velocity as in the case of Fig. 5, but the complexity of the FIS behavior increases when there are more than one antecedent variable because the angular velocity is determined depending of the value of dL , dF , dR and the rules defined in the FIS as shown in Fig. 10.

In the next section we will talk about the creation of the FIS rules and the aspects that were considered for its definition.

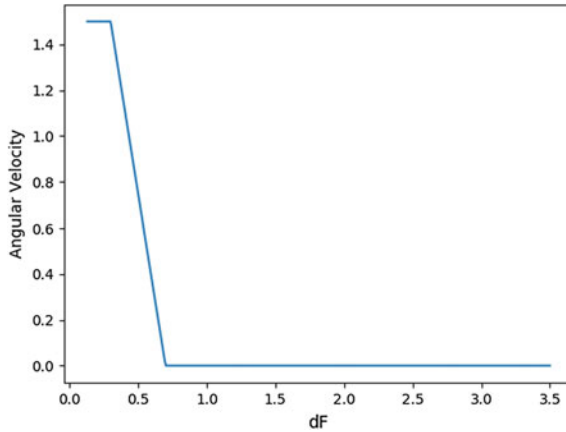


Fig. 9 The angular velocity increase as dF decreases

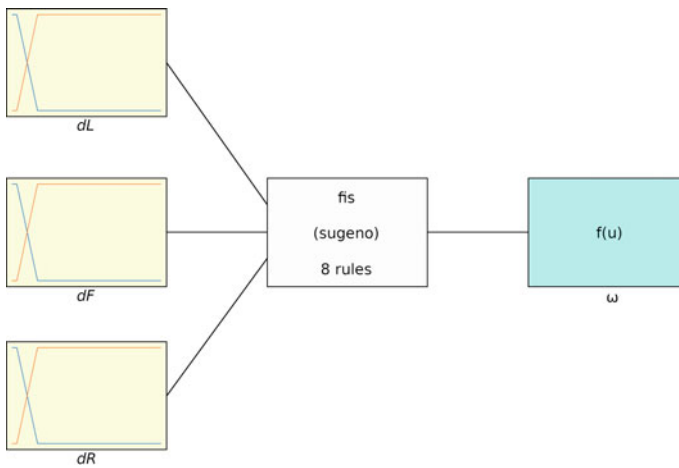


Fig. 10 Controller fuzzy inference system

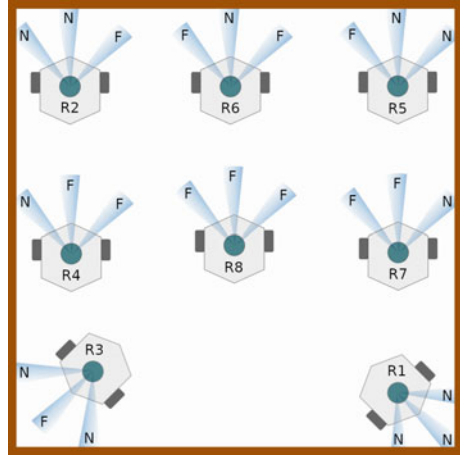
3.1.1 Rules Definition

The definition of the FIS rules require the use of expert knowledge, with these, we say how the angular velocity we want to be determined considering the different values that can be the antecedent variables.

There are 8 perceptual situations that the robot can have with three input groups and two linguistic values as shown in Fig. 11 and is associate a reaction to each of these situations defined by 8 simple rules:

- $R1 : IF(dL, dF, dR) \text{ is } (N, N, N) \text{ THEN } \omega \text{ is } NB$
- $R2 : IF(dL, dF, dR) \text{ is } (N, N, F) \text{ THEN } \omega \text{ is } NB$

Fig. 11 Different perceptual situations with three inputs and two linguistic values



- R3 : IF(dL, dF, dR) is (N, F, N) THEN ω is ZR*
- R4 : IF(dL, dF, dR) is (N, F, F) THEN ω is NB*
- R5 : IF(dL, dF, dR) is (F, N, N) THEN ω is PB*
- R6 : IF(dL, dF, dR) is (F, N, F) THEN ω is NB*
- R7 : IF(dL, dF, dR) is (F, F, N) THEN ω is PB*
- R8 : IF(dL, dF, dR) is (F, F, F) THEN ω is ZR*

The rules 1 and 6 can be *NB* (turn right) or *PB* (turn left), depending on the behavior we want the robot to have when it is close to an object in front and *dL* and *dR* are in the same state. In the fuzzylab library the rules are defined in form of lists using membership function indices:

```
ruleList = [[0, 0, 0, 0, 1, 1], # Rule 1
            [0, 0, 1, 0, 1, 1], # Rule 2
            [0, 1, 0, 1, 1, 1], # Rule 3
            [0, 1, 1, 0, 1, 1], # Rule 4
            [1, 0, 0, 2, 1, 1], # Rule 5
            [1, 0, 1, 0, 1, 1], # Rule 6
            [1, 1, 0, 2, 1, 1], # Rule 7
            [1, 1, 1, 1, 1, 1]] # Rule 8
```

The first columns specifies input membership function indices, the following specifies output membership function indices, the penultimate the rule weight and the last the antecedent fuzzy operator, where 1 specifies the “and” operator. The rules are added to *fis* with the *addRule* method:

```
fis.addRule(ruleList)
```

With all the previous steps a FIS is created with the fuzzylab library, we only need to determine *dL*, *dF* and *dR* from the sensor readings and evaluate these values in the *fis* with the *evalfis* function. This will be explained in the next section.

3.2 The Fuzzy Logic Controller

A controller has the task to determine the actions to take based on the perception of some sensor to resolve some problems. In our case, the controller has the task to calculate the angular velocity at which the robot needs to go based on the readings of the LIDAR distance sensor in order to not hit an object. This task is carried out mostly by the FIS created in the previous section, but part of the controller tasks is to process the sensor readings and to reflect their actions in the system, for this, ROS offers an easy API to realize those actions. To get the readings of the sensor and define dL , dF and dR values we use:

```
distances = rospy.wait_for_message('scan', LaserScan)

dL = mean(distances[43:48])
dF = mean(distances[:3] + distances[-2:])
dR = mean(distances[313:318])
```

Once the distances are determined, the new angular velocity is calculated from the FIS and reflects the result by publishing the value using the ROS functions:

```
new_ang_vel = evalfis(fis, [dL, dF, dR])

twist = Twist()
twist.linear.x = lin_vel
twist.angular.z = new_ang_vel

cmd_pub = rospy.Publisher('cmd_vel', Twist, queue_size=1)
cmd_pub.publish(twist)
```

It is necessary that the robot is initialized to receive the updates:

```
roslaunch turtlebot3_bringup TurtleBot3_robot.launch
```

More detailed documentation can be found in the ‘TurtleBot3 Fuzzy Logic Controllers with Python’ tutorial [14], this contains information about the necessary configurations to be made to test the controller in the robot, simulated and physical.

4 Experiments and Results

A real stage based on the stage 1 of the TurtleBot3 ML tutorial [13] was created, that is a 4×4 map with no obstacles as shown in Fig. 12a. In first instance the stage was made with black walls as shown in Fig. 12c but when bad readings were observed as seen in Fig. 12d they were painted white having better readings as shown in Fig. 12b.

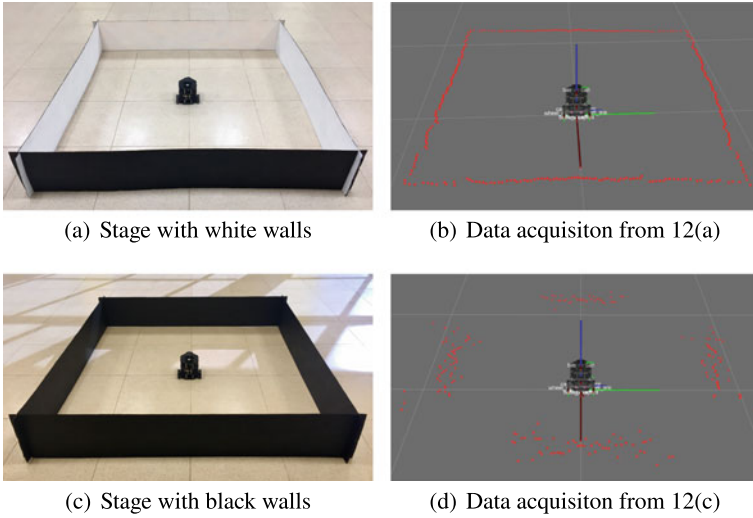


Fig. 12 Physical stage. Data acquisition is better in (a) than (c)

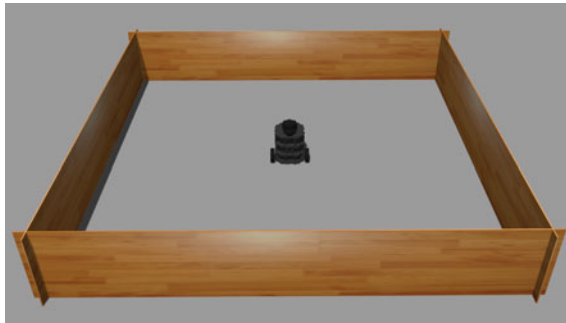


Fig. 13 Virtual Gazebo stage

This stage was made in the Gazebo simulator as a shown in Fig. 13 with the same size of the physical stage, enabling the first controller tests in a simulated environment to then see the behavior that they have in the real stage.

The controller worked correctly in simulated and real environments, causing the robot to move on stage without hitting the walls, both in the simulated environment shown in Fig. 14a and in the real environment shown in Fig. 14b. Different behaviors can be observed manipulating d_m and d_s values. A lower d_m value makes that the robot has a closer approximation to the objects.



(a) Behavior of the robot in the simulated stage. (b) Behavior of the robot in the real stage.

Fig. 14 Behavior of the robot in the simulated and real stage

5 Conclusions and Future Research

The FLC created with the fuzzylab library works correctly in the obstacle avoidance task with the TurtleBot3 robot. In future work more complex controllers can be designed to work in more complex stages, implementing optimization algorithms and evaluating the efficiency in some tasks. For the simplicity of the stage created, the manipulation of the linear velocity was not considered but in more complex stages the determination of the linear velocity could be considered.

Reinforcement learning (RL), an area of machine learning, is a computational approach to understanding and automating goal-directed learning and decision making [15]. Many of the RL algorithms such as Q-Learning [16] have been used to optimize fuzzy logic controllers, starting with the adaptation of the Q-learning algorithm for fuzzy inference systems by Glorennec [17] and Berenji [18] and more recent works [10, 19, 20] for control robot navigation. For these reasons, its use in future works with use of the fuzzylab library and the ROS platform has been contemplated.

References

1. Saffiotti, A.: The uses of fuzzy logic in autonomous robot navigation. *Soft Comput.* **1**, 180–197 (1997)
2. Siegwart, R., Nourbakhsh, I.R., Scaramuzza, D.: *Introduction to Autonomous Mobile Robots*, 2nd ed. The Mit Press (2011)
3. Reignier, P.: Fuzzy logic techniques for mobile robot obstacle avoidance. *Robot. Auton. Syst.* **12**(3), 143–153 (1994)
4. Duan, Y., Xin-Hexu: Fuzzy reinforcement learning and its application in robot navigation, vol. 2, pp. 899–904 (2005)
5. Raguraman, S.M., Tamilselvi, D., Shivakumar, N.: Mobile robot navigation using fuzzy logic controller. pp. 1–5 (2009)
6. Faisal, M., Hedjar, R., Sulaiman, M.A., Al-Mutib, K.: Fuzzy logic navigation and obstacle avoidance by a mobile robot in an unknown dynamic environment. *Int. J. Adv. Robot. Syst.* **10**(1), 37 (2013)
7. Zadeh, L.A.: Fuzzy logic = computing with words. *IEEE Trans. Fuzzy Syst.* **4**, 103–111 (1996)
8. Zadeh, L.A.: Fuzzy sets. *Inf. Control.* **8**, 338–353 (1965)

9. Mendel, J.M.: *Uncertain Rule-Based Fuzzy Systems: Introduction and New Directions*, 2nd ed. Springer (2017)
10. Boubertakh, H., Tadjine, M., Glorennec, P.-Y.: A new mobile robot navigation method using fuzzy logic and a modified q-learning algorithm. *J. Intell. Fuzzy Syst.* **21**, 113–119 (2010)
11. Fuzzylab, a python fuzzy logic library. <https://github.com/ITTCs/fuzzylab>. Accessed 21 March 2019
12. Octave fuzzy logic toolkit. <https://sourceforge.net/projects/octave-fuzzy/>. Accessed 21 March 2019
13. Turtlebot3 ml. http://emanual.robotis.com/docs/en/platform/turtlebot3/machine_learning/. Accessed 21 March 2019
14. Fuzzy logic controllers with python for turtlebot3 robot. <https://github.com/eavelardev/turtlebot3 FLC>. Accessed 21 March 2019
15. Sutton, R.S., Barto, A.G.: *Reinforcement Learning: An Introduction*, 2nd ed.. The Mit Press (2018)
16. Watkins, C.J.C.H., Dayan, P.: Q-learning. *Mach. Learn.* **8**, 279–292 (1992)
17. Glorennec, P.Y., Jouffe, L.: Fuzzy q-learning. In: *Proceedings of 6th International Fuzzy Systems Conference*, vol. 2, pp. 659–662 (1997)
18. Berenji, H.R.: Fuzzy q-learning: a new approach for fuzzy dynamic programming **1**, 486–491 (1994)
19. Cherroun, L., Boumehraz, M.: Intelligent systems based on reinforcement learning and fuzzy logic approaches. *Appl. Mob. Robot.* 1–6 (2012)
20. Cherroun, L., Boumehraz, M., Kouzou, A.: Mobile robot path planning based on optimized fuzzy logic controllers. 255–283 (2019)

Neural Networks Applications

Neural Evolutionary Predictive Control for Linear Induction Motors with Experimental Data



Alma Y. Alanis, Nancy Arana-Daniel, Carlos Lopez-Franco and Jorge D. Rios

Abstract Model Predictive Control is a well-suited control strategy; however, it needs to solve two main problems in order to be applied, first, it is necessary to have a suitable model of the plant to be controlled, in order to allow an adequate prediction, second, it is necessary to solve an optimization problem. These two problems not always can be solved particularly for nonlinear complex systems. Therefore, in this chapter we propose two variations for Model Predictive Control, in a first stage a recurrent high order neural network is proposed to obtain a fitting model for the plant to be controlled, and, at the same time this neural model identifies the system on-line through available measurements. Then, in a second stage, the optimization problem is solved using a particle swarm optimization algorithm. Using these two modifications, it is proposed a Neural Evolutionary Predictive Control for discrete-time nonlinear systems under disturbances under disturbances, and its effectiveness is shown in the experimental results by using data obtained from a linear induction motor prototype.

Keywords Model predictive control · Neural evolutionary predictive control · Linear induction motors · Experimental data · Kalman filter learning

1 Introduction

Model Predictive Control (MPC) has been widely used in the academy as well as in the industry, as a useful way to solve multivariable constrained control problems, these ideas were originated on 1960 [1]. MPC is a control strategy that predicts the future behavior of the controlled system over a finite time horizon, in order to compute an optimal control input that minimizes a priory defined cost functional [2]. The control input is calculated by solving, at each control instant, a finite horizon open-loop optimal control problem; the first part of the resulting optimal input trajectory is then

A. Y. Alanis (✉) · N. Arana-Daniel · C. Lopez-Franco · J. D. Rios
CUCEI, Universidad de Guadalajara, Blvd. Marcelino Garcia Barragan 1500, Col. Olimpica,
Guadalajara, Jalisco, C.P 44430, Mexico
e-mail: almayalanis@gmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_28

373

applied to the system until the next control instant, at which the horizon is shifted. The whole procedure is repeated again [2]. MPC suffers of several problems, stability, performance and feasibility of the on-line optimization, particularly for non-linear systems, although there has been progress to solve these problems [2], for practical applications many questions still remain open, including the efficiency and reliability of the on-line computation scheme. Another important issue that remains open in MPC is its robustness to model uncertainty, noise and delays.

In [3], it is proposed a model predictive neural control with a differential neural network, if well such proposal solves the need of previous knowledge of a model to predict the behavior of the system under the control actions, the optimization problems still remain open, besides the implementation requires hard computation doing difficult to be implemented in real-life sceneries. [4], proposes a Neural Predictive Control Design for Uncertain Nonlinear Systems with a NARMAX structure to describe an unknown system and a neural model used to learn the systems. In [5], it is proposed a neural network model predictive control of nonlinear systems using genetic algorithms, this scheme is used to solve the control problem of a linear system; [6] proposes a neural network model predictive controller, the control law is represented by a neural network function approximator. In [7], it is developed a Robust Model Predictive Control Using a Discrete-Time Recurrent Neural Network applied for a linear system. [8], proposes an adaptive predictive control using neural network for a class of pure-feedback systems in discrete time; [9], develops a robust model predictive control of nonlinear systems with unmodelled dynamics and bounded uncertainties based on neural networks; [10], presents Neural Networks for Model Predictive Control for a crystallizer dynamic simulator. [11], presents an adaptive neural network model predictive control. If it is well-known that neural network model predictive controllers have demonstrated high potential in the non-conventional branch of nonlinear control, however, they have a main issue, all the reviewed works only deal with numerical results, this is mainly due to the proposed solutions are usually computational very intensive. This issue provoke that neural predictive controllers cannot be implemented in real-time.

On the other hand, optimization problem remains open due to complications to solve nonlinear optimization problems in real time, in this way in [12] proposes a particle swarm optimization (PSO) algorithm with simulated annealing technique to obtain optimal future control inputs in MPC for a discrete time nonlinear process applied in simulation to a thermal power unit load system, the prediction is obtained through a complex radial basis neural network trained off-line. While the idea of use neural networks to obtain the prediction is excellent, the proposed structure has an inherit high computational complexity doing its experimental implementation impossible.

Therefore, taking in mind all these previous works and its difficulties to be experimentally implemented, in this chapter we propose the use of a Recurrent High Order Neural Network (RHONN), which has been prove to be reliable for real-time implementations due to its capability to be trained on-line and its need of few units to develop complex applications [13–15].

Then the main contribution of this chapter is modifying the classical MPC, in two stages: In the first stage a RHONN trained with an Extended Kalman Filter (EKF) is used to model an unknown discrete-time nonlinear system, besides, this Neural model allow to predict the future output behavior subject to constraints on the manipulated inputs and outputs without the need of use an explicit model of the plant to be controlled. In the second stage, a PSO algorithm is used to solve the optimal control problem without the need of finding an explicit optimization solution. These modifications allow the experimental implementation of the predictive controller for unknown discrete-time nonlinear systems under external and internal disturbances, in presence of noise.

2 Neural Model

As it was explained in Sect. 1, the proposed methodology includes two main variations to classic MPC: in the first stage the model is obtained through data processing instead of use an academic model of the plant to be controlled, then the model is obtained using a RHONN, the second variation, about the optimization problem solver is going to be treated in the next section. Through this chapter, sub-index k is used as the sampling time, with $k \in \{0\} \cup \mathbb{Z}^+$. Let, consider the affine discrete-time nonlinear system.

$$\chi_{k+1} = f(\chi_k) + g(\chi_k)u_k \tag{1}$$

where $\chi_k \in \mathbb{R}^n$ is the state of the system, $u_k \in \mathbb{R}^m$ is the control input and $f(\chi_k) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ and $g(\chi_k) : \mathbb{R}^n \rightarrow \mathbb{R}^{n \times m}$ are smooth mappings. Without loss of generality, system (1) is supposed to have an equilibrium point at $\chi_k = 0$.

The discrete-time RHONN employed for identification of the nonlinear system (1) is defined as

$$x_{i,k+1} = w_i Z_i(x_k, u_k), \quad i = 1, \dots, n \tag{2}$$

where x_i is the state of the i th neuron, w_i is the respective online adapted weight vector, n is the system state dimension and $Z_i(x_k, u_k)$ is given by:

$$Z_i(x_k, u_k) = \begin{bmatrix} Z_{i_1} \\ Z_{i_1} \\ \vdots \\ Z_{i_{L_i}} \end{bmatrix} = \begin{bmatrix} \prod_{j \in I_1} y_{i_j}^{d_{i_j}(1)} \\ \prod_{j \in I_2} y_{i_j}^{d_{i_j}(2)} \\ \vdots \\ \prod_{j \in I_{L_i}} y_{i_j}^{d_{i_j}(L_i)} \end{bmatrix} \tag{3}$$

with L_i being the respective number of high-order connections, I_1, I_2, \dots, I_{L_i} is a collection of non-ordered subsets of dimension $1, 2, \dots, n + m$ is the number of external inputs, d_{i_j} is a nonnegative integer and y_i is defined as follows:

$$y_i = \begin{bmatrix} y_{i_1} \\ \vdots \\ y_{i_n} \\ y_{i_{n+1}} \\ \vdots \\ y_{i_{n+m}} \end{bmatrix} = \begin{bmatrix} S(x_1) \\ \vdots \\ S(x_n) \\ u_1 \\ \vdots \\ u_m \end{bmatrix} \tag{4}$$

In (4), $u_k = [u_1, u_2, \dots, u_m]^T$ is the input vector to the RHONN and $S(\cdot)$ is defined by

$$S(x) = \tanh(\beta x) \tag{5}$$

with β a positive constant.

The neural weights are updated using an EKF-based training algorithm, described by the following set of equations:

$$\begin{aligned} w_{i,k+1} &= w_{i,k} + \eta_i K_{i,k} e_{i,k} \\ K_{i,k+1} &= P_{i,k} H_{i,k} [R_i + H_{i,k}^T P_{i,k} H_{i,k}]^{-1} \\ P_{i,k+1} &= P_{i,k} - K_{i,k} H_{i,k}^T P_{i,k} + Q_i \end{aligned} \tag{6}$$

with

$$e_{i,k} = \chi_{i,k} - x_{i,k} \tag{7}$$

where w_i is the i th weight vector, e_i is the neural identification error, $P_i \in \mathbb{R}^{L_i \times L_i}$ is the weight estimation error covariance matrix, L_i is the respective number of neural network weights, $\chi \in \mathbb{R}^n$ is the plant vector state, $x \in \mathbb{R}^n$ is the neural network output, $K_i \in \mathbb{R}^{L_i \times n}$ is the Kalman gain matrix, $Q_i \in \mathbb{R}^{L_i \times L_i}$ is the neural network weight estimation noise covariance matrix, $R_i \in \mathbb{R}^{n \times n}$ is the measurement noise covariance. As additional parameter, the rate learning η_i is introduced in (6), such that $0 < \eta_i \leq 1$. Usually $P_{i,k}$, Q_i and $R_{i,k}$ are initialized as diagonal matrices. Finally, $H_i \in \mathbb{R}^{L_i \times n}$ is a matrix of the derivatives of the network outputs with respect to all trainable weights as follows

$$H_{i,k} = \left[\frac{\partial x_{i,k}}{\partial w_{i,k}} \right]^T \tag{8}$$

Then (2), is the model to be used to construct the Neural Predictive Control instead of model (1), some of the advantages of the use of model (2) include: (1) the structure of model (2) can be defined according to the implementations needs, (2) the model (2) includes all the disturbances, unmodelled dynamics, saturations, noises, parametric variations, time-varying modifications, among others due to it is computed on-line through an EKF-training algorithm with measured information, (3) RHONN model (2) always provides an accurate model for the nonlinear plant due to the RHONN is trained on-line with experimental data. Stability proof of RHONN trained with an EKF algorithm has been demonstrated on [13].

3 Particle Swarm Optimization

As explained in Sect. 1, the proposed methodology includes two main variations to classic MPC: the first one was presented in Sect. 2, and the remaining variation is implemented in a second stage of the proposed approach, in which an evolutionary algorithm is used to solve the optimization problem of the MPC. The objective of optimal control theory is to compute the control law which will force a process to fulfill physical restrictions and at the same time to optimize a performance criterion [16]. To solve the above mentioned, unfortunately it is required to solve the associated Hamilton Jacobi Bellman (HJB) equation, which is not an easy task, however, recently optimization research has proposed new strategies to deal with optimization of non-linear systems with multiple variables, multiple objective functions and under restrictions, one of these alternatives is the use of evolutionary techniques to solve this kind of problems that are hard to solve through traditional optimization techniques. In this chapter it is proposed the use of Particle Swarm Optimization (PSO) algorithm, this modification allows to solve complex nonlinear optimization problems. The PSO algorithm consists of an iterative adaptation of set of multidimensional vectors called particles that communicate information among them as a swarm which provides a set of candidate solutions for an objective function. This objective function is normally multi-objective, multimodal and high dimensional [17]. The mathematical model of the described algorithm reads as follows

$$V_{ij,k+1} = V_{ij,k} + c_1 \mathbf{R}_1 (p_{ij,k} - X_{ij,t}) + c_2 \mathbf{R}_2 (p_{gj,k} - X_{ij,t}) \quad (9)$$

$$X_{ij,k+1} = X_{ij,k} + V_{ij,t} \quad (10)$$

where $i = 1, 2, \dots, s$ is the i th particle of a swarm that satisfies $s \in \mathbb{R}^D$, and $j = 1, 2, \dots, D$ is the j th element of dimension problem D . Also, k represents the iteration counter, \mathbf{R}_1 and \mathbf{R}_2 are random, normalized and uniformly distributed values. c_1 and c_2 represent the social and cognitive parameter, $X_{ij,k}$ is the particle ij position for the iteration k ; $X_{ij,k+1}$ is the particle ij position for $k + 1$ iteration, $V_{ij,k}$ is the particle ij velocity for k iteration. $p_{ij,k}$ represents the local best position for

particle ij in iteration k and $p_{gj,k}$ represents the global best position for entire swarm in iteration k . In order to improve the PSO algorithm performance, an inertial weight vector is introduced in a variation of PSO [16]; this variation, introduces a dynamic constriction factor for velocity vectors in order to concentrate the swarm around the most promising solution. Resulting in the following equations:

$$V_{ij,k+1} = \omega * V_{ij,k} + c_1 \mathbf{R}_1(p_{ij,k} - X_{ij,t}) + c_2 \mathbf{R}_2(p_{gj,k} - X_{ij,t}) \quad (11)$$

where ω represents the inertial weight factor; this quotient can either be pre-defined as a constant value with previous knowledge of the problem, or can be adjusted dynamically as a decreasing linear function as follows:

$$\omega_k = \omega_{up} - \frac{(\omega_{up} - \omega_{low})}{T_{max}}k \quad (12)$$

where ω_k is the time-dependent inertial weight, ω_{up} is the inertial weight upper boundary, ω_{low} is the inertial weight lower boundary. The iteration counter is represented by k , T_{max} is the iteration total.

The PSO algorithm used in this chapter, can be defined as follows [18]:

1. Initialize a population of particles with random positions and velocities in the problem space.
2. For each particle, evaluate the desired optimization fitness function.
3. Compare the particles fitness evaluation with the particles p_{id} , if the current value is better than the p_{id} , then set p_{id} value equal to the current location.
4. Compare the best fitness evaluation with the population's overall previous best. If the current value is better than the p_{gd} , then set p_{gd} to the particle's array and index value.
5. Update the particle's velocity and position as follows:

The velocity of the i th particle of d dimension is given by (11)

The position vector of the i th particle of D dimension is updated by (10)

6. Repeat the step 2 until a criterion is met, usually a sufficiently good fitness or a maximum number of iterations or epochs.

In case the velocity of the particle exceeds V_{max} (the maximum velocity for the particles) then it is reduced to V_{max} . Thus, the resolution and fitness of search depends on the V_{max} . If V_{max} is too high, then particles will move in larger steps and so the solution reached may not be the as good as expected. If V_{max} is too low, then particles will take a long time to reach the desired solution [18]. Due the above explained, PSO are very suitable models of noisy problems, as the one we are considering, besides PSO has shown good results in optimization problems [18] it will be used to optimize complex optimization problems instead of user solutions. In order to carry out the estimation of parameters using metersticks, the fitness function plays an important role [17].

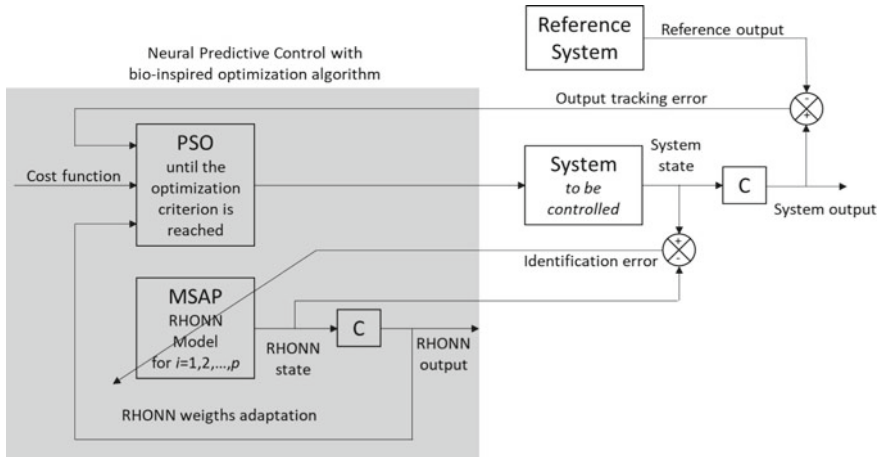


Fig. 1 Neural predictive control with bio-inspired optimization algorithm

4 Neural Evolutionary Predictive Control

In this section it is presented the proposed neural predictive control with a PSO algorithm, the whole scheme is presented in Fig. 1, and it is composed by the following elements:

1. An unknown discrete-time nonlinear system, that can include: disturbances (internal and external), noises, saturations, unknown delays, unknown dynamics, etc.
2. A reference system
3. A RHONN trained on-line with an extended Kalman filter, to model the system allowing configuring as a MSAP
4. A PSO algorithm to compute the optimal control law over a fixed future horizon.

5 Experimental Results

This section presents the experimental implementation of the Neural Evolutionary Predictive Control proposed in Sect. 4.

5.1 Linear Induction Motor Model

Even, if in this work the plant model is considered unknown, it is presented only for completeness of the readership. Linear induction motors (LIM) is a special electrical

machine, in which the electrical energy is converted directly into mechanical energy of translation motion. After 1980, LIM found their first noticeable applications in, among others, transportation, industry, automation, and home appliances [19, 20]. LIM has many excellent performance features such as high-starting thrust force, elimination of gears between motor and motion devices, reduction of mechanical losses and the size of motion devices, high speed operation, silence, and so on [19, 21]. The driving principles of the LIM are similar to the traditional rotary induction motor (RIM), but its control characteristics are more complicated than the RIM, and the parameters are time varying due to the change of operating conditions, such as speed, temperature, and rail configuration.

The α - β model of a LIM discretized by the Euler technique [22–24] can be written as follows:

$$\begin{aligned}
 q_{m,k+1} &= q_{m,k} + Tv_k \\
 v_{k+1} &= (1 - k_2T)v_k - k_1T\lambda_{r\alpha,k}\rho_1i_{s\alpha,k} - k_1T\lambda_{r\beta,k}\rho_2i_{s\beta,k} - k_1T\lambda_{r\beta,k}\rho_1i_{s\beta,k} - k_3TF_L \\
 \lambda_{r\alpha,k+1} &= (1 - k_6T)\lambda_{r\alpha,k} + k_4Tv_k\rho_1i_{s\alpha,k} - k_4T\rho_2i_{s\beta,k} - k_4Tv_k\rho_2i_{s\beta,k} + k_5T\rho_1i_{s\beta,k} \\
 \lambda_{r\beta,k+1} &= (1 - k_6T)\lambda_{r\beta,k} + k_4Tv_k\rho_2i_{s\alpha,k} - k_4T\rho_1i_{s\beta,k} + k_4Tv_k\rho_1i_{s\beta,k} + k_5T\rho_2i_{s\beta,k} \\
 i_{s\alpha,k+1} &= (1 + k_9T)i_{s\alpha,k} - k_7T\lambda_{r\alpha,k}\rho_2 - k_8T\lambda_{r\alpha,k}v_k\rho_1 + k_7T\lambda_{r\beta,k}\rho_1 \\
 &\quad - k_8T\lambda_{r\beta,k}v_k\rho_2 - k_{10}Tu_{\alpha,k} \\
 i_{s\beta,k+1} &= (1 + k_9T)i_{s\beta,k} - k_7T\lambda_{r\alpha,k}\rho_1 + k_8T\lambda_{r\alpha,k}v_k\rho_2 - k_7T\lambda_{r\beta,k}\rho_2 \\
 &\quad - k_8T\lambda_{r\beta,k}v_k\rho_1 - k_{10}Tu_{\beta,k}
 \end{aligned} \tag{13}$$

with

$$\begin{aligned}
 \rho_1 &= \sin(n_p q_{m,k}), \quad \rho_2 = \cos(n_p q_{m,k}), \quad k_1 = \frac{n_p L_{sr}}{D_m L_r}, \quad k_2 = \frac{R_m}{D_m}, \\
 k_3 &= \frac{1}{D_m}, \quad k_4 = n_p L_{sr}, \quad k_5 = \frac{R_r L_{sr}}{L_r}, \quad k_6 = \frac{R_r}{L_r}, \quad k_7 = \frac{L_{sr} R_r}{L_r (L_{sr}^2 - L_s L_r)} \\
 k_8 &= \frac{L_{sr} n_p}{L_{sr}^2 - L_s L_r}, \quad k_9 = \frac{L_r^2 R_s + L_{sr}^2 R_r}{L_r (L_{sr}^2 - L_s L_r)}, \quad k_{10} = \frac{L_r}{L_{sr}^2 - L_s L_r}
 \end{aligned}$$

where $q_{m,k}$ is the position, v_k is the linear velocity, $\lambda_{r\alpha,k}$ and $\lambda_{r\beta,k}$ are the α -axis and β -axis secondary flux, respectively, $i_{s\alpha,k}$ and $i_{s\beta,k}$ are the α -axis and β -axis primary current, respectively, $u_{\alpha,k}$ and $u_{\beta,k}$ are the α -axis and β -axis primary voltage, R_s is the winding resistance per phase, R_r is the secondary resistance per phase, L_{sr} is the magnetizing inductance per phase, L_s is the primary inductance per phase, L_r is the secondary inductance per phase, F_L is the load disturbance, D_m is the viscous friction and iron-loss coefficient and n_p is the number of poles pairs, T is the sample period [23].

5.2 Linear Induction Motor Prototype

The LIM prototype used to perform the parametric identification by means of the PSO algorithm, it is depicted in Fig. 1 and it consists of the following elements (Fig. 2).

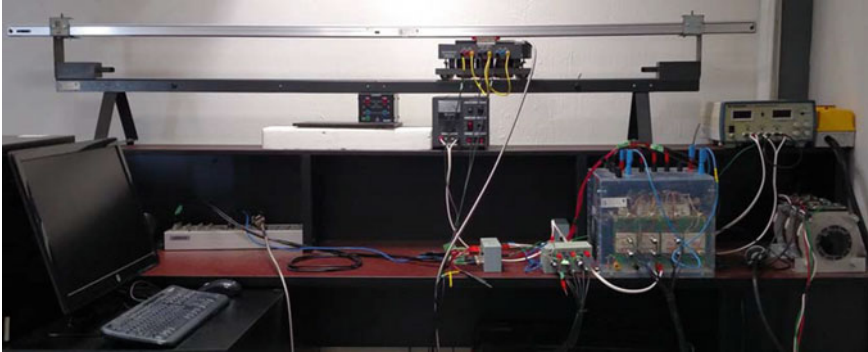


Fig. 2 LIM benchmark

- The prototype computer has MATLAB[®]/Simulink installed and the dSPACE[®] ControlDesk software and the DS1104 libraries, this software forms the interface between the computer and the dSPACE[®] DS1104 card. The MATLAB[®]/Simulink models are compiled with the dSPACE software to obtain the files that are loaded on the controller card, and provide functions to monitor the signals in real time.
- The dSPACE[®] DS1104 controller card is designed for the implementation of high-speed multivariable controllers and real-time simulations.
- The RTI1104 connector panel provides access to external input signals (not direct from the computer) and output from the card.
- The Linear Induction Motor is the LAB-Volt[®] 8228 model consisting of a moving vehicle and a stationary rail. The mobile vehicle has four metal wheels and the stator. Currents α and β are obtained from the actual LIM currents converted to the model α and β , the position and velocity are obtained from the linear precision encoder SENC[®] 150 that is mounted on the LIM.
- The inputs to the LIM are a frequency-varying signal provided by the autotransformer and a PWM (pulse-width modulation) signal, both signals passing through the Power module that allows the handling of the voltages and currents necessary to move the stator of the LIM.

5.3 RHONN as N-Step Ahead Predictor

There are two common strategies for multiple step ahead prediction (1) the iterated prediction and (2) direct prediction. For n step ahead prediction: the first method iterates n times a one step ahead model whereas the second method trains the model for generate a direct prediction at time $k + n$ [25, 26]. It is not clear which method is better; however, both methods present the problem of propagating the error when the prediction horizon time increases, in consequence the accuracy of predictions decreases. One solution to this problem is to implement on-line learning techniques

to repeatedly adjust the parameters of the model with recent information, including the last real values observed [25]. An on-line learning algorithm allows to predict in each on-line learning to improve the performance of the model. The EKF training algorithm allows on-line implementation for adjusting network weights. In this work we focus on n step ahead iterated prediction and we prioritize identification to obtain an accurate model with neural networks, and then used it as a predictor.

Multiple step ahead prediction can be carried out recursively relating $\hat{y}_{k+n|k}$ with $\hat{y}_{k+n-1|k}$, $\hat{y}_{k+n-2|k}$, $\hat{y}_{k+n|k}$, \dots , y_k , y_{k-1} , $\hat{u}_{k+n-1|k}$, $\hat{u}_{k+n-2|k}$, \dots , u_k , u_{k-1} . The values of $\hat{y}_{k+r|k}$ for $r = 1, 2, \dots, n$ can be obtained using a single model.

The recursive relation between inputs and outputs in MSA prediction is expressed using the general non-linear models:

$$\hat{y}_{k+n|k} = f(\hat{y}_{k+n-1|k}, \hat{y}_{k+n-2|k}, \dots, y_k, \dots, y_{k-\varphi}, \hat{u}_{k+n-1|k}, \hat{u}_{k+n-2|k}, \dots, u_k, \dots, u_{k-\phi}) \tag{14}$$

where n is the multiple step ahead prediction horizon, φ and ϕ represent delays in the output and input, respectively. The functional form $f(\cdot)$ in Eq. (14) is approximated using a neural network.

In this case the multiple step ahead predictor for LIM, take the following form:

$$\begin{aligned} x_{1,k+1} &= w_{11,k}S(v_k) + w_{12,k}S(v_k)S(\lambda_{r\beta,k})i_{s\alpha,k} + w_{13,k}S(v_k)S(\lambda_{r\alpha,k})i_{s\beta,k} \\ x_{2,k+1} &= w_{21,k}S(v_k)S(\lambda_{r\beta,k}) + w_{22,k}i_{s\beta,k} \\ x_{3,k+1} &= w_{31,k}S(v_k)S(\lambda_{r\alpha,k}) + w_{32,k}i_{s\alpha,k} \\ x_{4,k+1} &= w_{41,k}S(\lambda_{r\alpha,k}) + w_{42,k}S(\lambda_{r\beta,k}) + w_{43,k}S(i_{s\alpha,k}) + w_{44,k}u_{\alpha,k} \\ x_{5,k+1} &= w_{51,k}S(\lambda_{r\alpha,k}) + w_{52,k}S(\lambda_{r\beta,k}) + w_{53,k}S(i_{s\beta,k}) + w_{54,k}u_{\beta,k} \end{aligned} \tag{15}$$

where x_1 predicts position $q_{m,k}$; x_2 and x_3 predicts fluxes $\lambda_{r\alpha}$ and $\lambda_{r\beta}$, respectively; x_4 and x_5 predicts the currents $i_{s\alpha}$ and $i_{s\alpha}$, respectively, in this case, prediction horizon is for 4 steps ahead, i.e. $\varphi = 4$, this has been experimentally obtained. The training is performed on-line, using a series-parallel configuration. All the NN states are initialized in a random way as well as the weights vectors, without any restriction in their initialization. It is important to remark that the initial conditions of the plant are completely different from the initial conditions for the NN. It is important to note that the objective of this work is to develop a controller without previous knowledge of the plant model or of its parameters. Hence, Eq. (13) only serves as a guide to design the neural model (15). This NN structure is determined heuristically in order to minimize the state estimation error; however, it has a block control form [13] in order to simplify the synthesis of nonlinear controllers; regarding to the number of neurons, it is determined by the number of state variables to be identified.

5.4 Real-Time Implementation Results

For PSO algorithm the design parameters have been selected as:

$$\begin{aligned}
 & \text{population size} = 10 \\
 & c_1 = 1.05 \\
 & c_2 = 1.1 \\
 & \text{Maximun number of iterations} = 100
 \end{aligned}$$

The designed cost function is defined in two steps as:

$$J_{1,k} = (i_{\alpha,k} - i_{\alpha ref,k})^2 + (i_{\beta,k} - i_{\beta ref,k})^2$$

where $i_{\alpha ref}$ and $i_{\alpha \beta ref}$ are computed on-line from another cost function defined as:

$$J_{2,k} = (q_{m,k} - q_{m ref,k})^2 + (\Psi_k - \Psi_{ref,k})^2$$

where $q_{m ref}$ and Ψ_{ref} are defined by a reference system that can be modified on-line as depicted on Fig. 1.

It is important to note that $P_{ij,0}$ and $Q_{ij,0}$ are diagonal matrixes which non-zero elements are defined as p_{ij} and q_{ij} , respectively, selected as: $p_{ij} = 1 \times 10^3$, $q_{ij} = 1 \times 10^{-3}$, $p_{ij} = 1 \times 10^4$ and $\eta_{ij} = 1$, with $i = 1, 2, \dots, n$ and $j = 1, 2, \dots, \varphi$, where n is the number of plant state variables ($n = 5$) and φ corresponds to the number of step ahead to be predicted ($\varphi = 4$).

Experimental results are presented as follows: Fig. 3, presents the result for position control; Fig. 4. displays result for magnitude of flux control, then Figs. 5 and 6 show the performance of α -axis and β -axis, respectively; next Figs. 7 and 8 depicts performance of currents in α -axis and β -axis, respectively; Then, Fig. 9, shows the

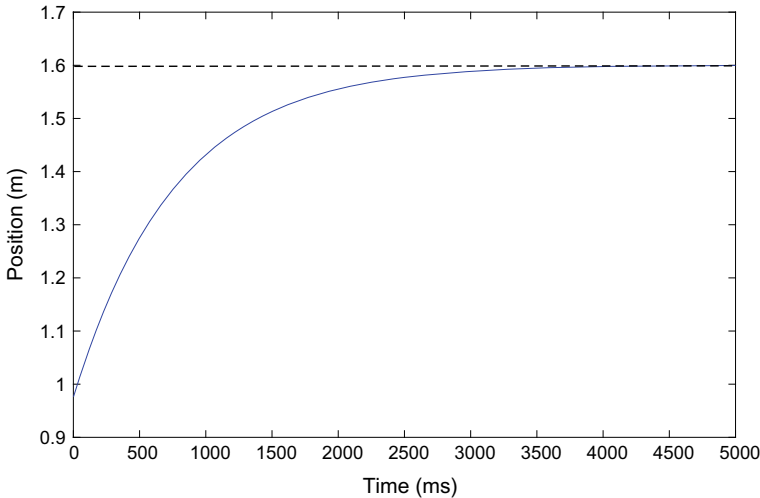


Fig. 3 Position control result (plant signal in solid line, reference signal in dash line)

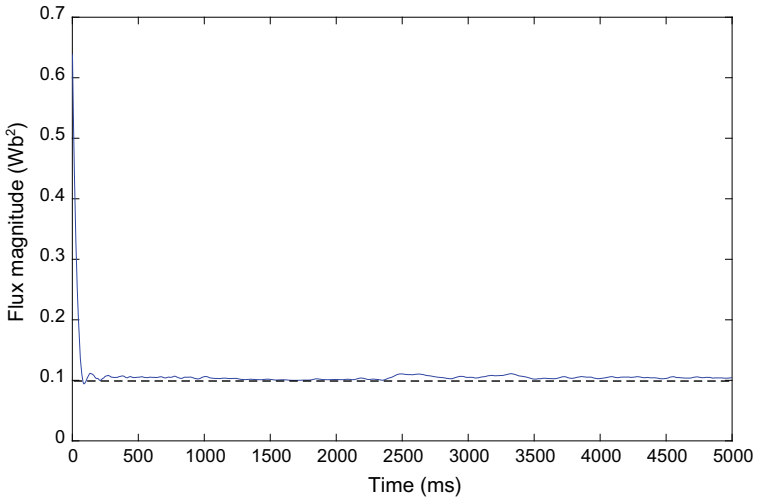


Fig. 4 Flux magnitude control result (Plant signal in solid line, reference signal in dash line)

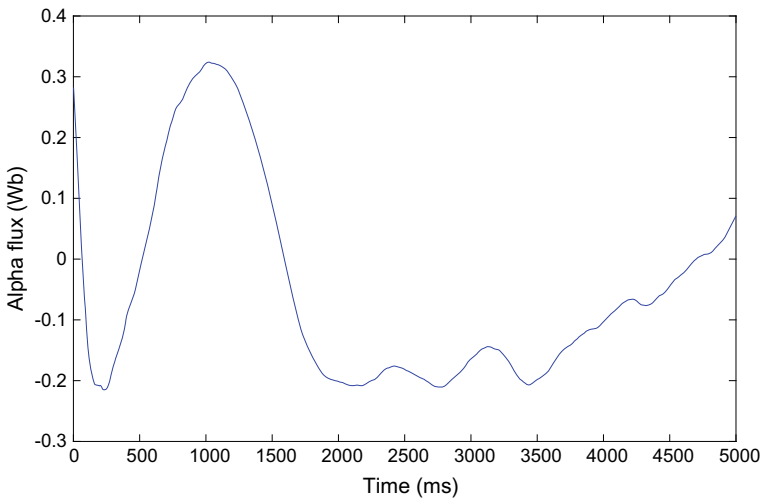


Fig. 5 Alpha flux performance

development of speed signal; finally, Fig. 10 and Fig. 11 include applied voltages in α -axis and β -axis, respectively.

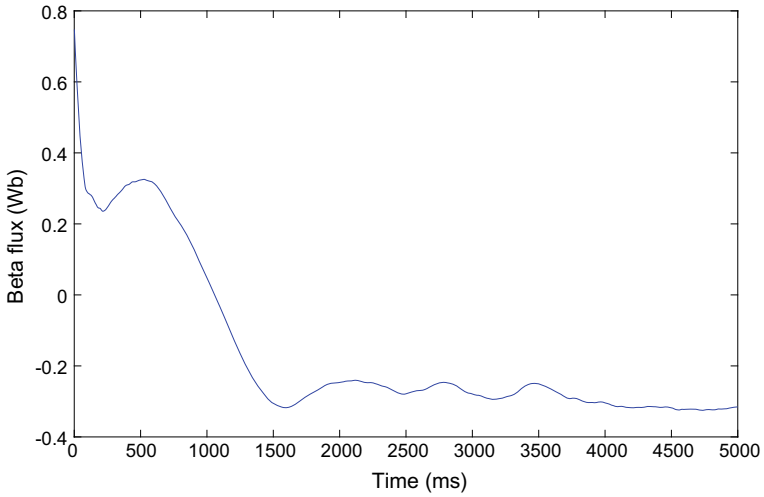


Fig. 6 Beta flux performance

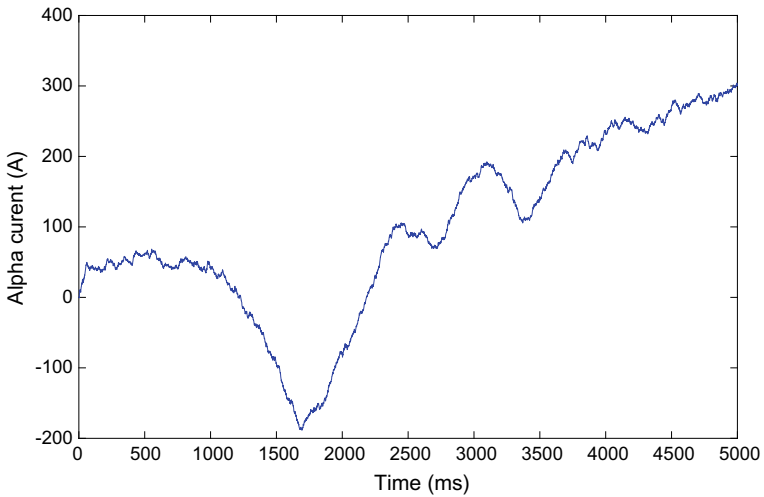


Fig. 7 Alpha current performance

6 Conclusions

This chapter proposes two modifications for MPC, in order that it can be applied to control discrete-time unknown nonlinear systems. These two modifications consist in a first stage into model the system to be controlled, by using a RHONN trained with an EKF algorithm, this neural model, has two functions, first provide an accurate model for the plant to be controlled and then produce the prediction needed to evaluate

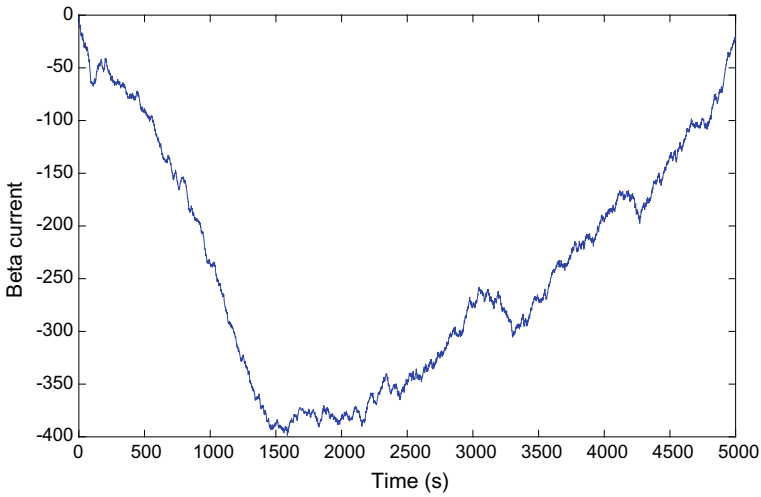


Fig. 8 Beta current performance

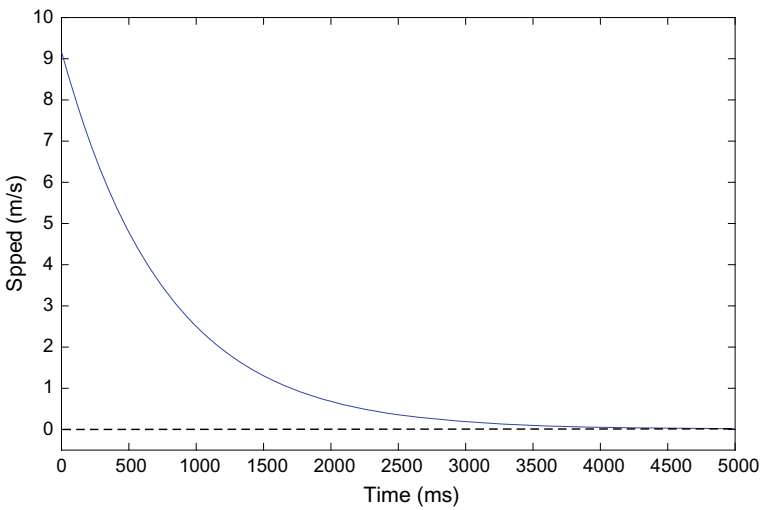


Fig. 9 Speed development

the optimal controller to be obtained in a second stage of modifications. The second stage of modified MPC, consists in using an evolutionary algorithm, PSO in this case, in order to solve the optimal control problem, this allows the application of the proposed controller to unknown discrete-time nonlinear systems without the need of solve complex optimization equations. Whith these two modifications it is obtained a neural predictive controller with evolutionary optimization algorithms to solve trajectory tracking problems of unknown discrete-time nonlinear systems

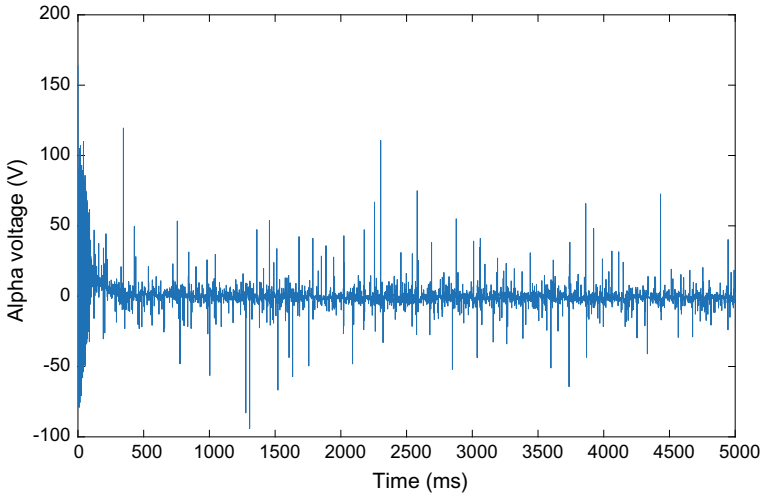


Fig. 10 Applied alpha voltage

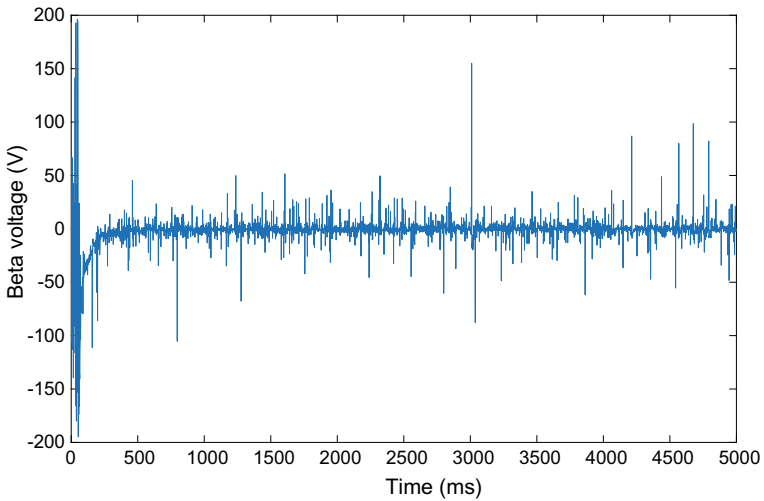


Fig. 11 Applied beta voltage

with experimental results, due to the neural model this controller can be used under disturbances, parametric variations, noises, unknown delays, saturations, unknown dynamics, etc.

Acknowledgements Authors thank the support of CONACYT Mexico, through Projects CB256769 and CB258068 (Project supported by Fondo Sectorial de Investigacion para la Educacion).

References

1. Bemporad, A., Morari, M.: Robust model predictive control: a survey. In: Garulli, A., Tesi, A. (eds.) *Robustness in Identification and Control*. Lecture Notes in Control and Information Sciences, vol. 245. Springer, London (1999)
2. Allgower, F., Zheng, A.: *Nonlinear Model Predictive Control*. Springer, Berlin (2000)
3. Chairez, I., García, A., Poznyak, A., Poznyak, T.: Model predictive control by differential neural networks approach. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*, Barcelona, pp. 1–8. <https://doi.org/10.1109/ijcnn.2010.5596521>. (2010)
4. Wu, W., Cbang, J.-X.: Neural predictive control design for uncertain nonlinear systems. In: *Proceedings of the IFAC Dynamics and Control of Process Systems*, Cambridge, Massachusetts, USA (2004)
5. Rankovic, V., Radulovic, J., Grujovic, N., Divac, D.: Neural network model predictive control of nonlinear systems using genetic algorithms. *Int. J. Comput. Commun. Control* **7**(3), 540–549 (2012)
6. Akesson, B.M., Toivonen, H.T.: A neural network model predictive controller. *J. Process Control* **16**, 937–946 (2006)
7. Pan, Y., Wang, J.: Robust model predictive control using a discrete-time recurrent neural network. In: Sun, F. et al. (eds.) *Part I, LNCS 5263*, pp. 883–892, Springer (2008)
8. Ge, S.S., Yang, C., Lee, T.H.: Adaptive predictive control using neural network for a class of pure-feedback systems in discrete time. *IEEE Trans. Neural Netw.* **19**(9), 1599–1614 (2008)
9. Yan, Z., Wang, J.: Robust model predictive control of nonlinear systems with unmodeled dynamics and bounded uncertainties based on neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **25**(3), 457–469 (2014)
10. Georgieva, P., Feyo de Azevedo, S.: Neural networks for model predictive control. In: *Proceedings of the 2011 International Joint Conference on Conference: Neural Networks (IJCNN)*, San Jose California, USA (2011)
11. Hedjar, R.: Adaptive neural network model predictive control. *Int. J. Innov. Comput. Inf. Control.* **9**(3) (2013)
12. Wang, X., Xiao, J.: PSO-Based Model Predictive Control for Nonlinear Processes. In: Wang, L., Chen, K., Ong, Y.S. (eds.) *ICNC 2005, LNCS 3611*, pp. 196–203, Springer (2005)
13. Alanis, A.Y., Sanchez, E.N., Loukianov, A.G.: Real-time recurrent neural state estimation. *IEEE Trans. Neural Netw.* **22**(3), 497–505 (2011)
14. Alanis, A.Y., Rios, J.D., Rivera, J., Arana-Daniel, N., Lopez-Franco, C.: Real-time discrete neural control applied to a linear induction motor. *Neurocomputing* **164**, 240–251 (2015)
15. Rios, J.D., Alanis, A.Y., Lopez-Franco, C., Arana-Daniel, N.: RHONN identifier-control scheme for nonlinear discrete-time systems with unknown time-delays. *J. Franklin Inst.* **355**, 218–249 (2018)
16. Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pp. 1942–1948 (1995)
17. Arora, R.K.: *Optimization: Algorithms and Applications*. Chapman and Hall/CRC Press, UK (2015)
18. Das, T.K., Venayagamoorthy, G.K.: Bio-inspired algorithms for the design of multiple optimal power system stabilizers: SPPSO and BFA. *IEEE Trans. Ind. Appl.* **44**(5), 1445–1457 (2008)
19. Boldea, I., Nasar, S.A.: *Linear Electric Actuators and Generators*. Cambridge University Press, Cambridge, England (1997)
20. Gieras, J.F.: *Linear Inductions Drives*. Oxford University Press, Oxford, England (1994)
21. Takahashi, I., Ide, Y.: Decoupling control of thrust and attractive force of a LIM using a space vector control inverter. *IEEE Trans. Ind. Appl.* **29**, 161–167 (1993)
22. Loukianov, A.G., Rivera, J., Cañedo, J.M.: Discrete time sliding mode control of an induction motor. In: *Proceedings IFAC'02, Barcelona, Spain* (2002)
23. Benitez, V.H., Sanchez, E.N., Loukianov, A.G.: Neural identification and control for linear induction motors. *J. Intel. Fuzzy Syst.* **16**(1), 33–55 (2005)

24. Kazantzis, N., Kravaris, C.: Time-discretization of nonlinear control systems via Taylor methods. *Comput. Chem. Eng.* **23**, 763–784 (1999)
25. Chen, P.-A., Chang, L.-C., Chang, F.-J.: Reinforced recurrent neural networks for multi-step-ahead flood forecasts. *J. Hydrol.* **497**, 71–79 (2013). <https://doi.org/10.1016/j.jhydrol.2013.05.038>
26. Chi, J., Kim, H.-C.: Prediction of arctic sea ice concentration using a fully data driven deep neural network. *Remote Sens.* **9**(12), 1305 (2017). <https://doi.org/10.3390/rs9121305>

Filter Size Optimization on a Convolutional Neural Network Using FGSA



Yutzil Poma, Patricia Melin, Claudia I. González and Gabriela E. Martínez

Abstract This paper presents an approach to optimize the filter size of a convolutional neural network using the fuzzy gravitational search algorithm (FGSA). The FGSA method has been applied in others works to optimize traditional neural networks achieving good results; for this reason, is used in this paper to optimize the parameters of a convolutional neural network. The optimization of the convolutional neural network is used for the recognition and classification of human faces images. The presented model can be used in any image classification, and in this paper the optimization of convolutional neural network is applied in the CROPPED YALE database.

Keywords FGSA · Convolutional neural network · Optimization · Neural network · Pattern recognition

1 Introduction

There are many methods for the identification and classification of images, one of them is the use of convolutional neural networks (CNN) which are neural networks that specialize in the recognition and classification of images. This convolutional neural network reduces the number of connections, as well as the number of parameters used in the training stage, which is considered one of its main advantages. In

Y. Poma · P. Melin (✉) · C. I. González · G. E. Martínez
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: pmelin@tectijuana.mx

Y. Poma
e-mail: yutpoma@hotmail.com

C. I. González
e-mail: cgonzalez@tectijuana.mx

G. E. Martínez
e-mail: gmartinez@tectijuana.mx

1990, a CNN was first used for the recognition of handwritten numbers using Back-propagation [1]. Totally connected neural networks have been used in recent time in various applications such as the integration of a modular neural network which is based on the Type 1 and Type 2 integral of Choquet, used for the recognition of human faces [2]. In addition, MNN have been used in applications such as the optimization of the weights of a neural network based on a hierarchical multi-objective genetic algorithm [3]. Convolutional neural networks have been used in commercial applications such as LeNet: which is the automatic check reading used in U.S.A. and Europe in the 90s in a DSP by AT & T [4], or in the detection and monitoring of customers in supermarkets by NEC Labs [5], as well as in the detection and automatic erased of faces and license plates for the protection of privacy in Google Street View [6], in addition the use of these networks has been made in experimental applications such as in the identification of facial expressions [7], or the segmentation of biological images [8], and in turn in the reconstruction of neural circuits from cross-sectional images of the brain as nanometric thickness [9]. In recent times CNNs have been used for different applications, these networks have also been optimized different methods, such as in the [10] where a network parameter is optimized, which is responsible for dividing the input data that subsequently enter to the training stage or the optimized based image recognition through target region selection [11]. The main contribution of this work is to optimize the filter size used in the convolutional layer for a CNN, this parameter is important because it determines the characteristics to obtain from the image, so the architecture of the network is more suitable for the processing of images and to improve the classification accuracy. This network was used in the detection of faces with record speed and precision [12], the method used for the optimization of the convolutional neural network is the Fuzzy Gravitational Search Algorithm, also called FGSA by its acronym in English [13]. This method is a transformation of the Gravitational Search Algorithm or GSA that is inspired by Newton's second law [14]. The FGSA differs from the GSA because it alters the alpha variable using a fuzzy system, unlike its predecessor that the alpha value keeps it static, as in [15] and [16].

The rest of the paper is organized as follow. In Sect. 2 we present the theoretical framework about the CNN and FGSA method. The proposed methodology is presented in Sect. 3. In Sect. 4 we have the results and discussions and finally in Sect. 5 we can see the conclusion of the experiments done.

2 Literature Review

This section presents the basic concepts necessary to understand the proposed method.

2.1 Convolutional Neural Networks

They are also called CNN or ConvNet, these networks are very popular within deep neural networks, since they are characterized for being experts in the extraction of characteristics of the input data. This network consists of different types of layers, these are the convolution layer, the grouping layer, making use of an activation function between each layer, ending with a multilayer classification layer [17]. Often these neural networks are used in conjunction with GPUs because they help to optimize the time of network processing [18].

2.1.1 Convolution Layer

In this layer a filter is used which consists of a size of $N * N$ which travels through the entire input image, convolving this image with the created convolution filter and its values, obtaining the main characteristics of the image called “map of characteristics” [19].

2.1.2 Pooling Layer

This part combines the pixels closest to the delimited area, as well as reduces the size of the image, takes with a mask of $m * m$, small samples of the map of characteristics created in the convolution layer [20], calculating the average or the maximum [21].

2.1.3 Classifier Layer

In this layer a fully connected layer is used, and acts as a multilayer perceptron since each neuron is a pixel. In this layer we have the same number of neurons as the number of classes we want to predict. In this phase of the convolutional neural network, the images that will be taken as output are classified and recognized [22].

2.2 GSA

It is known as a Gravitational Search Algorithm, this method has been inspired by the Law of Gravity that was proposed by Newton in 1685, which tells us: “The gravitational force between two bodies is directly proportional to the product of their masses and inversely proportional to the square of the distance that separates them” [23]. In Eq. (1) this law is represented:

$$F = G \frac{M_1 M_2}{R^2} \quad (1)$$

where R^2 represents the distance between two particles, G is the gravitation constant, M_1 and M_2 are the mass of the first and second particles, finally F is the magnitude of the gravitational force.

In Fig. 1, a general flow chart of the GSA process is shown. This begins by starting the population, evaluates each agent, also updates G and obtains the worst and best agent, calculates alpha and M for each agent, then updates the position and speed ending with take a criterion of whether or not you have found the best result and if you have not found it, go back to the process to evaluate each agent and successively.

2.3 FGSA

The fuzzy gravitational search algorithm (FGSA) is a modification to the gravitational search algorithm or GSA because it is responsible for modifying the alpha parameter using a fuzzy system, which modifies this parameter gradually in a range of 0 to 1.

In Fig. 2, we can appreciate the FGSA process, this starts the population, then evaluates each agent, updates G and gets the best and worst agent, calculates M and alpha for each agent, later updates the position and speed ends with deciding if it found the better result and if we have not found it, start the process to evaluate each agent and then carry out the following steps.

3 Proposed Method

The main idea of this proposal is to optimize the filter size used in the convolutional layer of a CNN. In Fig. 3 we present a flow chart about how the FGSA method interacts and works with the convolutional neural network to obtain the best solution, representing the highest percentage of recognition.

The steps below state how the FGSA and CNN work:

- Step 1. The FGSA generates an initial matrix with values, in which each vector is an agent and each agent has N dimensions in this study case we used 1, the value of the vector corresponds to the filter size of convolution layer, see Table 1.
- Step 2. The FGSA obtains the value corresponding to the Filter size and it is send to the CNN.
- Step 3. CNN performs the architecture based on the parameters sent by the FGSA method.
- Step 4. The neural network creates the filter of the convolution layer, based on the parameter given by the FGSA.
- Step 5. The database enters the training stage.
- Step 6. CNN obtains the best recognition value of the images and this value is returned to the FGSA to evaluate the best fitness and to generate the new solution.

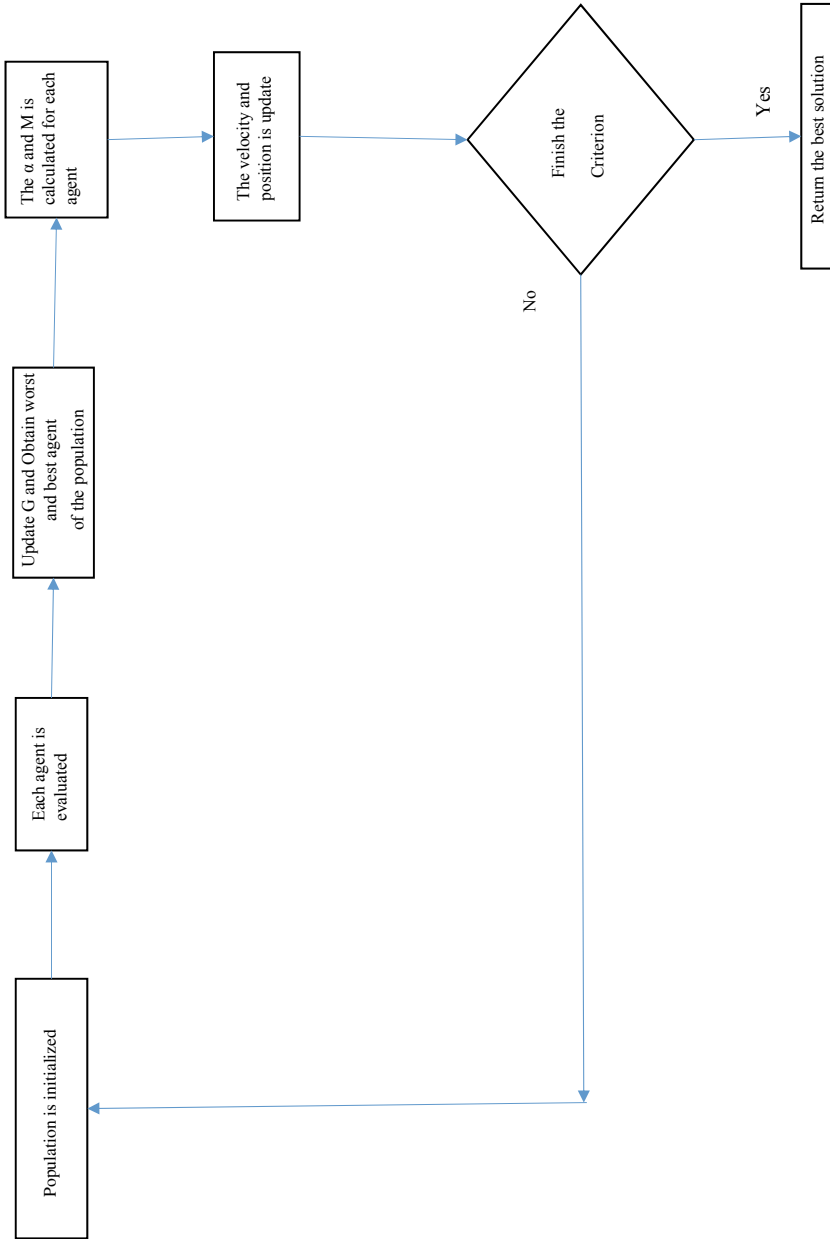


Fig. 1 Flow chart of GSA

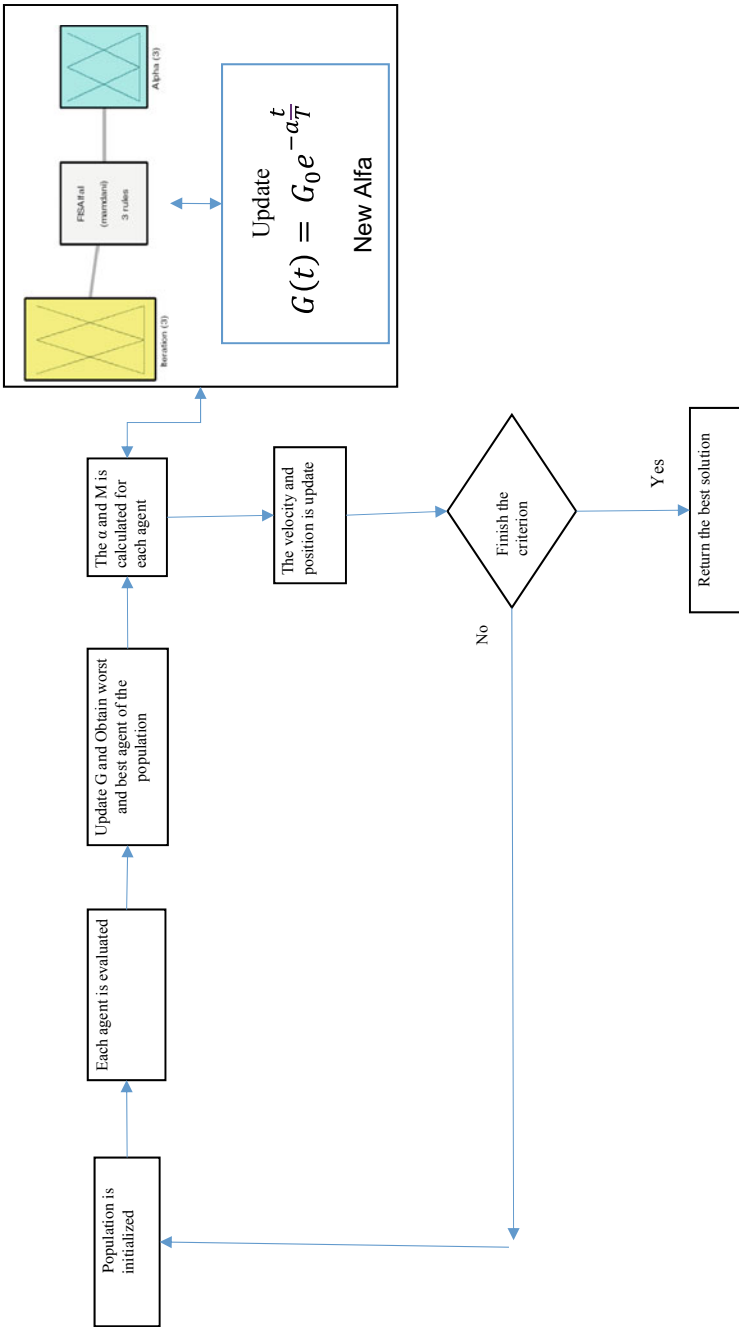


Fig. 2 Flow chart of FGSA

Fig. 3 Proposed method CNN working with the FGSA

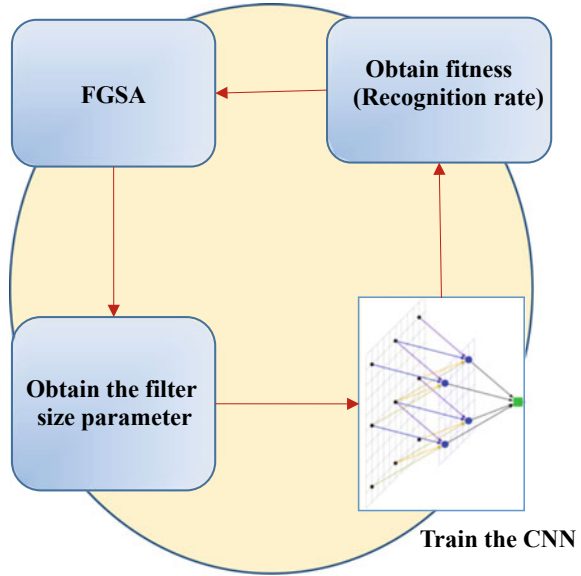


Table 1 Example of matrix initial with random values

Dimension	1	2	3
Number of agents	-	-	Size of filter (1-50)
1	14	8	17
2	38	20	9
3	19	7	5

Bold means the best solution

The architecture of the convolutional neural network is presented in Table 2, which shows us the data entry (images from the database), passing to the first layer of the network called convolution, this layer is responsible for extracting the main

Table 2 Architecture of convolutional neural network

Layer	Observation
Input	M * N
Convolution	FGSA Obtain size of filter of convolution (x * x), 1 layer
ReLU	
Pooling	1 layer, Average (2 * 2)
ReLU	
Hidden layers	100 nodes
Output/softmax	38 nodes

Bold means the part that is being optimized

characteristics of the image using a filter called “convolution filter” whose size is decided from the parameter sent by the FGSA. Then with the extraction of characteristics the “characteristics map” is created, which is subsequently applied activation function called ReLU (rectified linear unit) is responsible for allowing the passage of all positive values without changing them, but assigns all negative values to zero, continuing to the next layer called pooling layer (Pooling) which consists of extract the main characteristics of the feature map, by means of a $2 * 2$ mask which travels the characteristics map and obtains the average or the maximum value of the pixels that cover the mask, thus forming a new feature map, for this experiment it was taken as the average in the pooling layer. Finally it passes to the classification layer and finally it obtains a recognition percentage of the image.

The parameter to optimize (filter size) is responsible for obtaining the best filter size that will be used in the convolution stage to perform the characteristics map. This value is important because it influence the mask that will take on more or less characteristics of the image. These characteristics help the network to have a better image recognition. A small filter gets more characteristics of the image because it segmented the image more times, thus taking more characteristics of it, unlike a large filter that segments the image less times, therefore the taking of characteristics is smaller and tends to give lower percentage in the recognition of the employers.

4 Results and Discussion

In the case study the CROPPED YALE database was used, which consists of 380 images of human faces, these represent 38 humans with 10 images taken at different angles of each. To carry out the experiments, a sample of 80% of the images (304 images) was taken for the training phase of the network, and the remaining 20% for the tests, which correspond to 76 images.

In Fig. 4 a sample of the images contained in the CROPPED YALE database is illustrated.

In Table 3, the parameters used for the convolutional neural network are presented.

The parameters used by the FGSA method are presented in Table 4.

In Table 5, the results obtained for the filter size optimization that the convolution layer uses to take the characteristics of the image are presented; where the best value was when the filter size is $9 * 9$ achieving a recognition rate of 100%.

A manual test was carried out to verify the results obtained by the FGSA method, this consists in modifying the filter size from $1 * 1$ to $50 * 50$. In the experiment was observed that the filters with an even number size are not applied in the process of the convolution, only the filter sizes that are odd. The results obtained can be summarized in Table 6.

In Table 7 we can find the best percentages of recognition, where the highest value obtained was 100%. The size of filter $9 * 9$ was identified as the best result because it is the one that took less time for image recognition.



Fig. 4 Images CROPPED YALE database

Table 3 Parameters of convolutional neural network

CNN parameters	
Total epochs	50
Total images	380
Training images	304
Testing images	76
Number of block for images to training	38
Size of filter	FGSA
Number of filter	20

Bold indicates the parameter that is optimized by the FGSA

Table 4 Parameters used for FGSA

FGSA parameters	
Iterations	15
Agents	15
Dimensions	3

Table 5 Size of filter optimized with FGSA

Experiment number	Filter size	Recognition rate
1	17 * 17	98.68
2	9 * 9	100
3	9 * 9	100
4	9 * 9	100
5	9 * 9	100
6	9 * 9	100
7	9 * 9	100
8	9 * 9	100
9	9 * 9	100
10	9 * 9	100
11	9 * 9	100
12	9 * 9	100
13	9 * 9	100
14	9 * 9	100
15	9 * 9	100

Bold value represents the best solution found

5 Conclusions

It is concluded that the best value for the filter size of the convolution layer has been obtained by the optimized convolutional neural network, which gives us a perfect recognition percentage. The results show that the size of the filter has a great influence on the taking of characteristics of the image, thus affecting the percentage of recognition given by the network.

Future work is intended to modify the architecture of the convolutional neural network, adding more layers of convolution and pooling that may reduce the time of image recognition in the network.

Table 6 Manually modified filter size

Filter size	Recognition rate	Time: seconds
1 * 1	100	3553.9
2 * 2	0	2.6
3 * 3	17.11	3226.8
5 * 5	3.95	2860.6
7 * 7	15.79	2958
8 * 8	0	2.5
9 * 9	100	3356.5
10 * 10	0	3
11 * 11	100	3860.4
12 * 12	0	2.6
13 * 13	100	3928.9
14 * 14	0	2.3
15 * 15	100	3757.7
16 * 16	0	2.4
17 * 17	98.68	4726.2
18 * 18	0	2.6
19 * 19	100	5180
20 * 20	0	2.5
21 * 21	94.74	5174.2
22 * 22	0	2.4
23 * 23	96.05	29,650
24 * 24	0	2.7
25 * 25	96.05	5047.4
26 * 26	0	2.9
27 * 27	98.68	5235.6
28 * 28	0	2.6
29 * 29	100	5881.6
30 * 30	0	2.7
31 * 31	88.16	6631.5
32 * 32	0	2.5
33 * 33	5.26	6253.1
34 * 34	0	2.7
35 * 35	6.58	6073.4
36 * 36	0	2.7
37 * 37	3.94	7165.2
38 * 38	0	2.8
39 * 39	48.68	7129.9

(continued)

Table 6 (continued)

Filter size	Recognition rate	Time: seconds
40 * 40	0	2.7
41 * 41	2.63	7614.2
42 * 42	0	2.6
43 * 43	3.94	37,042.8
44 * 44	0	2.9
45 * 45	2.63	7544.2
46 * 46	0	2.8
47 * 47	3.94	8681.8
48 * 48	0	3
49 * 49	2.63	8631.9
50 * 50	0	4.5

Table 7 Abstract of manually modified filter size of the best results

Filter	Recognition rate	Time: seconds
9	100	3356.5
1	100	3553.9
15	100	3757.7
11	100	3860.4
13	100	3928.9
19	100	5180
29	100	5881.6

Bold value represents the best solution found

Acknowledgements We thank our sponsor CONACYT & the **Tijuana Institute of Technology** for the financial support provided with the scholarship number 816488.

References

1. Le Cun Jackel, L.D., Boser, B., Denker, J S., Henderson, D., Howard, R.E., Hubbard, W., Le Cun, B., Denker J., Henderson, D.: Handwritten digit recognition with a back-propagation network. *Adv. Neural Inf. Process. Syst.* 396–404 (1990)
2. Martínez, G.E., Melin, P., Mendoza, O., Castillo, O.: Face recognition with a Sobel edge detector and the Choquet integral as integration method in a modular neural networks. In: *Neural Networks and Nature-Inspired Optimization*, pp. 59–70. Springer, Cham (2015)
3. Melin, P., Sánchez, D.: Multi-objective optimization for modular granular neural networks applied to pattern recognition. *Inf. Sci. (Ny)* **460–461**, 594–610 (2018)
4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. IEEE* **86**(11), 2278–2323 (1998)
5. Jialue, F., Wei, X., Ying, W., Yihong, G.: Human tracking using convolutional neural networks. *IEEE Trans. Neural Networks* **21**(10), 1610–1623 (2010)

6. Frome, A., Cheung, G., Abdulkader, A., Zennaro, M., Wu, B., Bissacco, A., Hartwig, H., Neven, H., Vincent, L.: Large-scale privacy protection in google street view. In: ICCV, pp. 2373–2380. IEEE (2009)
7. Kanou Kanou, S.E., Ferrari, R.C., Mirza, M., Jean, S., Carrier, P.L., Dauphin, Y., Boulanger-Lewandowski, N., Aggarwal, A., Zumer, J., Lamblin, P., Raymond, J.P., Pal, C., Desjardins, G., Pascanu, R., Warde-Farley, D., Torabi, A., Sharma, A., Bengio, E., Konda, K.R., Wu, Z., Bouthillier, X., Froumenty, P., Gülçehre, G., Memisevic, R., Vincent, P., Courville, A., Bengio, Y.: Combining modality specific deep neural networks for emotion recognition in video. In: Proceeding 15th ACM International Conference Multimodal Interact—ICMI’13, pp. 543–550 (2013)
8. Ning, F., Delhomme, D., LeCun, Y., Piano, F., Bottou, L., Barbano, P.E.: Toward automatic phenotyping of developing embryos from videos. *IEEE Trans. Image Process.* **14**(9), 1360–1371 (2005)
9. Jain, V., Murray, J., Roth, F., Turaga, S., Zhigulin, V.: Supervised learning of image restoration with convolutional networks supplementary material: specific methods. *Parameters* **2**(2), 3–5 (2007)
10. Poma, Y., Melin, P., González, C.I., Martínez, G.E.: Optimal recognition model based on convolutional neural networks and fuzzy gravitational search algorithm method (2018)
11. Wu, H., Bie, R., Guo, J., Meg, X., Wang, S.: Optimized CNN based image recognition through target region selection **156**, 772–777 (2018)
12. Nasse, F., Thureau, C., Fink, G.A.: Face detection using gpu-based convolutional neural networks. *Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artificial Intelligence Lect. Notes Bioinformatics)*, **5702**, 83–90 (2009)
13. Sombra, A., Valdez, F., Melin, P., Castillo, O.: A new gravitational search algorithm using fuzzy logic to parameter adaptation. In: Proceeding EEE Congress on Evolutionary Computation (CEC), pp. 1068–1074 (2013)
14. Rashedi, E., Nezamabadi-pour, H., Saryazdi, S.: GSA: a gravitational search algorithm. *Inf. Sci. (Ny)* **179**(13), 2232–2248 (2009)
15. Hatamlou, A., Abdullah, S., Othman, Z.: Gravitational search algorithm with heuristic search for clustering problems. *Conf. Data Min. Optim.* **June**, 190–193 (2011)
16. Verma, O.P., Sharma, R.: Newtonian gravitational edge detection using gravitational search algorithm. In: International Conference on Communication Systems and Network Technologies IEEE, Rajkot, India, pp. 184–188 (2012)
17. Bengio, Y., LeCun, Y.: Convolution networks for images, speech, and time-series. In: Arbib, M.A. (ed.) *The Handbook of Brain Theory and Neural Networks*. MIT Press, vol. 1, pp. 1–5 (1998)
18. Chellapilla, K., Puri, S., Simard, P.: High performance convolutional neural networks for document processing. *Int. Work. Front. Handwrite. Recognition* (2006)
19. Venkatesan, R., Li, B.: *Convolutional Neural Networks in Visual Computing: A Concise Guide*. Engineering, Taylor and Francis Group, CRC Press, Data-Enabled (2017)
20. Ranzato, M.A., Huang, F.J., Boureau, Y.L., LeCun, Y.: Unsupervised learning of invariant feature hierarchies with applications to object recognition. In: *IEEE Conference Computing Vis. Pattern Recognition*, pp. 1–8 (2007)
21. Wang, T., Wu, D.J., Coates, A., Ng, A.Y.: End-to-end text recognition with convolutional neural networks. In: *ICPR, International Conference Pattern Recognition*, pp. 3304–3308 (2012)
22. Lu, L., Zheng, Y., Carneiro, G., Yang, L.: *Deep Learning and Convolutional Neural Networks for Medical Image Computing*. Springer (2017)
23. Schutz, B.: *Gravity from the ground up*. Cambridge University Press, Cambridge (2003)

Evaluation and Analysis of Performances of Different Heuristics for Optimal Tuning Learning on Mamdani Based Neuro-Fuzzy System



Sukey Nakasima-López, Mauricio A. Sanchez and Juan R. Castro

Abstract The optimization algorithms based on gradients have a series of hyperparameters that will allow among other things to reach a global minimum of the error function, convergence, and at an adequate velocity, trying to avoid overtraining, divergent solutions and overload of computation to execute an exaggerated number of iterations for its process, hence, greater attention is paid to the learning rate as the main parameter to be adjusted during training to achieve a better performance in learning, for this reason, the contribution of this paper is to show initially the positive and negative impact that would have to use a suitable or not learning rate, the benefits of its adjustment during the learning phase, as well as the evaluation and analysis of the performance of different heuristics implemented on a gradient-based algorithm with momentum and adaptive learning rate, with the purpose of being able to have knowledge of their performance in different contexts and scenarios, since we know that this parameter will depend greatly on the complexity of the data to be treated, so that a fixed number cannot be determined for all cases, but if you can try to identify different scenarios, which can be managed from different heuristics for the optimal adjustment of the parameter during its learning process.

Keywords Learning rate · Tuning · Heuristics · Optimization algorithms

1 Introduction

Models generated from machine learning techniques based on gradients, have the ability to adjust design parameters (which act as weights in artificial neural networks) from an iterative process known as backpropagation, which can occur each epoch

S. Nakasima-López (✉) · M. A. Sanchez · J. R. Castro
Universidad Autonoma de Baja California, 22390 Tijuana, Mexico
e-mail: sukey.nakasima@uabc.edu.mx

M. A. Sanchez
e-mail: mauricio.sanchez@uabc.edu.mx

J. R. Castro
e-mail: jrcastro@uabc.edu.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_30

as is the case of batch learning or during their training phase that it is the case of online and mini-batch learning, which allows achieving estimated models with a close approximation to the real model based on their input data.

One of the challenges that arise during the above-mentioned learning processes is with respect to how much these design parameters should vary to achieve error minimization, adequate convergence, and efficient performance, since different degrees of variation of those parameters applied to the inputs are very sensitive to change, which is why they become critical for obtaining pseudo optimal solutions, the construction of generalized models and the reduction of their complexity.

Learning rate is considered one of the most important hyperparameters during the adjustment of design parameters when optimization algorithms based on gradients are used, since it allows among other things to reach a global minimum of the error function, convergence, and at an adequate velocity, trying to avoid overtraining, divergent solutions and overload of computation to execute an exaggerated number of iterations for its process, hence, greater attention is paid to the learning rate as the main parameter to be adjusted during training to achieve a better performance in learning [1–3].

The contribution of this paper is to show initially the positive and negative impact that would have to use a suitable or not learning rate, the benefits of its adjustment during the learning phase, as well as the evaluation and analysis of the performance of different heuristics implemented on a gradient-based algorithm with momentum and adaptive learning rate, with the purpose of being able to have knowledge of their performance in different contexts and scenarios, since we know that this parameter will depend greatly on the complexity of the data to be treated, so that a fixed number cannot be determined for all cases, but if you can try to identify different scenarios, which can be managed from different heuristics for the optimal adjustment of the parameter during its learning process.

This paper is organized as follow, in section one we will talk about the general concepts like the problem presented when the learning rate is not adjusted properly, the different heuristics and its generalities to highlights the relevance of the study, in the second section we will explain in detail each one of heuristic chosen, its functionality and implementation, and in the last sections we will treat subjects about experimentation, results obtained, discussions and conclusions.

1.1 Learning Rate as a Critical Hyperparameter

The **learning rate** (lr) is a hyperparameter that has the goodness to control the steps of the descent in direction of a good global minimum value when we use optimization algorithms based on gradients, an adequate value of this hyperparameter could represent a better performance and obtain an early convergence avoiding high cost to process for excessive iterations or to get stuck in some local minimum [4]. Hereunder going to present different behavior of the learning rate [5].

As we can observe in Fig. 1, when the lr has a lower value, it could take a lot step to reach the optimum and could tend to fall in local minimums, because the steps in descent are very small.

In Fig. 2, we can observe that when the value of lr is higher, the direction of gradient could have irregular descent, faster but with the possibility to losing the minimum global and even diverge.

Given this problematic, we need to find the mechanism of control this hyper-parameter, to obtain an adequate velocity of the gradient descent, trying to find a global minimum or at least a good local minimum and an early convergence, these objectives could be represented that the behavior shown in Fig. 3.

Fig. 1 Behavior of the gradient descent, when lr is initialized with a lower value

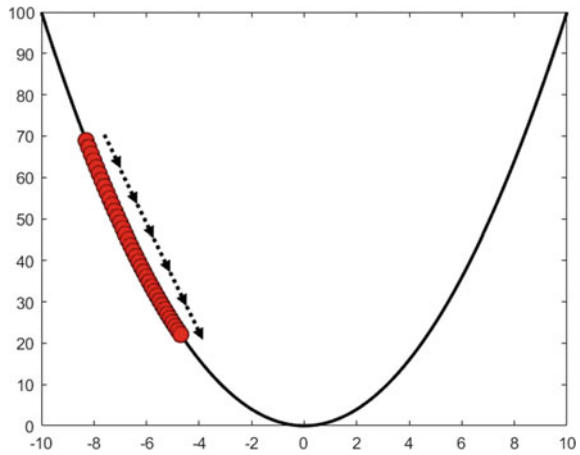


Fig. 2 Behavior of the gradient descent, when lr is initialized with a higher value

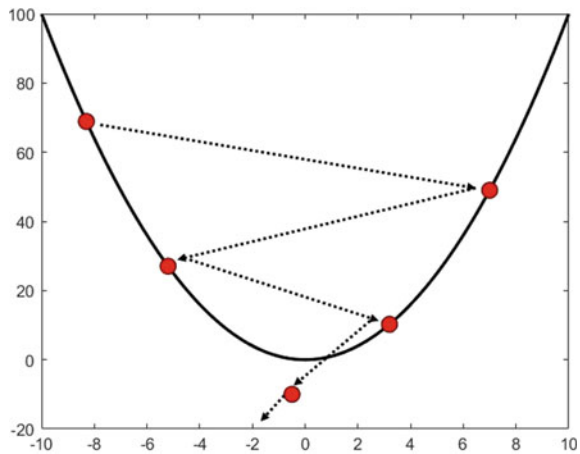
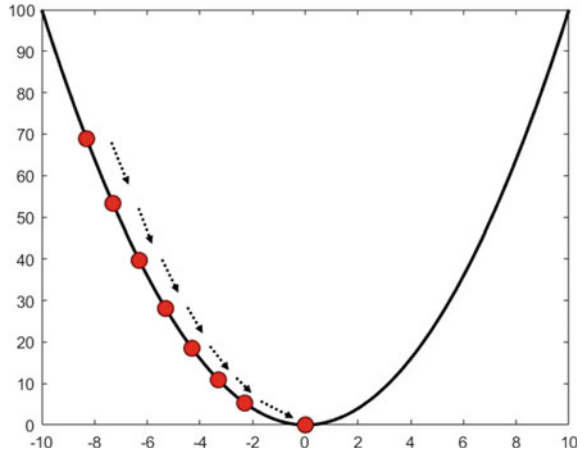


Fig. 3 Behavior of the gradient descent, when lr is initialized with an adequate value



1.2 Gradient Descent and Adaptive Learning Rate

Gradient descent is one an optimization algorithm most used, that during the learning process look to minimize a cost function defined. This is a differentiable function, that can be expressed in Eq. (1) as the following manner [6, 7]:

$$\nabla E(\theta) \stackrel{\text{def}}{=} \left[\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_i}, \frac{\partial E(\theta)}{\partial \theta_n} \right]^T \tag{1}$$

Where vector gradient is denoted by $\nabla E(\theta)$, that is calculated from the partial derivate of cost function (or error function) denoted by $\partial E(\theta)$ with respect to all design parameters expressed as $\partial \theta_i \therefore i = 1, \dots, n$.

Its behavior can be exemplified as shown in Fig. 4, where we can observe that from initial point represented by θ_{now} , d are possible directions that the descent vector could take from the steepest descent direction calculated from $\nabla E(\theta)$.

The learning rule allow apply to the actual point a change from its vector gradient and learning rate, it is expressed as follow Eq. (2):

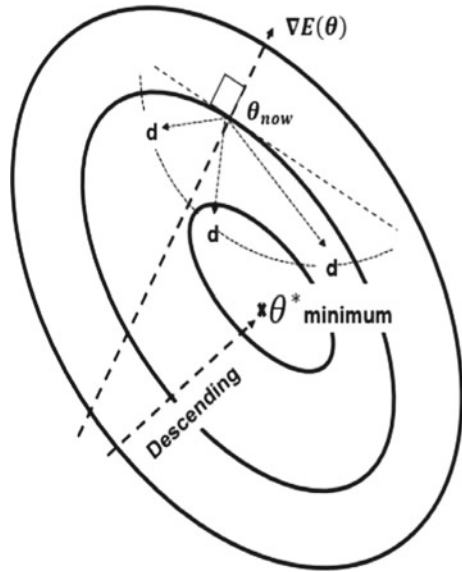
$$\theta_{next} = \theta_{now} - \eta \nabla E(\theta) \tag{2}$$

Where θ_{next} is the change of direction obtained from θ_{now} initial point, η learning rate and $\nabla E(\theta)$ gradient descent.

With the aims to get a better global minimum, it is necessary to implement an iterative process to calculate and obtain different directions, one of the methods most used for this purpose is known as Backpropagation [5, 8], that using the chain rule can compute gradients in each layer of the network for all design parameters.

It is in this method (referred above), that we can implement heuristics to change from fixed to adaptive the learning rate and get better results.

Fig. 4 Possible steepest descent directions obtained when gradient descent is calculated from a defined point



1.3 Heuristics for Optimal Learning Rate

A heuristic can be defined as a practical method, which based on patterns and behaviours observation, looking accelerate the process to reach a satisfactory solution. But these heuristics do not guarantee to be optimal, perfect, logical, or rational, however, these are a great help to reach immediate goals [9].

In optimization methods based on gradients, if these are processed in its classical form, it could take a long time its execution and present a slow performance, in this case, it is imperative to look for strategies that improve the behavior of these algorithms to get better its results [6].

As we discussed earlier, one of the critical hyperparameters, when we need to control the velocity and direction of the descent to minimize the error function, it is the so-called learning rate. In base of this argument, annealing heuristics represent an excellent choice to adapt this parameter in its learning process. Below is detailed some heuristics, where we will observe its mathematical form, its functionality, and behavior, among other relevant information.

1.4 Gradient Descent with Momentum and Adaptive Learning Rate (GDx)

It is considered as an improved form from the classical gradient, since a parameter is added to the method known as Momentum [7, 9], and also in each change of

epoch, applying a heuristic that compares the change of the error function (be this an increment or decrement), and based on some other considerations, lr is adapted.

The momentum term allows regulating the influence of the previous descent direction, it is usually set with a value between 0.1 and 1. Another advantage of this term is that smoothes weight updating and tends to overcome an unstable descent due to gradient noise [6]. Its mathematical form is as shown in Eq. (3):

$$\theta_{now} = mc \theta_{previous} - (1 - mc) * \eta * \nabla E(\theta) \quad (3)$$

where the definition of its parameters is as follow; θ_{now} current design parameters, mc momentum established as a constant value, η learning rate that will be adaptive during the learning process, $\theta_{previous}$ it is the previous design parameters and $\nabla E(\theta)$ gradient vector. Finally, the change of design parameters (weights) obtained θ_{now} , it is applied to the previous design parameters $\theta_{previous}$, to then use it in the learning process. It can be expressed as follow in Eq. (4):

$$\theta_{new} = \theta_{now} + \theta_{previous} \quad (4)$$

As we can observe in Fig. 5, the lr adaptation, it is given when is evaluated a change in error function calculated (both increment and decrement), the error function chosen for our experiments was Sum Squared Error (SSE), where its definition is like as shown in Eq. (5):

$$SSE = \sum_{i=1}^q (y - \hat{y})^2 \forall i = 1, \dots, q \quad (5)$$

In the heuristic implemented in Fig. 5, we can observe that the increment of the proportion between SSE_{now} and SSE_{prev} with respect to the **max_perf_inc** (that is maximum increment of performance, a training parameter defined before of being the learning process), it will cause that the decreasing of lr (through of the training parameter defined as $lr_{decrement}$, initialized before to training), and then this new value in lr will applied to vector gradient, and so going to the next epoch to recalculate the change of the new design parameters and proceed with the flow.

In the case where this proportion decrement, we evaluate if SSE_{now} is less than SSE_{prev} , and if it true we apply an increment to the lr through the training parameter $lr_{increment}$, independently that the improve is reflected in the decrease in the SSE_{new} with respect to SSE_{prev} , we replaced the values of $\theta_{previous}$ with those of θ_{new} and SSE_{prev} with the values of SSE_{new} , finally gradient vector is recalculated, going to the next epoch and proceed with its natural flow.

The possibility to adapt lr , in based on behavior of the error function, give us the opportunity to explorer different local minimum, and find a good global minimum, also the velocity to reach a convergence in less epochs is better than the classical gradient where the lr remain fixed.

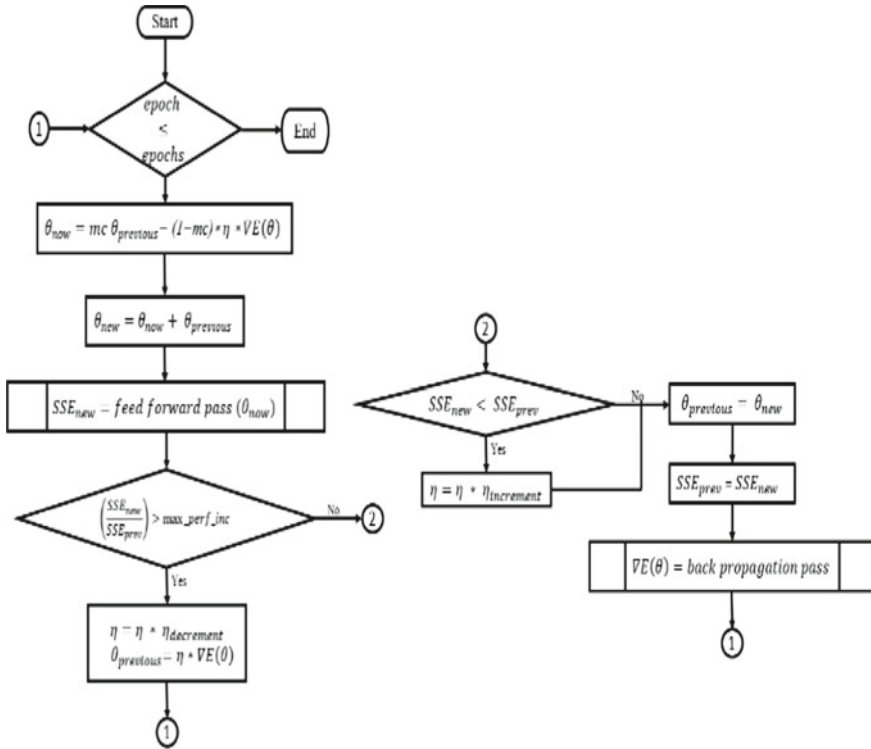


Fig. 5 Flow chart about of gradient descent with momentum and learning rate adaptive

1.5 Momentum Method to Update Design Parameters

This method, unlike the method of Gradient Descend with Momentum and Learning Rate Adaptive, use a velocity parameter and it is who change iteratively in each epoch.

Momentum determines how quickly accumulated previous gradients will degrade. If mc (momentum) is much larger than η (learning rate), the accumulated previous gradients will be dominant in the update rule, so the gradient in the iteration will not change the current direction quickly, but if on the other hand mc is much smaller than η , the previous accumulated gradients can act as a smoothing factor for the gradient [10, 11]. Its mathematical form is presented as follow in Eq. (6).

$$v_{now} = mc * v_{previous} - \eta * \nabla E(\theta) \tag{6}$$

The definition of the parameters that compound this equation are: η is learning rate, mc is momentum term, v is velocity (direction and speed), $\nabla E(\theta)$ is vector gradient and θ are design parameters.

Finally, the parameter v_{now} is applied to the previous design parameter $\theta_{previous}$ as we can observe in Eq. (7), that represents the learning rule.

$$\theta_{new} = \theta_{previous} + v_{now} \tag{7}$$

2 Annealing the Learning Rate

These kinds of heuristics also are known as Learning Rate Schedule or Adaptive Learning Rate, among the advantages offered are the increase in performance and reduction of training time [12–14]. Its general proceed starts by setting a high value to the lr and as the learning process progress, this value will be lower, so this behavior allows that the begin to generate big change and later take control by making smaller changes in both velocity and descent step. Some heuristics based on this technique have been chosen, there are detailed below [15–17]:

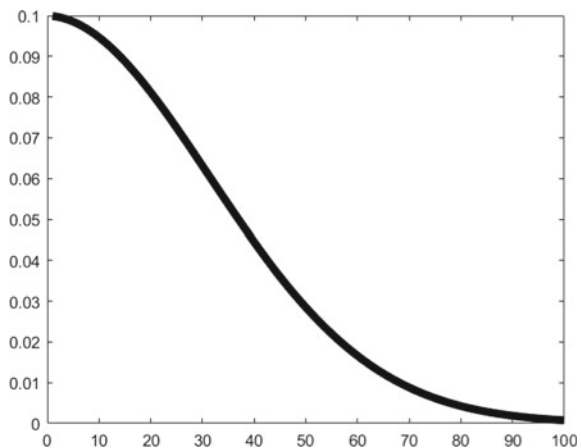
1. Time-Based learning rate schedule.

Of easy implementation and behavior simple, this heuristic allows us to decrement the lr in each epoch change. The lr is initialized with a high value, but continuously form this parameter will going decreasing, so in the begin the steps of descent will be big, but in the time will be more controlled because will going to be decreasing, this behavior can be seen on Fig. 6.

Its mathematical form is defined as shown in Eq. (8):

$$\eta_t = \frac{\eta_{t-1}}{(1 + kt)} \tag{8}$$

Fig. 6 Behavior lr adaptive, when time-based learning rate schedule is used



Where the definition of the parameter of the Eq. (8) are: η_t it is the lr in its actual time, being t the number of iterations or epochs, η_{t-1} it is the lr in its before time, ultimately a parameter that is established as fixed is k that is a decay value, that can be interpreted as the proportion of increment together with parameter t , allowing a continuous descent and soften.

2. Drop-Based Learning Rate Schedule.

The different of this heuristic with respect to the *time-based* mentioned before lies in that the decrement of lr is not continuous, in place, it can be interpreted as a discrete event, because its drop is given by a factor defined as *DropRate* (training parameter) and this change is executed in a determined epoch number, latter is defined in training parameter named *DropRate* (that is the time where lr will change), all foregoing can be visualized in Fig. 7, that shows the behavior above mentioned.

Also, as we can observe in Fig. 7, the decrement step to the next lr change is bigger than the consequent changes, and this is given with the intention to conserve a better control in the descent of gradient since this are very sensitive to the changes. Its mathematical form is expressed as is shown in Eq. (9):

$$\eta_t = \eta_{initial} (DropRate)^{t/EpochDrop} \tag{9}$$

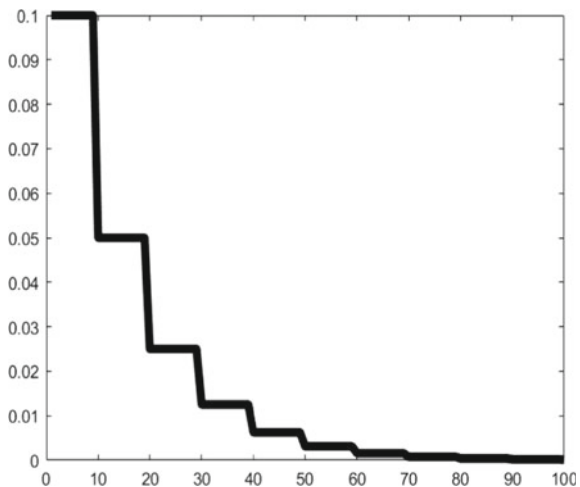
The definition of its training parameters is described as follow; η_t is lr

In its actual time, $\eta_{initial}$ is initial learning rate, *DropRate* is the decay value, *EpochDrop* is the time where the η will change, and t is the iteration or epoch number.

3. Exponential decay Learning Rate Schedule.

This heuristic presents an accelerated decrement in lr , consequently, this behavior is given for its exponential condition so that systematically the decrement is

Fig. 7 Behavior lr adaptive, when Drop-based learning rate schedule is used



generating in accordance with the time pass. This behavior can be visualized in Fig. 8.

The mathematical definition of this heuristic is shown in Eq. (10):

$$\eta_t = \eta_{initial} e^{-kt} \tag{10}$$

Where the definition of its parameters are described as follow: η_t is learning rate in its actual time, $\eta_{initial}$ is initial learning rate, k is decay value and t is iteration or epoch number.

4. Cyclical Learning Rate Schedule.

This heuristic is presented as a more efficient option compared to the others mentioned above, because unlike the previous ones, it does not handle a progressive decrease, if not on the contrary, generate ranges where the lower limit and upper limit is determined, being on these range over which will be swinging during the time defined or once obtained its convergence and minimization of the error function [18, 19]. We can observe this behavior as is shown in Fig. 9.

As we can observe in Fig. 9, the function that defines the behavior of this technique is a triangular, it is used for its easy to implement the ranges with lower and upper limits with a linear progress between this range defined, what helps to increase its velocity, another of the differences presented by this heuristic with respect to those previously evaluated, is that it starts from a low point and going to increasing step by step, once the upper point has been reached, it begins its descent again, and so on, hence the name cyclical.

This heuristic is useful because eliminate the necessity to perform many experiments to find the optimal value to lr . Its definition formal is presented in Eq. (11):

Fig. 8 Behavior lr adaptive, when Exponential Decay learning rate schedule is used

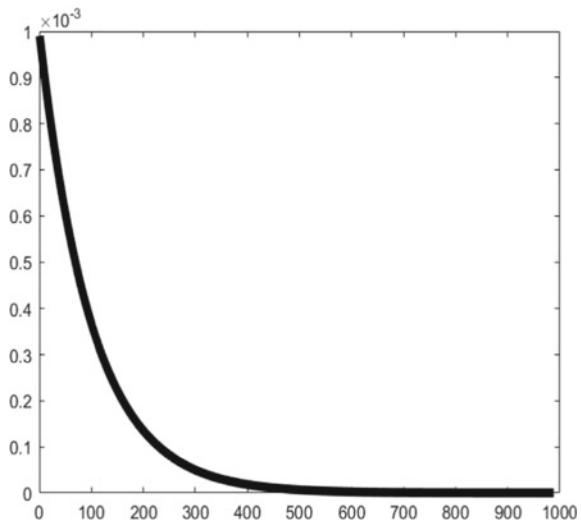
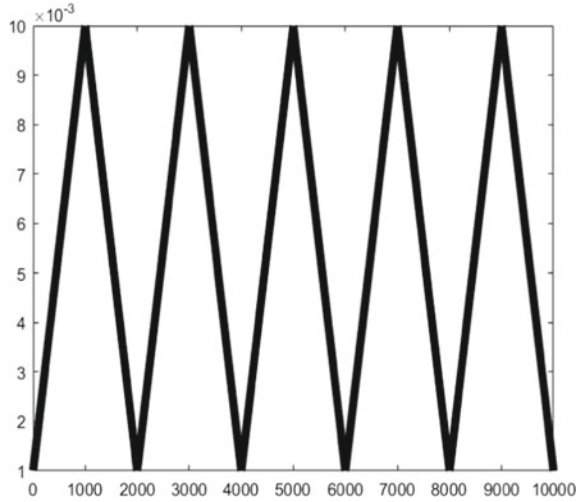


Fig. 9 Behavior lr adaptive, when Cyclical learning rate schedule is used



$$\eta_t = \eta_{base} + (\eta_{max} - \eta_{base})\max(0, (1 - x)) \tag{11}$$

Hereunder, we will detail each one of the parameters that participate in the Eq. (11), η_t is lr in its actual time, η_{base} is minimum lr (inferior starting point), η_{max} is maximum lr (superior boundary point), x is the proportion of increment or decrement in base to the *cycle* and *step size* previously defined and t that is iteration or epoch number. We can observe all these elements as is shown in Fig. 10.

As is shown in Fig. 10, there are two elements that do not appear in eq. (11), these are *cycle* and *step size*, and this is because are needed to calculate a variable x

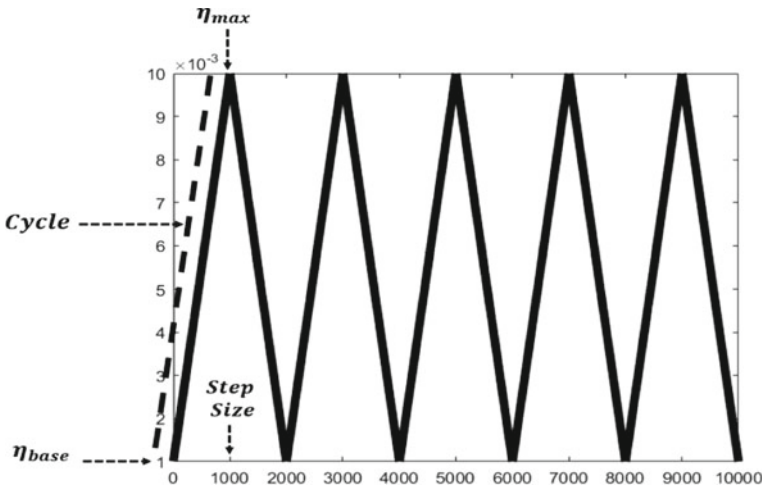


Fig. 10 Detailing all parameters of the cyclical learning rate schedule in the triangular function

(part of eq. 9). *Cycle* is the iterations or epochs numbers that going repeat until the *lr* returns to the initial point (η_{base}) and vice versa. *StepSize* is the velocity at which want to go from an inferior limit to the upper limit and vice versa, this represents the cycle half. The mathematical form to calculate the cycle is represented as is shown in Eq. (12).

$$cycle = \frac{1 + t}{2 * stepsize} \quad (12)$$

The mathematical form to calculate the parameter x is defined as is shown in Eq. (13).

$$x = \left| \left(\frac{t}{stepsize} \right) - 2 * cycle + 1 \right| \quad (13)$$

Others functions that have been applied to this heuristic are parabolic and sinusoidal, but for simplicity of its implementation in this work a triangular function was used.

2.1 Experiments Description

The experiments were done on Mamdani-based neuro-fuzzy system designed for us, this hybrid system was done to the objective to give interpretability to the decisions of the neural network through the rules resulting [20, 21]. In this system the optimization algorithm used was Gradient descent with momentum and learning rate adaptive, and we could realize that one of the critical parameters for the optimal performance of the system was the learning rate, so we also observed that finding an initial value was not an easy task.

The architecture of the Neuro-Fuzzy system is composed of four layers all of these are completely connected, the zero layer correspond to the inputs, in the layer one is defined the firing force, in this case as we use Mamdani-based, the function that was defined for this firing force was a Gaussian, being its means and standard deviations the design parameters that will be adjusted during the training and processing learning though backpropagation algorithm. In layer two has been defined the normalization of the signals coming from the previous layer, and in layer three, the output signal is calculated, this layer has another design parameter that needs to be adjusted during the training, this is a centroid. The design of this architecture can be visualized in Fig. 11.

The parameters observed on the architecture are described as follow, a_i^l is referred to inputs and in the next layers are defined as signals, where i is the number of inputs variables and in the next layers will represent as nodes, p represent the data number and l is the layer number.

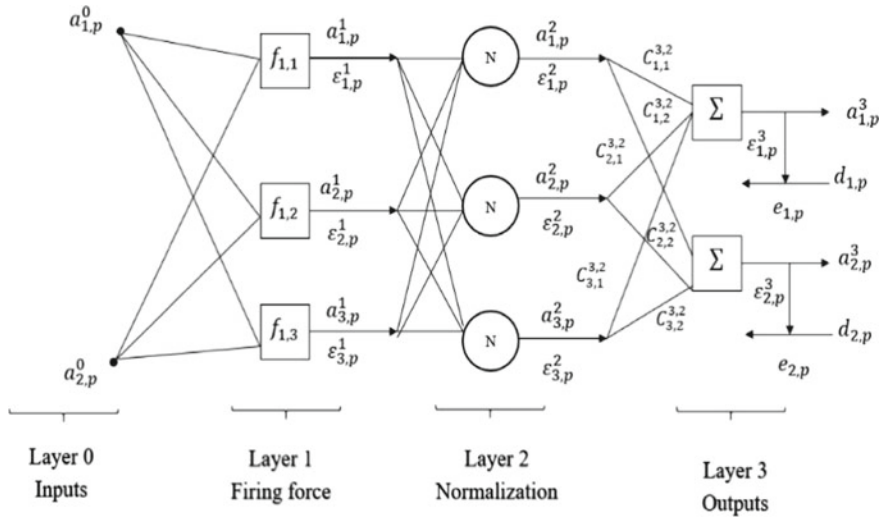


Fig. 11 The architecture of the Mamdani-based neuro-fuzzy system

In the layer one, we can observe that the form of the node represents an adaptive node, that means that its defined function ($f_{l,i}$) has implicit design parameters that will be adapted, in layer two we can observe that the node form define a fixed node because only performs normalization operations and it does not has design parameters that must be adjustment.

The last layer is an adaptive node because has a design parameter that is $C_{i-1,i}^{l,l-1}$, where $l - 1$ represent the layer before and l the current layer and $i - 1$ represent the node before and i the current node.

Finally, $d_{i,p}$ represents the estimated values obtained through to the learning process, $e_{i,p}$ is the error obtained from the difference between expected value and the obtained, and lastly $e_{i,p}^l$ that is the partial derivate with respect to all parameters.

All heuristics were used precisely to adapt the learning rate, except in the momentum method where the parameter that was changed was velocity. **Three synthetic datasets** were used to make the experimentation, and these were partitioned with the **holdout method** in 60–40% and **random sub-sampling method** where were executed **30 experiments**.

The metrics that was used to do an analysis comparative and demonstrate the effectiveness and performances between the different heuristics experimented are detailed as follow:

Hereunder, we will detail the results of each one of the datasets that were analyzed: **R** represent the correlation coefficient that allows us to know the model tend, that is, if the estimated model behavior that follows the tendency by the original model, **STD** standard deviation and **C.V.** coefficient of variation, The latter with the purpose of explaining the homogeneity or not of the sample, as well as its dispersion and variability, in addition, the minimization of error was also measured with the metric

RMSE this is the root mean squared error. In the following tables we will observe the variable \bar{R} this means that is the mean correlation coefficient and \overline{RMSE} this means that is the mean of the root mean squared error.

2.2 Results and Discussions

1. First synthetic dataset with 1 attribute (variable) and 94 instances.

The training parameter defined for the experimentation (for all heuristics) of this dataset was: $\eta = 0.01$, $\eta_{increment} = 1.04$, $\eta_{decrement} = 0.9$, $mc = 0.9$ and $rules=6$. The results obtained are shown in Table 1, the *GDX heuristic* obtained better results than the other techniques, its dispersion is controlled, and its variability is also low, this has better visualization in Fig. 12.

Table 1 Results obtained from the \bar{R} for the first dataset evaluated

Heuristics	\bar{R}	STD	C.V.
GDX	0.9703	0.0239	2.47
Time-based	0.9595	0.0325	3.39
Drop-based	0.8809	0.0265	3.01
Exponential-based	0.8874	0.0373	4.21
Cyclical-based	0.9683	0.0253	2.61
Momentum	0.9512	0.0434	4.56

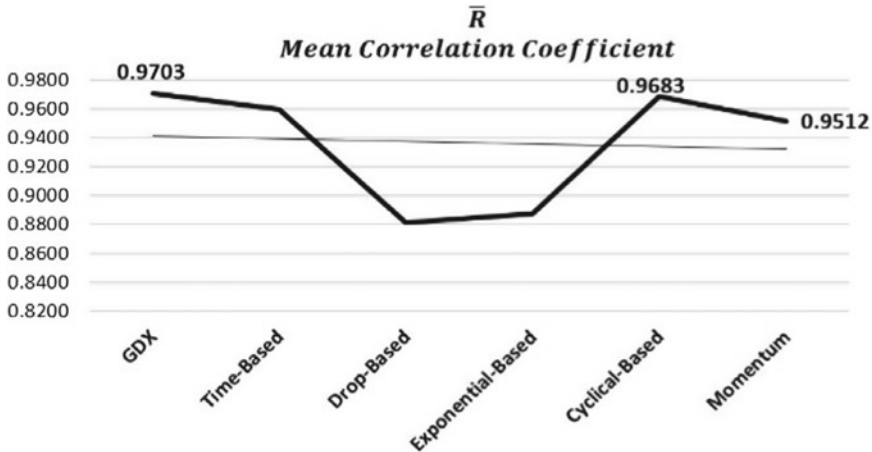


Fig. 12 Mean correlation coefficient for the first dataset

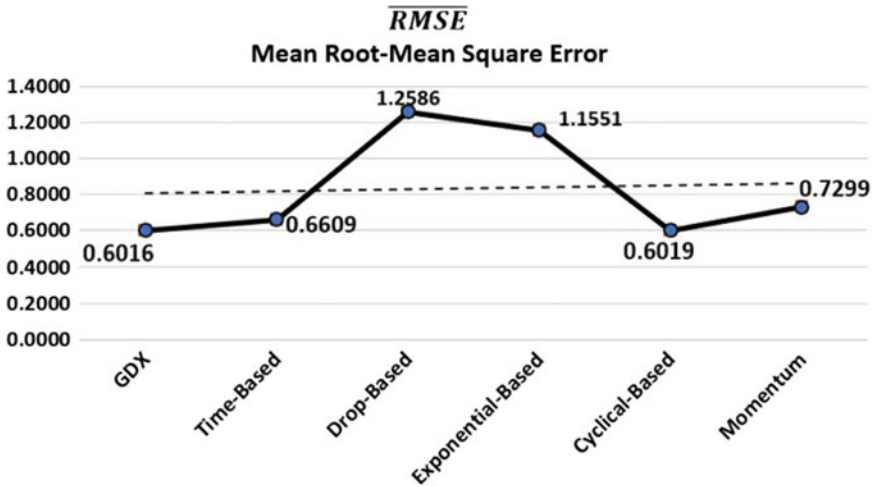


Fig. 13 Mean of the root mean squared error of this first dataset

Table 2 Results obtained from the \overline{RMSE} for the first dataset evaluated

Heuristics	\overline{RMSE}	STD	C.V.
GDX	0.6016	0.30	49.11
Time-based	0.6609	0.3781	57.21
Drop-based	1.2586	0.19	15.19
Exponential-based	1.1551	0.27	23.79
Cyclical-based	0.6019	0.31	51.07
Momentum	0.7299	0.22	30.05

However, it is observed that in the minimization of the error throughout the experimentation it is not so stable since there are great variability and dispersion, but even so, GDX shows better performance as we can observe in Fig. 13 and Table 2.

Finally, a comparative evaluation was also done, with respect to the best result obtained in its correlation coefficient (individual experiment), in how many times it was concluded and its coefficient of variability. As we can see in Fig. 14, it is the Drop-based heuristic done in fewer epochs, but if we look at Table 3, it was not the one that obtained the best R , those who obtained the best R were Exponential-based and Cyclical-based but with a high epochs number, so we can determine that again GDX is better, because it gets a good R value (as we can see in Figs. 15 and 16), in a good epochs number and with controlled variability.

2. Second synthetic dataset with 1 attribute (variable) and 250 instances.

The training parameter defined for the experimentation (for all heuristics) of this dataset was: $\eta = 0.01$, $\eta_{increment} = 1.04$, $\eta_{decrement} = 0.9$, $mc = 0.9$ and $rules=6$. The results obtained are shown in Table 4, the *GDX heuristic* obtained better results

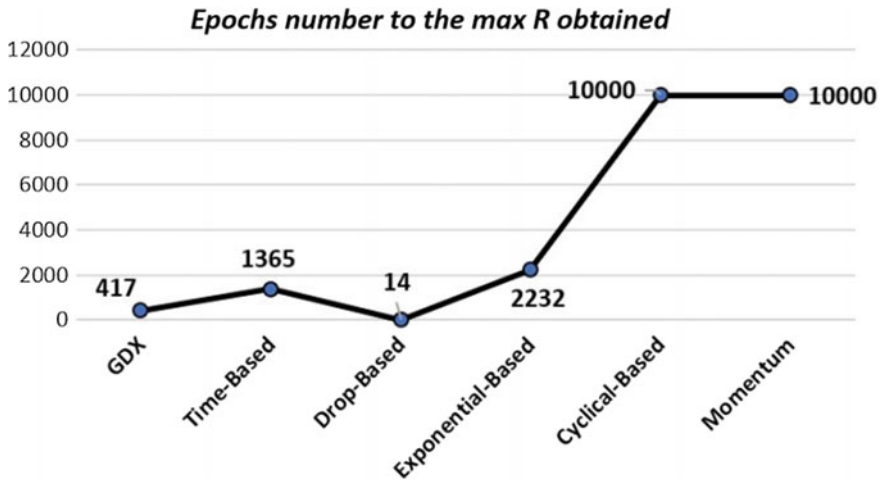


Fig. 14 Epochs number to the max *R* obtained for the first dataset

Table 3 Analysis of the best *R* obtained with its epoch number and coefficient of variation for the first dataset

Heuristics	Epochs	Max	C.V
GDX	417	0.9985	2.47
Time-based	1365	0.9995	3.39
Drop-based	14	0.8809	3.01
Exponential-based	2232	0.9999	4.21
Cyclical-based	10,000	0.9999	2.61
Momentum	10,000	0.9977	4.56

Fig. 15 The best result obtained of *R* in an individual experiment to the first dataset

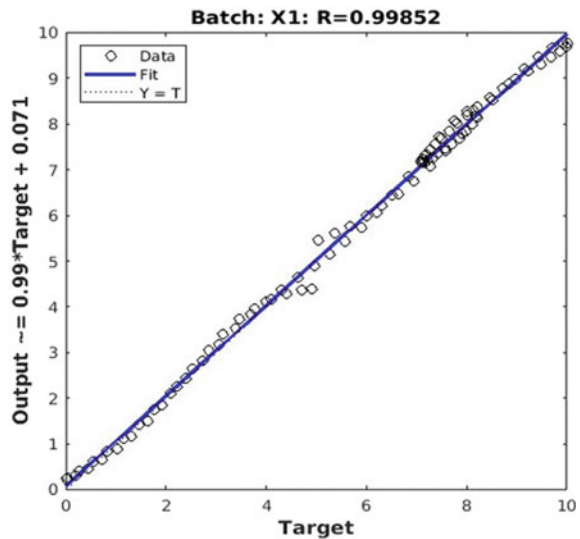


Fig. 16 The follow-up of the behavior of the estimated model with respect to the original model for the first dataset evaluated

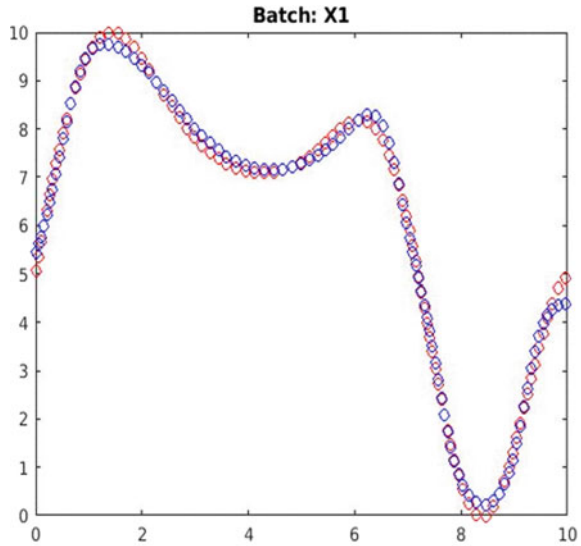


Table 4 Results obtained from the \bar{R} for the second dataset evaluated

Heuristics	\bar{R}	STD	C.V.
GDX	0.9975	0.0007	0.07
Time-based	0.9765	0.0291	2.98
Drop-based	0.9754	0.0217	2.23
Exponential-based	0.9053	0.0313	3.46
Cyclical-based	0.8904	0.0164	1.84
Momentum	0.9178	0.0351	3.83

than the other techniques, its dispersion is controlled, and its variability is also low, this has better visualization in Fig. 17.

However, it is observed that in the minimization of the error throughout the experimentation it is not so stable since there are great variability and dispersion, but even so, GDX shows better performance as we can observe in Fig. 18 and Table 5.

Finally, a comparative evaluation was also done, with respect to the best result obtained in its correlation coefficient (individual experiment), in how many times it was concluded and its coefficient of variability. As we can see in Fig. 19, it is the Cyclical-based heuristic done in fewer epochs (for long difference between the others heuristics), but if we look at Table 6, it was not the one that obtained the best R, those who obtained the best R were GDX-based, Exponential-based and Momentum-based but finishing this latter with excessive epochs number, so we can determine that again GDX is better, because it gets a good R value (as we can see in Figs. 20 and 21), in a good epochs number and with controlled variability.

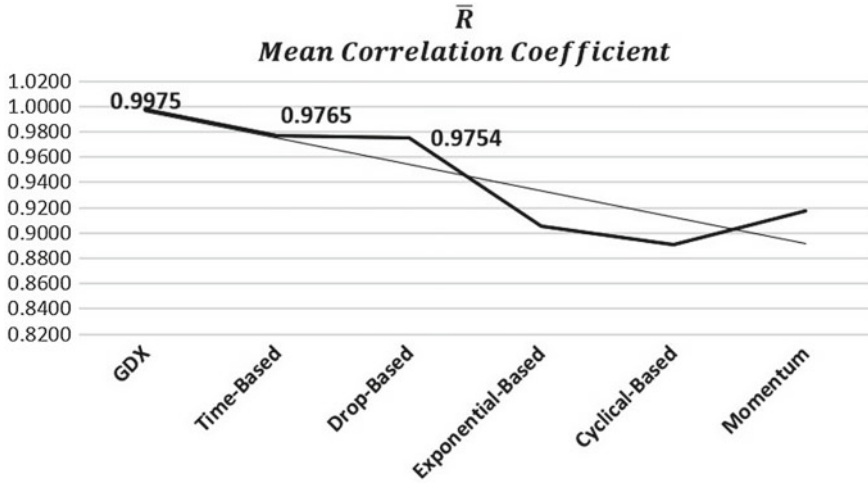


Fig. 17 Mean correlation coefficient for the second dataset

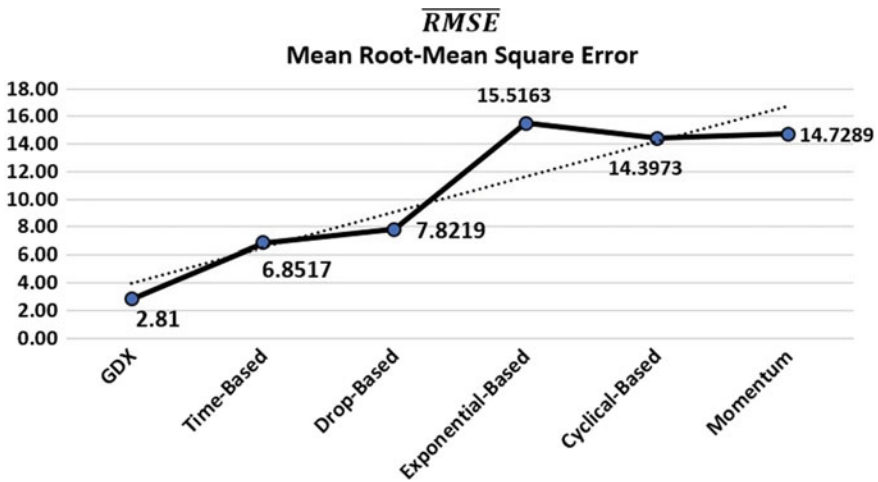


Fig. 18 Mean of the root mean squared error of this second dataset

Table 5 Results obtained from the RMSE for the second dataset evaluated

Heuristics	\overline{RMSE}	STD	C.V.
GDX	2.81	0.35	12.34
Time-based	6.8517	3.7694	55.01
Drop-based	7.8219	2.9046	37.13
Exponential-based	15.5163	3.2628	21.03
Cyclical-based	14.3973	1.7238	11.97
Momentum	14.7289	3.3451	22.71

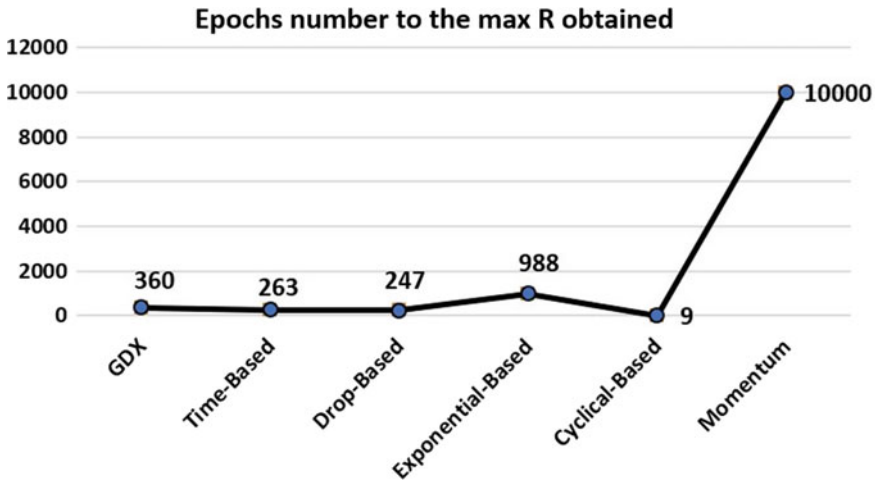


Fig. 19 Epochs number to the max R obtained for the second dataset evaluated

Table 6 Analysis of the best R obtained with its epoch number and coefficient of variation for the second dataset

Heuristics	Epochs	Max	C.V
GDX	360	0.9983	0.07
Time-based	263	0.9959	2.98
Drop-based	247	0.9754	2.23
Exponential-based	988	0.9983	3.46
Cyclical-based	9	0.9187	1.84
Momentum	10,000	0.9984	3.83

3. Third synthetic dataset with 2 attribute (variable) and 1199 instances.

The training parameter defined for the experimentation (for all heuristics) of this dataset was: $\eta = 0.0001, \eta_{increment} = 1.045, \eta_{decrement} = 0.7, mc = 0.9$ and $rules=15$. The results obtained are shown in Table 7, the Time-based heuristic obtained better R- result than the other techniques, but nevertheless, it got higher values in STD and C.V. with respect to obtained by GDX Heuristic which shows that its dispersion is controlled, and its variability is also low, this has better visualization in Fig. 22.

However, it is observed that in the minimization of the error throughout the experimentation it is not so stable since there are great variability and dispersion, nevertheless, who presented better results was Exponential-based, that shows better performance as we can observe in Fig. 23 and Table 8.

Finally, a comparative evaluation was also done, with respect to the best result obtained in its correlation coefficient (individual experiment), in how many times it was concluded and its coefficient of variability. As we can see in Fig. 24, it is the Exponential based heuristic done in fewer epochs (with a good difference between the others heuristics), but if we look at Table 9, it was not the one that obtained

Fig. 20 The best result obtained of R in an individual experiment to the second dataset evaluated

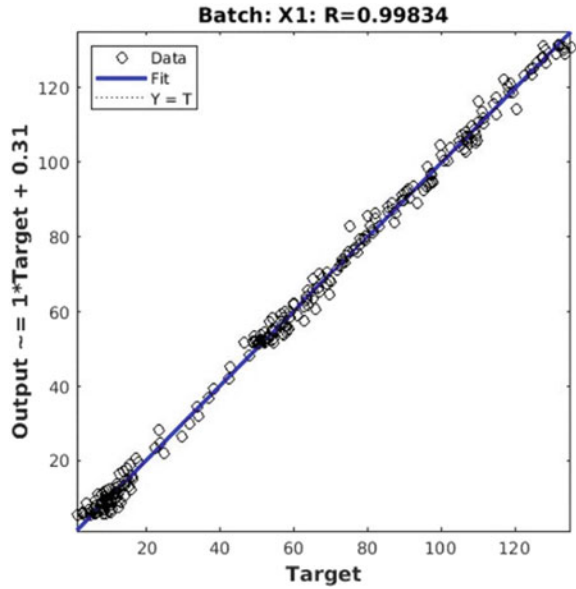
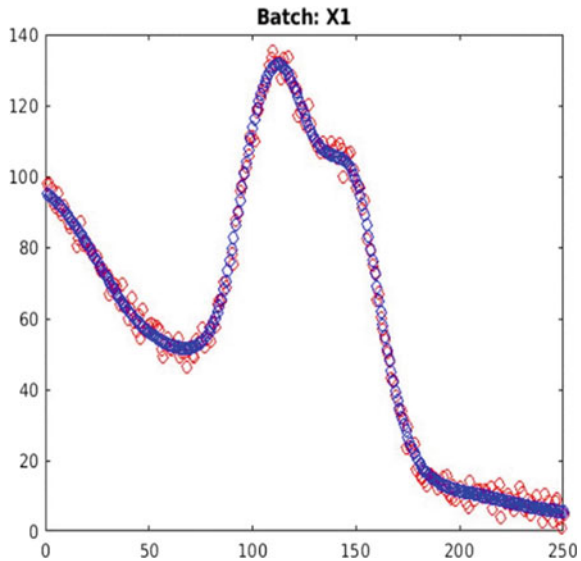


Fig. 21 The follow-up of the behavior of the estimated model with respect to the original model for the second dataset evaluated



the best R , those who obtained the best R were Cyclical-based and Time-based but finishing this latter with a lot epochs number, so we can determine that Cyclical-based is better, because it gets a good R value (as we can see in Figs. 25 and 26), in a good epochs number and with controlled variability.

Table 7 Results obtained from the \bar{R} for the third dataset evaluated

Heuristics	\bar{R}	STD	C.V.
GDX	0.9020	0.0324	3.59
Time-based	0.9215	0.0937	10.17
Drop-based	0.8938	0.0352	3.94
Exponential-based	0.6225	0.2618	42.06
Cyclical-based	0.8807	0.0577	6.55
Momentum	0.8334	0.0853	10.23

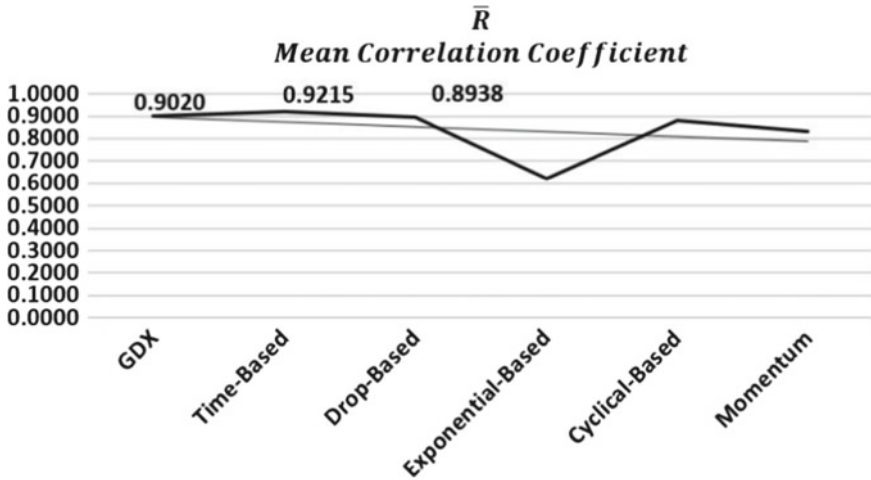


Fig. 22 Mean correlation coefficient for the third dataset evaluated

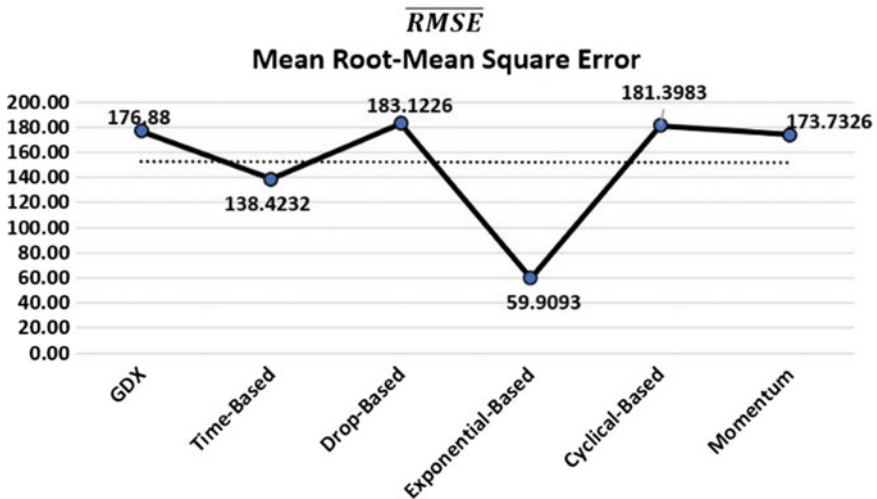


Fig. 23 Mean of the root mean squared error of this third dataset evaluated

Table 8 Results obtained from the \overline{RMSE} for the third dataset evaluated

Heuristics	\overline{RMSE}	STD	C.V.
GDX	176.88	19.02	10.75
Time-based	138.4232	48.1213	34.76
Drop-based	183.1226	29.95	16.35
Exponential-based	59.9093	37.54	62.66
Cyclical-based	181.3983	43.27	23.86
Momentum	173.7326	37.50	21.58

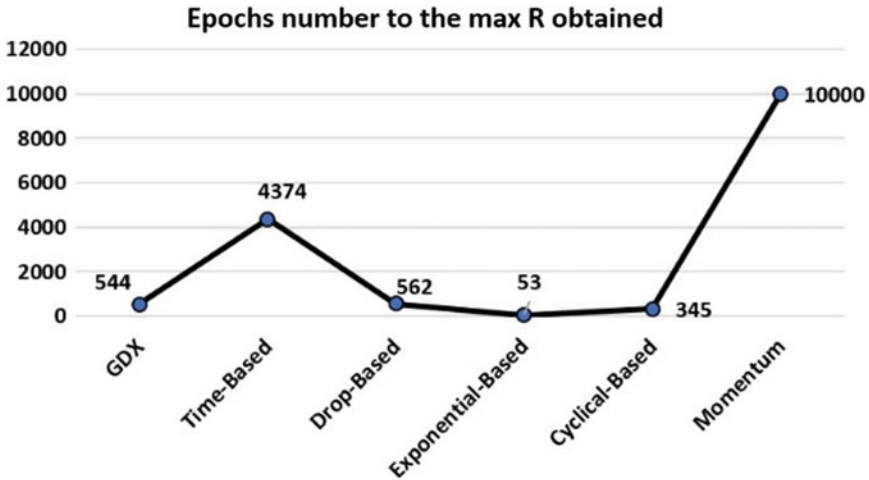


Fig. 24 Epochs number to the max R obtained for the third dataset evaluated

Table 9 Analysis of the best R obtained with its epoch number and coefficient of variation for the third dataset evaluated

Heuristics	Epochs	Max	C.V
GDX	544	0.9326	3.59
Time-based	4374	0.9849	10.17
Drop-based	562	0.8938	3.94
Exponential-based	53	0.9362	42.06
Cyclical-based	345	0.9802	6.55
Momentum	10,000	0.9350	10.23

2.3 Conclusions

As we could observe throughout the study, having the possibility of having mechanisms in our case heuristic, that allow us to navigate between different contexts and complexity level, in an efficient way and with great possibilities of arriving

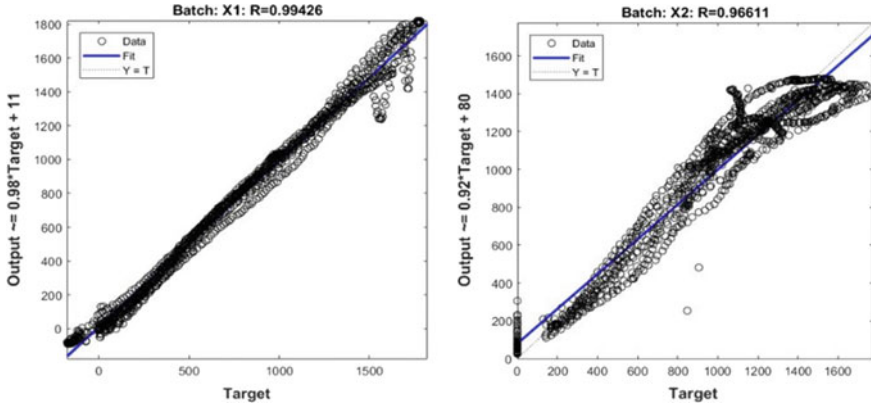


Fig. 25 The best result obtained of R in an individual experiment to the third dataset evaluated

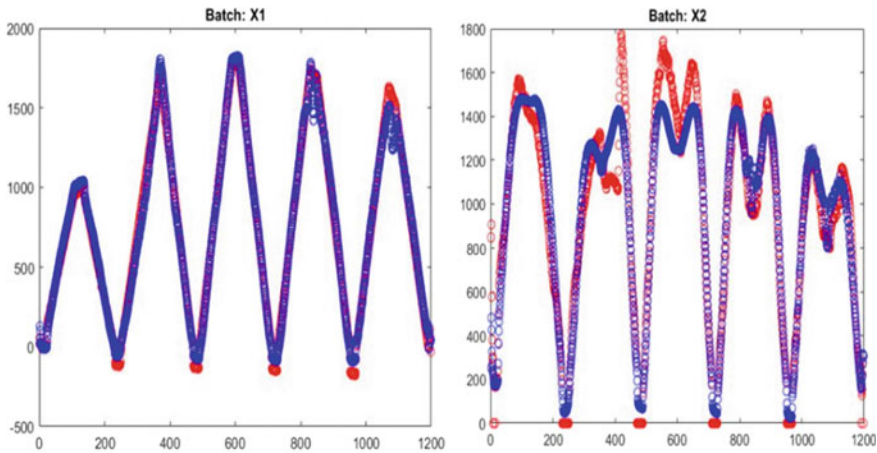


Fig. 26 The follow-up of the behavior of the estimated model with respect to the original model for the third dataset evaluated

at very approximate solutions, is of great help and is what is sought. It was mentioned above, the learning rate is one of the most critical hyperparameters when we use gradient-based learning, so different heuristics were explored to know its functionality, implementation, and behavior.

In the experimentation carried out, three datasets with synthetic data were used, the results showed that one of the heuristics with better performance in correlation coefficient, error minimization, control in the variability of their estimates was obtained by GDX heuristic. This indicates that small variations of decrease and increase in the learning rate during the change of each epoch offer great results, but the challenge to which we are then, lying in the initial parameter to achieve more smoothed changes and greater stability.

Other heuristics that showed a good performance in the achievement of results as correlation coefficient and error handling were Cyclical-based and Momentum-based heuristics, however, their execution time if it represents a limitation at the time of choosing them. For later work, we consider that using these heuristics in conjunction with other learning methods such as online and mini-batch learning can help improve their results and accelerate their convergence since the adaptation would be made during training and not until the epoch change.

References

1. Wu, X., Ward, R., Bottou, L.: WNGrad: Learn the Learning Rate in Gradient Descent. arXiv Prepr. [arXiv:1803.02865](https://arxiv.org/abs/1803.02865) (2018)
2. Im, D.J., Tao, M., Branson, K.: An empirical analysis of the optimization of deep network loss surfaces (2016)
3. Goodfellow, I.J., Vinyals, O., Saxe, A.M.: Qualitatively characterizing neural network optimization problems (2014)
4. Li, X.-M., Ouyang, J.-H.: Tuning the learning rate for stochastic variational inference. *J. Comput. Sci. Technol.* **31**(2), 428–436 (2016)
5. LeCun, Y.A., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient backprop. In: *Neural Networks: Tricks of the Trade*, pp. 9–48. Springer (2012)
6. Jang, J., Sun, C.T., Mizutani, E.: Neuro-fuzzy and soft computing—a computational approach to learning and machine intelligence. *Autom. Control. IEEE* **42**(10), 1482–1484 (1997)
7. Izui, K., Yamada, T., Nishiwaki, S., Tanaka, K.: Multiobjective optimization using an aggregative gradient-based method. *Struct. Multidiscip. Optim.* **51**(1), 173–182 (2015)
8. Liao, R., et al.: Reviving and improving recurrent back-propagation (2018)
9. Miljkovic, D., Aleksovski, D., Podpečan, V., Lavrač, N., Malle, B., Holzinger, A.: Machine learning and data mining methods for managing parkinson’s disease. In: Holzinger, A. (ed.) *Machine learning for health informatics: state-of-the-art and future challenges*, pp. 209–220. Springer International Publishing, Cham (2016)
10. Hameed, A.A., Karlik, B., Salman, M.S.: Back-propagation algorithm with variable adaptive momentum. *Knowledge-Based Syst.* **114**, 79–87 (2016)
11. Istook, E., Martinez, T.: Improved backpropagation learning in neural networks with windowed momentum. *Int. J. Neural Syst.* **12**(3), 303–318 (2002)
12. Bengio, Y.: Practical recommendations for gradient-based training of deep architectures. *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7700 LECTU, pp. 437–478 (2012)
13. Senior, A., Heigold, G., Ranzato, M., Yang, K.: An empirical study of learning rates in deep neural networks for speech recognition. In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing—Proceedings*, pp. 6724–6728 (2013)
14. Schaul, T., Zhang, S., LeCun, Y.: No more pesky learning rates (2012)
15. Hsueh, B.Y., Li, W., Wu, I.-C.: Stochastic gradient descent with hyperbolic-tangent decay on classification (2018)
16. An, W., Wang, H., Zhang, Y., Dai, Q.: Exponential decay sine wave learning rate for fast deep neural network training. *2017 IEEE Vis. Commun. Image Process*, pp. 1–4 (2017)
17. Stanford University et al.: CS231n convolutional neural networks for visual recognition, **22**(10), 1345–1359 (2016). arXiv Prepr. [arXiv:1511.07289](https://arxiv.org/abs/1511.07289)
18. Smith, L.N.: Cyclical learning rates for training neural networks. In: *Proceedings—2017 IEEE winter conference on applications of computer vision, WACV 2017*, pp. 464–472 (2017)
19. Smith, L.N., Topin, N.: Exploring loss function topology with cyclical learning rates (2017)

20. Chai, Y., Jia, L., Zhang, Z.: Mamdani model based adaptive neural fuzzy inference system and its application. *Int. J. Comput. Intell.* **5**(1), 22–29 (2009)
21. Chai, Y., Jia, L., Zhang, Z.: Mamdani model based adaptive neural fuzzy inference system and its application in traffic level of service evaluation. In: 2009 Sixth International Conference on Fuzzy Systems and Knowledge Discovery, vol. 4, pp. 555–559 (2009)

Direct and Indirect Evolutionary Designs of Artificial Neural Networks



O. Alba-Cisneros, A. Espinal, G. López-Vázquez, M. A. Sotelo-Figueroa, O. J. Purata-Sifuentes, V. Calzada-Ledesma, R. A. Vázquez and H. Rostro-González

Abstract Evolutionary Algorithms (EAs) and other kind of metaheuristics are utilized to either design or optimize the architecture of Artificial Neural Networks (ANNs) in order to adapt them for solving a specific problem; these generated ANNs are known as Evolutionary Artificial Neural Networks (EANNs). Their architecture components, including number of neurons or their kind of transfer functions, connectivity pattern, etc., can be defined through direct or indirect encoding schemes; the former, directly codifies ANN architecture components into the genotype of solutions, meanwhile the last one, requires to transform the solution's genotype through mapping processes to generate an ANN architecture. In this work, architectures of three-layered feed-forward ANNs are designed by using both encoding schemes to solve several well-known benchmark datasets of supervised classification problems. The results of designed ANNs by using direct and indirect encoding schemes are compared.

Keywords Artificial neural networks · Differential evolution · Direct encoding scheme · Indirect encoding scheme · Pattern classification

O. Alba-Cisneros · A. Espinal (✉) · M. A. Sotelo-Figueroa · O. J. Purata-Sifuentes
Departamento de Estudios Organizacionales, DCEA-Universidad de Guanajuato,
Guanajuato, Mexico
e-mail: aespinal@ugto.mx

G. López-Vázquez
División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México / Instituto
Tecnológico de León, León, Guanajuato, Mexico

V. Calzada-Ledesma
Tecnológico Nacional de México / Instituto Tecnológico Superior de Purísima del Rincón.
Purísima del Rincón, Guanajuato, Mexico

R. A. Vázquez
Grupo de Sistemas Inteligentes, Facultad de Ingeniería, Universidad de La Salle,
Delegación Cuauhtémoc, Ciudad de México, Mexico

H. Rostro-González
Departamento de Electrónica, DICIS-Universidad de Guanajuato,
Salamanca, Guanajuato, Mexico

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_31

1 Introduction

Artificial Neural Networks (ANNs) are mathematical models that mimic in some degree the neuronal dynamics observed in brains of living beings. They have been successfully applied in different disciplines both theoretical and practical to solve problems such as pattern recognition [28], forecasting [27], etc. With respect of an ANN's performance, it is well-known that it is highly related with its architecture or topology; which is mainly defined around three components: computing units, communication links and message type [17].

The designing process of an ANN's architecture is commonly performed by human experts that define several of its structural and other kind of components through rules of thumb, heuristics, by trial-and-error and expertise. However, such process may be time consuming or represent a challenge; e.g. combinatorial problems may arise when designing it [1] or issues related to learning an specific task given a fixed architecture of an ANN [2, 3, 16, 17].

The ANNs' designing and learning problems may be partially or completely overcome by using metaheuristic algorithms; these ANNs are known as Evolutionary Artificial Neural Networks (EANNs). Metaheuristics can be used as a replacement of traditional learning rules or to improve their learning speed, among other aspects. Besides, they can be implemented as mechanisms to design or modify architectures of ANNs [8, 24, 26]. The use of such algorithms allow to prescind in some manner of human experts in ANN's design phase.

Focusing in design of ANNs' architectures by means of metaheuristic algorithms, there can be distinguished two encoding schemes to represent architecture components of an ANN; which are: direct and indirect [6, 20]. Basically, the former scheme codifies ANN architecture components into the genotype of solutions, meanwhile, the last one requires to transform the solution's genotype through mapping processes to generate an ANN architecture.

In this paper, there are developed three-layered feed-forward ANNs with partial synaptic connections between their layers though evolutionary metaheuristics by using both direct and indirect encoding schemes. The partially-connected feed-forward EANNs contain a subset of the possible synaptic connections of their counterpart fully-interlayer-connected; which are smaller topologies that can have similar or better performances than their fully-connected version [7]. Besides, jointly with the evolutionary design process it is possible to attend some suggestions to ease the ANNs' learning such as constraint either the ANNs' architecture or the problem to deal with, or to have a learning algorithm that modifies the ANNs' architecture during training process [1, 17]. The rest of the paper is organized as follows. In Sect. 2, there are described the concepts required for the development of this work. The implementation details of designing methodologies using direct and indirect encoding schemes are explained in Sect. 3. The experimental configuration and obtained results are shown in Sect. 4. Finally, Sect. 5 presents the conclusions and future work.

2 Background

2.1 Artificial Neural Networks

Artificial Neural Networks (ANNs) are bioinspired models constituted by computing units (neurons) interconnected in a wiring pattern (synaptic connections) sharing messages of some type (neural activity) [17]; such elements define the architecture or topology of an ANN. Focusing in a three-layered feed-forward ANN, it can be considered as a system that maps an input signal ($x \in \mathbb{R}^N$) into an output signal ($y \in \mathbb{R}^M$) [15]; such mapping process is carried out by sending x through input, hidden and output layers. Each layer has computing units connected to other units in the contiguous layer; such connections can be full or partial. The number of computing units in input and output layers can be defined according the size of input and output signals; N and M nodes respectively. However, there is not a well-established way to define the number of units that hidden layer must have; it is usually define by trial-and-error, heuristics or human expertise.

Computing units in input layer passes directly their signal to nodes in hidden layer. Meanwhile, each unit in next two layers calculates and evaluates its input to produce its output. The input inp_i of i -th computing unit is commonly calculated by Eq. 1

$$inp_i = b_i + s \cdot w_i \quad (1)$$

where b_i is the bias, s is the input signal and w_i is the synaptic weight vector. The output of i -th unit is produced by means of an activation function $f_i(inp_i)$; which can be linear or nonlinear [14] and it is usually chosen according the problem to solve.

2.2 Differential Evolution

Differential Evolution (DE) [23] is an evolutionary metaheuristic for solving d -dimensional real-valued optimization problems. In DE, a population of NP solutions of d -dimensions is uniformly create over the search space and evaluated according a fitness function. For each individual x_i in the population is generated a mutant vector v_i by using Eq. 2

$$v_i = x_{r_1} + F(x_{r_2} - x_{r_3}) \quad (2)$$

where r_1, r_2 and r_3 are three randomly selected indexes ($i \neq r_1 \neq r_2 \neq r_3$ and $r_1, r_2, r_3 \in \{1, \dots, NP\}$) and $F \in (0, 2]$ is a differential weight. Next, a trial vector u_i is created based on x_i and v_i by using Eq. 3

$$u_{j,i} = \begin{cases} v_{j,i} & \text{if } r_j \leq C_r \text{ or } j = J_r \\ x_{j,i} & \text{if } r_j > C_r \text{ or } j \neq J_r \end{cases} \quad (3)$$

where $j = 1, \dots, d$, $Cr \in [0, 1]$ is a crossover rate, $J_r \in \{1, \dots, d\}$ is a random index generated for current individual i and $r_j \in U[0, 1]$ is an uniform random number generated for each dimension. An individual is selected for the next generation according to Eq. 4

$$x_i = \begin{cases} u_i & \text{if } f(u_i) \leq f(x_i) \\ x_i & \text{otherwise} \end{cases} \quad (4)$$

where $f(\bullet)$ is the fitness function. This process is repeated until a stop criterion is met.

2.3 Grammatical Evolution

Grammatical Evolution (GE) [22] is a grammar-based form of Genetic Programming [18]. In GE, the genotypic representation of individuals is formed by strings of numbers that are changed to the functional phenotypic representation through a mapping process [4] (also known as indirect representation). The mapping process uses a Context-Free Grammar (CFG) in Backus-Naur Form (BNF) to derive the phenotypic representation of an individual. A metaheuristic is used to perform the search process by modifying the genotype of the individual with only the knowledge of its fitness value.

A CFG is defined as 4-tuple as follows [19]: $G = \{\Sigma_N, \Sigma_T, P, S\}$ where Σ_T is the set of terminals, Σ_N is the set of non-terminals, $S \in \Sigma_N$ is the start symbol and P is the productions set; A production is usually written $A \rightarrow \alpha$. The BNF is a notation used to describe productions of CFGs as follows [25]:

- The symbol \models is used instead of \rightarrow .
- Every non-terminal symbol is enclosed in brackets of the form $\langle \rangle$.
- All production with the same non-terminal in the left-hand side members are combined into one statement with all the right-hand side members, separated by vertical bars ($|$).

The canonical GE's mapping process, the Depth-first [22]; is used to choose the next production based on the current left-most non-terminal symbol in the expression. The Depth-first mapping process is given in Eq. (5):

$$Rule = c\%r \quad (5)$$

where c is the current component value of individual's genotype and r is the number of production rules available for the current non-terminal. The selected production is used to replace the current left-most non-terminal in the word being derived.

3 Design Methodologies

The methodologies’ purpose is to design three-layered feed-forward ANNs by means of evolutionary algorithms through direct and indirect encoding schemes. The methodologies have the following considerations to design ANNs’ architectures:

- Computing units in hidden layer. The methodologies must define in some manner the number of computing units in hidden layer.
- Activation functions of units in hidden and output layers. The available activation functions for designing process are shown in Table 1.
- Synaptic connections and weights. At the end of designing process, the result must be a designed ANN with its connections defined and weights calibrated.

3.1 Design Methodology with Direct Codification

The ANN design methodology with direct codification is based on Garro’s works [9–13], where there were designed three-layered feed-forward ANNs with inter and supra layer connections and variable activation function types; i.e. computing units of input layers can be connected to units of hidden and output layers and units of hidden layer can only be connected to units in output layer. In this work, there are only developed three-layered feed-forward ANNs with inter layer connections. The number of computing units Q in the whole ANN is defined by Eq. 6

$$Q = (N + M) + \frac{(N + M)}{2} \tag{6}$$

Table 1 Activation functions

Name	ID	Label	Function
Logistic	0	LS	$f(inp) = \frac{1}{1+e^{-inp}}$
Hyperbolic Tangent	1	HT	$f(inp) = \frac{e^{inp} - e^{-inp}}{e^{inp} + e^{-inp}}$
Sinusoidal	2	SN	$f(inp) = \sin(inp)$
Gaussian	3	GS	$f(inp) = e^{-inp^2}$
Linear	4	LN	$f(inp) = inp$
Hard limit	5	HL	$f(inp) = \begin{cases} 1 & \text{if } inp \geq 0 \\ 0 & \text{otherwise} \end{cases}$
Rectified linear	6	RL	$f(inp) = \begin{cases} x & \text{if } inp > 0 \\ 0 & \text{otherwise} \end{cases}$

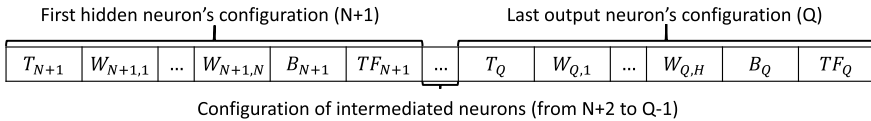


Fig. 1 Scheme for representing the design parameters of an ANN using the direct encoding scheme

where N and M are the size of input and output vector, respectively. Then, the number of computing units in the hidden layer is given by Eq. 7.

$$H = Q - (N + M) \tag{7}$$

In the adapted version for this work, the number of parameters dim_d for the designing process is calculated by Eq. 8; the codification rule is as explained in [13] (see Fig. 1), but there are only considered parameters of computing units in hidden and output layers. For each computing units there are codified next parameters: topology (presynaptic connections) T_{index} , set of synaptic weights W_{index} , bias B_{index} and activation function TF_{index} .

$$dim_d = [H * (N + 3)] + [M * (H + 3)] \tag{8}$$

3.2 Design Methodology with Indirect Codification

The ANN design methodology with indirect codification is based on Quiroz’s works [21], where there were designed three-layered feed-forward ANNs with inter layer connections with logistic activation functions. In this work, there are only developed three-layered feed-forward ANNs with inter layer connections and variable activation function types. In Fig. 2 is show the generic indirect representation of an ANN’s architecture, it has two main parts: hidden layer configuration and output layer configuration. The former allows to define the configuration of computing units in hidden layer without directly limiting the number of units; the configuration of hidden units includes: activation function, bias weight, presynaptic connection and weights. The last part defines the activation functions and bias weights of output units.

The CFG in BNF which allows to deploy expressions like In Fig. 2 is given in In Grammar. 1.1.

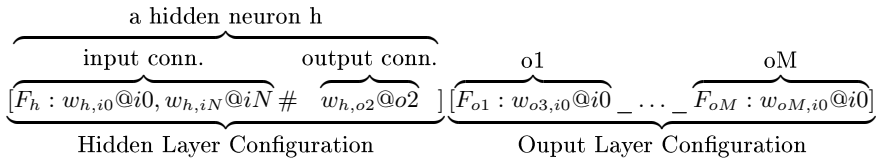


Fig. 2 Scheme for representing the design parameters of an ANN using the indirect encoding scheme

```

(network) = [(hiddenNeurons)][(outputNeurons)]
(hiddenNeurons) = (hiddenNeuron) | (hiddenNeuron)_(hiddenNeurons)
(hiddenNeuron) = (func):(weight)@i0,(inputs)#(outputs)
(outputNeurons) = (func):(weight)@i0..._ (func):(weight)@i0
(func) = LS | HT | SN | GS | LN | HL | RL
(inputs) = (input) | (input),(inputs)
(outputs) = (output) | (output),(outputs)
(input) = (weight)@(inputID)
(output) = (weight)@(outputID)
(inputID) = i1 | ... | iN
(outputID) = o1 | ... | oM
(weight) = (sign)(digitList).(digitList)
(sign) = + | -
(digitList) = (digit) | (digit)(digitList)
(digit) = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Grammar 1.1 BNF grammar for designing second-generation ANNs, as in [21]

4 Experiments and Results

The ANNs' methodologies design with direct and indirect encoding schemes were tested to assess the quality of generated EANNs. In Table 2 there are shown the well-

Table 2 Description of supervised classification benchmark datasets (obtained from [5])

Name	# Features	# Classes	# Instances
Balance	4	3	625
BCW	9	2	683
Glass	9	6	214
Ionosphere	33	2	351
Iris Plant	4	3	150
Wine	13	3	178

known supervised classification benchmarks that were used to test and compare both methodologies.

For design methodology with direct encoding scheme there was DE to drive the optimization process. Meanwhile, for the design methodology with indirect encoding scheme there were used GE with DE as search engine, the Depth-first mapping process and Grammar. 1.1 for the designing process. Both designing methodologies use the classification error rate as fitness function. The common configuration parameters of DE is as follows:

- Population size $NP = 100$.
- $F = 0.8$.
- $Cr = 0.9$.
- 1000000 function calls.

The dimension of individuals for methodology with direct encoding scheme is calculated by Eq. 8 and for the methodology with indirect encoding scheme the dimension was set to 500.

For each methodology, there were executed 35 runs per dataset; the obtained performance results in designing and testing phases for both methodologies are shown in Table 3. In bold font are marked the median value as representative result, it can be observed that methodology with direct encoding scheme got better results for most benchmark datasets in both phases.

In Figs. 3 and 4 are shown the best representative architectures obtained for BCW dataset by methodology with direct and indirect encoding schemes; respectively. Finally, there is presented a summary of architecture's characteristic obtained by both methodologies in Table 4, columns labeled as **Feat.** show the mean and standard deviation of used input features, columns labeled as **Hidden** show the mean and standard deviation the number of hidden computing units used or generated and columns labeled as **Conn.** show the mean and standard deviation of the total synaptic connections used; in this case, there can be observed that architectures generated with the methodology with indirect encoding scheme are more compact than those designed with the direct encoding scheme.

5 Conclusions and Future Work

In this work are designed EANNs through evolutionary algorithms by using direct and indirect encoding schemes. The results of designs over several well-known supervised classification benchmark dataset were numerically compared. The results show that methodology with direct encoding scheme achieves better performance than the other methodology; however the designs generated by the methodology with indirect encoding scheme are more compact than those designed through direct encoding scheme. Both methodologies show capabilities for reducing the size of input vector and allow to practically prescind of a human expert.

Table 3 Designing and testing performances obtained using direct and indirect encoding schemes for EANNs

	Direct Codification						Indirect Codification					
	Dataset	Min	Median	Max	$\mu \pm \sigma$	Min	Median	Max	$\mu \pm \sigma$			
Designing	Balance	0.77	0.86	0.92	0.86 ± 0.04	0.70	0.73	0.83	0.73 ± 0.02			
	BCW	0.95	0.98	0.99	0.97 ± 0.01	0.94	0.96	0.97	0.96 ± 0.01			
	Glass	0.45	0.47	0.51	0.47 ± 0.02	0.45	0.48	0.51	0.47 ± 0.01			
	Ionosphere	0.75	0.77	0.81	0.77 ± 0.02	0.81	0.83	0.87	0.84 ± 0.01			
	Iris Plant	0.84	0.92	0.99	0.92 ± 0.03	0.67	0.68	0.88	0.71 ± 0.05			
	Wine	0.66	0.72	0.81	0.72 ± 0.03	0.65	0.69	0.72	0.69 ± 0.02			
Testing	Balance	0.73	0.85	0.91	0.84 ± 0.04	0.64	0.70	0.80	0.71 ± 0.04			
	BCW	0.94	0.96	0.99	0.96 ± 0.01	0.92	0.95	0.98	0.95 ± 0.01			
	Glass	0.28	0.42	0.50	0.42 ± 0.04	0.32	0.45	0.48	0.43 ± 0.04			
	Ionosphere	0.67	0.73	0.80	0.73 ± 0.03	0.74	0.81	0.86	0.81 ± 0.03			
	Iris Plant	0.76	0.91	0.97	0.89 ± 0.06	0.63	0.67	0.95	0.69 ± 0.07			
	Wine	0.53	0.68	0.87	0.68 ± 0.08	0.59	0.67	0.70	0.67 ± 0.02			

Fig. 3 Representative ANN's architecture for BCW dataset obtained by using direct encoding scheme

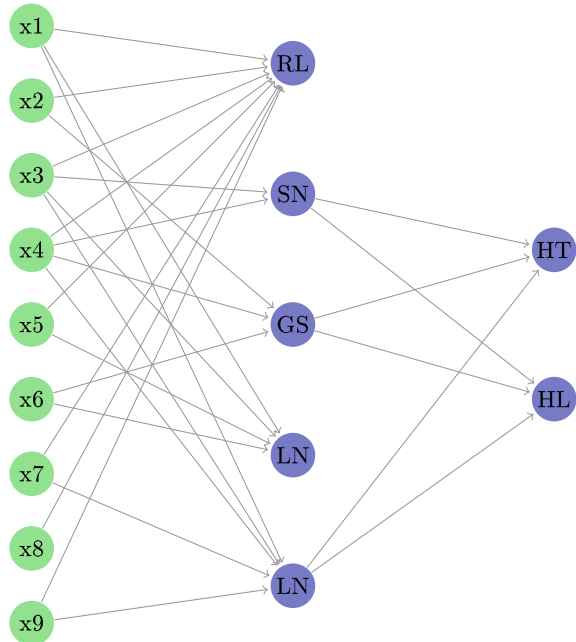


Fig. 4 Representative ANN's architecture for BCW dataset obtained by using indirect encoding scheme

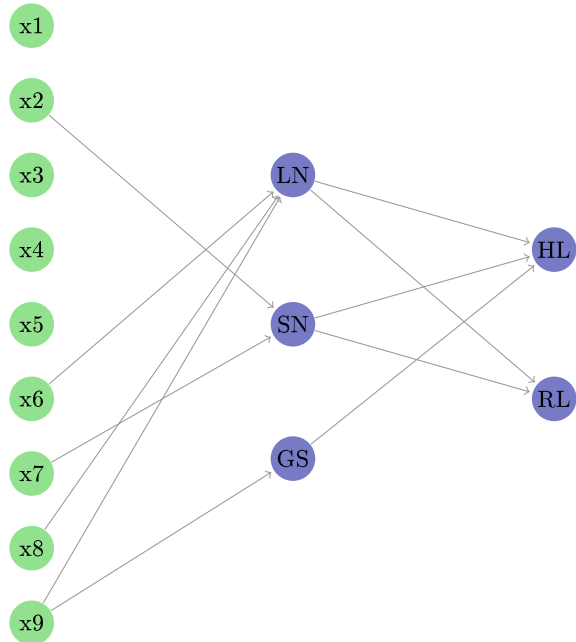


Table 4 Summary of input features used, number of hidden neurons and number of synaptic connections in architectures of generated ANNs

Dataset	Direct codification			Indirect codification		
	Feat.	Hidden	Conn.	Feat.	Hidden	Conn.
Balance	3.97 ± 0.17	3.00 ± 0.00	17.57 ± 1.67	3.20 ± 0.72	1.97 ± 1.07	12.89 ± 4.01
BCW	8.54 ± 0.70	5.00 ± 0.00	33.43 ± 3.97	3.86 ± 1.80	2.06 ± 1.53	12.06 ± 6.58
Glass	8.83 ± 0.38	7.00 ± 0.00	63.69 ± 5.79	3.49 ± 1.88	2.14 ± 1.26	17.20 ± 6.13
Ionosphere	31.00 ± 0.00	17.00 ± 0.00	469.29 ± 38.72	3.83 ± 3.28	1.97 ± 1.29	11.2 ± 5.96
Iris Plant	3.40 ± 0.69	3.00 ± 0.00	16.97 ± 1.54	2.20 ± 1.11	1.97 ± 1.34	11.83 ± 5.66
Wine	13.00 ± 0.00	8.00 ± 0.00	75.14 ± 5.60	3.09 ± 2.09	2.00 ± 1.06	12.23 ± 5.00

As future work, authors attempt to combine good features of both encoding schemes to improve ANNs' designs, generalization capabilities and designing times. To test these methodologies with other architectures and kind of ANNs. Besides, to test with other classification problems and another type of problems.

Acknowledgements Authors wish to thank Universidad de Guanajuato, Tecnológico Nacional de México y Universidad de la Salle. This work was supported by the CONACyT Project FC2016-1961 "Neurociencia Computacional: de la teoría al desarrollo de sistemas neuromórficos".

References

1. Amaldi, E., Mayoraz, E., de Werra, D.: A review of combinatorial problems arising in feed-forward neural network design. *Discrete Appl. Math.* **52**(2), 111–138 (1994)
2. Blum, A.L., Rivest, R.L.: Training a 3-node neural network is NP-complete. *Neural Netw.* **5**(1), 117–127 (1992)
3. DasGupta, B., Siegelmann, H.T., Sontag, E.: On the intractability of loading neural networks. In: *Theoretical Advances in Neural Computation and Learning*, pp. 357–389. Springer, Heidelberg (1994)
4. Dempsey, I., O'Neill, M., Brabazon, A.: *Foundations In Grammatical Evolution For Dynamic Environments*, vol. 194. Springer, Heidelberg (2009)
5. Dheeru, D., Karra Taniskidou, E.: *UCI Machine Learning Repository* (2017)
6. Ding, S., Li, H., Su, C., Yu, J., Jin, F.: Evolutionary artificial neural networks: a review. *Artif. Intell. Rev.* **39**(3), 251–260 (2013)
7. Elizondo, D., Fiesler, E.: A survey of partially connected neural networks. *Int. J. Neural Syst.* **8**(5–6), 535–558 (1997)
8. Floreano, D., Dürr, P., Mattiussi, C.: Neuroevolution: from architectures to learning. *Evol. Intell.* **1**(1), 47–62 (2008)
9. Garro, B.A., Sossa, H., Vázquez, R.A.: Design of artificial neural networks using a modified particle swarm optimization algorithm. In: *Proceedings Of The 2009 International Joint Conference On Neural Networks, IJCNN'09*, pp. 2363–2370, Piscataway, NJ, USA, IEEE Press (2009)
10. Garro, B.A., Sossa, H., Vázquez, R.A.: Design of artificial neural networks using differential evolution algorithm. In: *International Conference on Neural Information Processing*, pp. 201–208. Springer, Heidelberg (2010)
11. Garro, B.A., Sossa, H., Vázquez, R.A.: Artificial neural network synthesis by means of artificial bee colony (abc) algorithm. In: *2011 IEEE Congress of Evolutionary Computation (CEC)*, pp. 331–338. IEEE (2011)
12. Garro, B.A., Sossa, H., Vázquez, R.A.: Evolving neural networks: a comparison between differential evolution and particle swarm optimization. In: *International Conference in Swarm Intelligence*, pp. 447–454. Springer, Heidelberg (2011)
13. Garro, B.A., Vázquez, R.A.: Designing artificial neural networks using particle swarm optimization algorithms. *Comput. Intell. Neurosci.* **2015**, 61 (2015)
14. Hagan, M., Demuth, H., Beale, M.: *Neural Network Design*. Martin Hagan (2014)
15. Haykin, S.: *Neural Networks: Comprehensive Foundation*. Prentice Hall (1999)
16. Judd, J.S.: On the complexity of loading shallow neural networks. *J. Complexity* **4**(3), 177–192 (1988)
17. Judd, J.S.: *Neural Network Design and the Complexity of Learning*. Neural Network Modeling and Connectionism Series. MIT press, Cambridge (MA) (1990)
18. Koza, J.R.: *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. A Bradford Book, vol. 1. MIT press, Cambridge (MA) (1992)

19. Meduna, A.: Automata and Languages: Theory and Applications. Springer, London (2012)
20. Ojha, V.K., Abraham, A., Snášel, V.: Metaheuristic design of feedforward neural networks: a review of two decades of research. *Eng. Appl. Artif. Intell.* **60**, 97–116 (2017)
21. Quiroz-Ramírez, O., Espinal, A., Ornelas-Rodríguez, M., Rojas-Domínguez, A., Sánchez, D., Puga-Soberanes, H., Carpio, M., Espinoza, L.E.M., Ortíz-López, J.: Partially-connected artificial neural networks developed by grammatical evolution for pattern recognition problems. *Stud. Comput. Intell.* **749**, 99–112 (2018)
22. Ryan, C., Collins, J., O’Neill, M.: Grammatical evolution: evolving programs for an arbitrary language. *Genetic Programming: First European Workshop. EuroGP’98 Paris, France, 14–15 April 1998 Proceedings*, pp. 83–96. Springer, Heidelberg (1998)
23. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optimization* **11**(4), 341–359 (1997)
24. Tayefeh, M., Taghiyareh, F., Forouzideh, N., Caro, L.: Evolving artificial neural network structure using grammar encoding and colonial competitive algorithm. *Neural Comput. Appl.* **22**(1), 1–16 (2013)
25. Veerarajan, T.: Discrete Mathematics. McGraw-Hill Education (India) Pvt Limited (2006)
26. Yao, X.: Evolving artificial neural networks. *Proc. IEEE* **87**(9), 1423–1447 (1999)
27. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: the state of the art. *Int. J. Forecast.* **14**(1), 35–62 (1998)
28. Zhang, G.P.: Neural networks for classification: a survey. *IEEE Trans. Syst. Man Cybern. Part C (Applications and Reviews)* **30**(4), 451–462 (2000)

Studying Grammatical Evolution's Mapping Processes for Symbolic Regression Problems



B. V. Zuñiga-Nuñez, J. Martín Carpio, M. A. Sotelo-Figueroa, J. A. Soria-Alcaraz, O. J. Purata-Sifuentes, Manuel Ornelas and A. Rojas-Domínguez

Abstract Grammatical Evolution (GE) is a variant of Genetic Programming (GP) that uses a BNF-grammar to create syntactically correct solutions. GE is composed of the following components: the Problem Instance, the BNF-grammar (BNF), the Search Engine (SE) and the Mapping Process (MP). GE allows creating a distinction between the solution and search spaces using an MP and the BNF to translate from genotype to phenotype, that avoids invalid solutions that can be obtained with GP. One genotype can generate different phenotypes using a different MP. There exist at least three MPs widely used in the art-state: Depth-first (DF), Breadth-first (BF) and π Grammatical Evolution (piGE). In the present work DF, BF, and piGE have been studied in the Symbolic Regression Problem. The results were compared using a statistical test to determine which MP gives the best results.

Keywords Grammatical evolution · Symbolic problem · Mapping process

B. V. Zuñiga-Nuñez · J. M. Carpio · M. Ornelas · A. Rojas-Domínguez
Departamento de Estudios de Posgrado e Investigación, Tecnológico Nacional de México,
Instituto Tecnológico de León, Avenida Tecnológico S/N, 37290 León, GTO, Mexico
e-mail: m18240006@itleon.edu.mx

J. M. Carpio
e-mail: jmcarpio61@hotmail.com

M. Ornelas
e-mail: mornelas67@yahoo.com.mx

A. Rojas-Domínguez
e-mail: alfonso.rojas@gmail.com

M. A. Sotelo-Figueroa (✉) · J. A. Soria-Alcaraz · O. J. Purata-Sifuentes
División de Ciencias Económico Administrativas, Departamento de Estudios Organizacionales,
Universidad de Guanajuato, Fraccionamiento I El Establo, 36250 Guanajuato, GTO, Mexico
e-mail: masotelo@ugto.mx

J. A. Soria-Alcaraz
e-mail: jorge.soria@ugto.mx

O. J. Purata-Sifuentes
e-mail: opurata@ugto.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_32

1 Introduction

Automatic Programming (AP) [1, 2] has been defined as a program that can construct other programs by itself. One of the AP's elements [2] is the application domain, the output is an efficient program that satisfies the requirements.

Genetic Programming (GP) [1] is an AP technique proposed by Koza in 1998, it takes inspiration from nature and genetics. GP uses a tree representation [3], it starts with an initial population of trees and each one represents candidate solutions to a specific problem, and each one of these individuals is evaluated according to its performance to solve the problem employing an objective function. Each individual is evolved using the Genetic Algorithm operators [4], like the selection, crossover, and mutation.

Grammatical Evolution (GE) [5] is GP based form. GE uses a BNF-Grammar to generate syntactically correct solutions specifying the rules or restrictions that are necessary to create the solution. The individuals are a binary string that is evolved by a Search Engine (SE) [5], this allows making a distinction between the search space made by the individuals and the solution space created by the mapping process; thus this is possible to replace the SE used [6–8].

The main components of GE are the Problem Instance, the Search Engine, and the MP [7]. Each one of these represents a research area in itself and could be replaced. Different studies have been done in the field with the purpose of investigating the performance of using various SE like Differential Evolution (DE) [9], Estimation Distribution Algorithm (EDA) [8], Genetic Algorithm (GA) [10], Particle Swarm Optimization (PSO) [7] and Particle Evolutionary Swarm Optimization (PESO) [7]; as well as different types of grammars [11, 12].

The Mapping Process (MP) has the responsibility of making the genotype to phenotype mapping; it is possible to create different phenotypes from the same genotype just by changing the MP. There exist earlier research in the field of MPs, in [13] were studied 4 alternatives MPs: Depth-first (DF) [14], Breadth-first (BF) [15], π Grammatical Evolution (piGE) [16] and a Random control strategy applied to 4 benchmark problems; in [15] a two main MPs investigation between the DF and piGE processes is presented, but also another two MPs were studied. In both works it is concluded that piGE shows better performance as MP for GE.

GE has been used to solve diverse types of problems: The Bin Packing Problem (BPP) [7], Even-5-parity [13, 15], Santa Fe Ant trail (SFAT) [13, 15], the design of Partially-Connected Artificial Neural Networks (DANN) [10], and the Symbolic Regression Problem (SRP) [8, 13, 15].

In this paper the DF, BF, and piGE are applied to ten well-known instances of the traditional SRP. The results are compared using a statistical test.

The paper is structured as follows: Sect. 2 presents a brief introduction to GE, BNF-Grammars and MPs, in Sect. 3 the Symbolic Regression Problem is described; the corresponding setup used in the experimental approach is explained in Sect. 4; and, finally the results and conclusions are exposed in Sects. 5 and 6 respectively.

2 Grammatical Evolution

Grammatical Evolution (GE) [5] is a variant of Genetic Programming (GP) [1], proposed by Ryan and O'Neill in 1998; GE differs from the traditional GP in 3 fundamental ways: it employs linear genotypes, it performs a genotype to phenotype mapping, and makes use of a grammar to create the solutions.

A Backus Naur Form Grammar (BNF-Grammar) is used to generate syntactically correct solutions (phenotypes) [5, 17], attending to certain restrictions and considering the context of the problem [18] taking the form of production rules. In order to select the corresponding production rule, a Genotype (binary or integer string) and a MP are employed [5, 7, 17, 19].

Derived from the MP a derivative tree is created, and a phenotype is obtained.

The generic algorithm of GE is described in Algorithm 1.

Algorithm 1 Grammatical Evolution Algorithm

```

1: Population = new_population(pop_size)
2: Generations = num_gen
3: Solution_found = false
4: Initialize (Population)
5: while Generations  $\geq$  0 do
6:   PerformMappingProcess(Population)
7:   Evaluate(Population)
8:   if Solution_found then
9:     Return Best Solution
10:  end
11: end if
12: if Solution_found == false then
13:   PerformGeneticOperators(Population)
14:   Generations - -
15: end if
16: end while
17: Return the best solution found

```

Due to the use of an MP, GE allows a distinction between the solution and search spaces, which means that depending on the use of one or another MP, a single genotype can be derived into distinct phenotypes. This characteristic allows GE to avoid the problem of getting stuck in local optima that the traditional GP has [15].

It is necessary to define the following components for the GE [7, 8]:

- Problem Instance.
- BNF-Grammar.
- Search Engine.

Figure 1 shows the methodology used in GE. The mapping process has been added as a fourth component to the original proposal taken from [7].

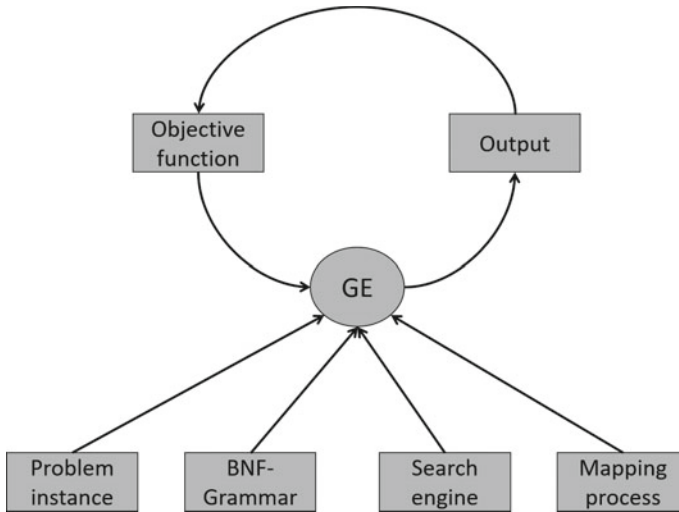


Fig. 1 Study approach GE’s methodology based on [7]

2.1 BNF-Grammar

A Context-Free Grammar (CFG) [11] don’t depend on the surroundings and is composed of a tuple defined as

$$G = \{N, T, R, S\}$$

where:

- N is the set of non-terminal symbols.
- R corresponds to the production rules.
- T is the set of terminal symbols.
- S corresponds to the start symbol, and $S \in N$.

Backus Naur Form (BNF) [20] is a notation used to express a grammar in a language in the form of production rules.

BNF consists of two main components: the terminals which are objects that can appear in the final expression, and the non-terminals which can be expanded into one or more terminals or non-terminals. A grammar is defined by a set of rules that determine a complex structure from small blocks.

The Grammar 1 represents an example of a BNF-Grammar, the non-terminals symbols are surrounded by “<>”, contrary to the terminals, which don’t have any surrounding symbols, and each non-terminal has a set of production rules or options that are separated by the symbol “|”. The production rules are separated by the symbol “[”.

$$\begin{aligned}
 \langle \text{start} \rangle &\models \langle e \rangle \\
 \langle e \rangle &\models \langle e \rangle \langle o \rangle \langle e \rangle \mid \langle v \rangle \\
 \langle v \rangle &\models X \mid Y \\
 \langle o \rangle &\models + \mid -
 \end{aligned}$$

Grammar 1 Example of a BNF-Grammar

As the grammar determine the structure of the solutions, changing the complete behavior of the solution is as simple as changing the grammar [13].

2.2 Search Engine

The principal objective of the Search Engine (SE) is to evolve the genotypes using as an objective value the evaluation of the instances in their phenotypic representation. Originally GE uses the the Genetic Algorithm (GA) [1] as SE. GA takes inspiration from the natural selection proposed by Darwin in 1859 and uses genetic operators such as selection, crossover, and mutation to evolve the genotype. The GA is shown in Algorithm 2.

Algorithm 2 Genetic Algorithm

```

1: Population = new_population(pop_size)
2: Generations = num_gen
3: Solution_found = false
4: Initialize (Population)
5: while termination condition not satisfied  $\geq 0$  do
6:   Generations += 1
7:   PerformGeneticOperators(Population)
8:   Evaluate(Population)
9:   if Solution_found then
10:    Return Best Solution
11:   end
12: end if
13: end while
14: Return the best solution found

```

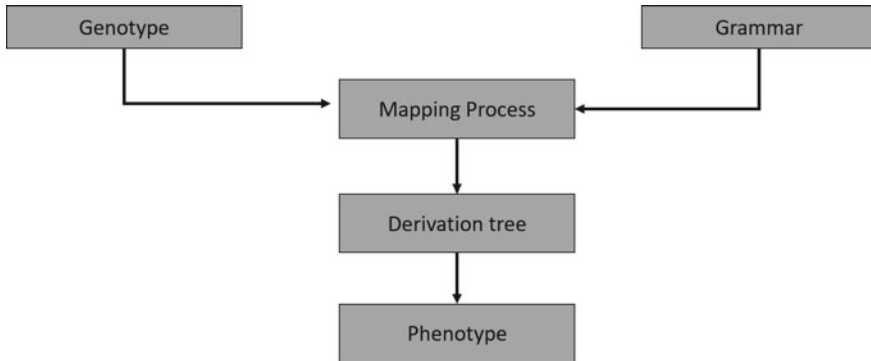


Fig. 2 Generic mapping process [15]

2.3 Mapping Process

The Mapping Process (MP) in GE is responsible for transforming a genotype to a phenotype, using a binary string as genotype and the productions derivation tree from the BNF grammar (Fig. 2 illustrates this process); the corresponding phenotype is obtained from this derivative tree. To do this, all the non-terminals (NT) are expanded using Eq. 1.

$$Prod\ rule = Codon\ value \% Number\ of\ production\ rules\ for\ the\ NT. \tag{1}$$

To exemplify the studied MPs, we will take the sample Grammar 1, and the following genotype to show the obtained derivative tree by each one of them.

Genotype = 2, 12, 7, 9, 3, 15, 23, 1, 11, 4, 6, 13, 2, 7, 8, 3, 35, 19, 2, 6.

Depth-First Mapping Process Depth-First (DF) [14] is considered the standard for GE, it begins from the start symbol and makes the expansion taking the left-most NT symbol available in the derivative tree. Figure 3 shows an example of the DF MP.

Equation 1 is performed using each codon value of the genotype. The number out of the parenthesis indicates the expansion order of the tree. In the example we take the start symbol <e>, the first codon value, which is 2, and the number of production rules that corresponds to the NT <e> which is 2; applying Eq. 1 we substitute $2\%2 = 0$, and this means that the corresponding production rule is the one in the position 0. Now we have three available NTs: <e>, <o> and <e>. We take the left-most NT, which corresponds to <e>.

We expand <e> into <e> <o> <e>, and so on, until no more NTs remain in the derivative tree. The corresponding algorithm for DF is shown in Algorithm 3.

Breadth-First Mapping Process The Breadth-First (BF) [15] uses an expansion process slightly different from DF. It uses the same equation to chose the

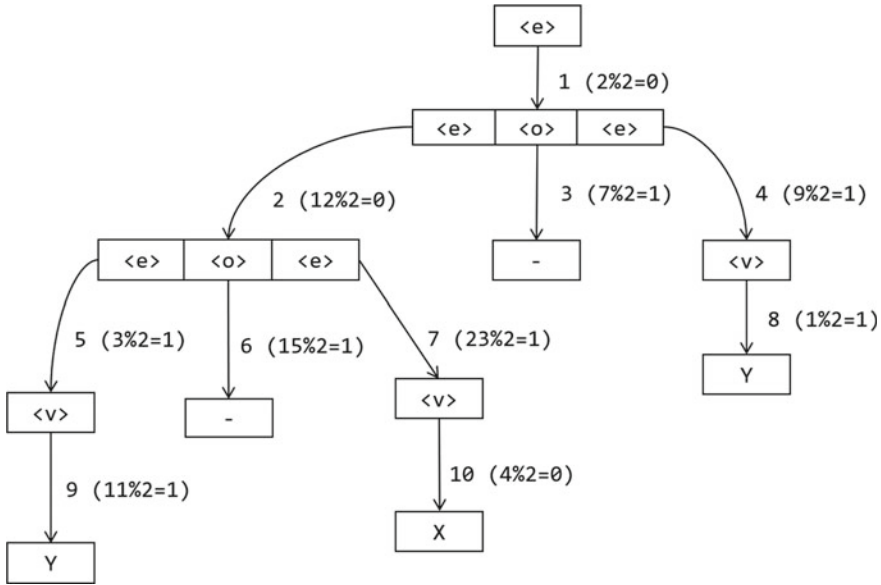


Fig. 4 Example of the Breadth-First Mapping Process [15]

corresponding production rule to expand, but the expansion is done by expanding all the NTs at the same level (taking the NTs from left to right) before moving to the next level in the expansion tree.

The process begins with the start symbol $\langle e \rangle$, applying the module rule, the corresponding production rule to expand is $\langle e \rangle \langle o \rangle \langle e \rangle$, we continue the expansion by taking these last three obtained NTs, and so on level by level, until no more NTs remain.

An example of this process can be seen in Fig. 4, where the expansion order is indicated by the numbers out the parenthesis. The corresponding algorithm for BF is shown in Algorithm 4.

π Grammatical Evolution Mapping Process The π Grammatical Evolution (piGE) [16] differs from the previous ones because it uses a pair of codons, the first one (the order codon) dictates the order of the expansion (it says which one of the available NTs in the derivative tree will be expanded); and the second one, called the content codon, works in the traditional way as in the other MPs.

piGE removes the linear dependency by evolving the expansion order in the evolutionary search [21] by making use of the order codon.

Equation 2 is used to choose the expansion order.

$$NT \text{ to expand} = \text{Order Codon value} \% \text{Number of available NTs}. \quad (2)$$

Figure 6 represents the order choice list of the expansions for the NTs, and Fig. 5 shows the corresponding derivative tree.

Algorithm 4 Breadth-First Mapping Process Algorithm [15]

```

1: listNT {List to store NTs seen}
2: Add start symbol from grammar to listNT
3: wraps = 0
4: while listNT is not empty do
5:   if reached end of chromosome then
6:     wraps++
7:     if wraps > max wraps allowed then
8:       return false
9:     end if
10:    reset chromosome iterator
11:   end if
12:   CurrentNT = get head of listNT
13:   CurrentCodon = get next codon value
14:   newProduction = currentCodon % number of productions for currentNT
15:   set currentNTs children = newProduction
16:   {This is the key to breadth first mapping}
17:   add newProduction to tail of listNT {Only adds NTs}
18: end while
19: Generate phenotype by traversing the leaf nodes of the derivation tree
20: return true

```

The example shown in Fig. 5 begins with the start symbol, as in the first step the only available NT is $\langle e \rangle$, Eq. 2 chose this NT that corresponds to the option in position 0. Using the second codon we select the corresponding production rule, and now we have three different options of NTs to expand. Taking the next pair of codons, we select and expand the available NT $\langle o \rangle$. We repeat this process until there is no more available NTs to expand in the order choice list.

The corresponding algorithm for piGE is shown in Algorithm 5.

3 Symbolic Regression Problem

The Symbolic Regression Problem (SRP) [1, 8, 15] is the process of obtaining a representative expression for a given set of finite points, it is used to know what was the expression that generated this data.

SRP represents an important task studied in the GP community [22].

The main objective of the SRP is finding the best combination of variables, coefficients, and symbols.

For example, the expression $\cos 2x$ can be represented with another equation, such as $1 - 2 \sin^2 x$. Both expressions give as a result the same values in a specific range of points.

Algorithm 5 π Grammatical Evolution Mapping Process Algorithm [16]

```

1: listNT {List to store NTs seen}
2: Add start symbol from grammar to listNT
3: wraps = 0
4: while listNT is not empty do
5:   if reached end of chromosome then
6:     wraps++
7:     if wraps > max wraps allowed then
8:       return false
9:     end if
10:    reset chromosome iterator
11:  end if
12:  {This is where the piGE order comes in}
13:  currentOrderCodon = get next codon value
14:  nextProductionIndex = currentOrderCodon % size of listNT
15:  currentNT = get listNT[nextProductionIndex]
16:  currentContentCodon = get next codon value
17:  newProduction = currentCodon % number of productions for currentNT
18:  set currentNTs children = newProduction
19:  {The new NTs are added where the parent NT was removed from}
20:  insert newProduction at listNT[nextProductionIndex]{Only adds NTs}
21: end while
22: Generate phenotype by traversing the leaf nodes of the derivation tree
23: return true

```

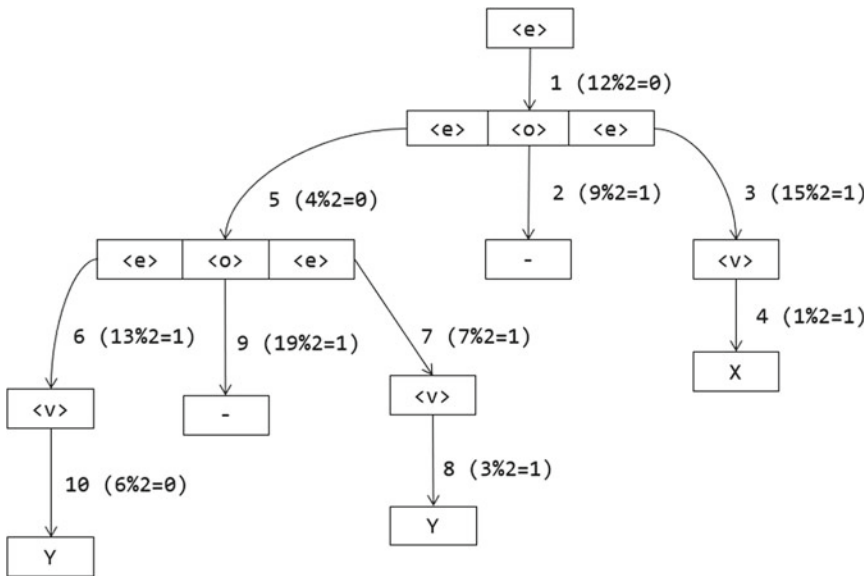


Fig. 5 Example of the π Grammatical Evolution Mapping Process [16]

Fig. 6 Order choice list for piGE

1. [(e)] → 2%1=0
2. [e, (o), e] → 7%3=1
3. [e, (e)] → 3%2=1
4. [e, (v)] → 23%2=1
5. [(e)] → 11%1=0
6. [(e), o, e] → 6%3=0
7. [v, o, (e)] → 2%3=2
8. [v, o, (v)] → 8%3=2
9. [v, (o)] → 35%2=1
10. [(v)] → 2%1=0

4 Experimental Setup

GA was used as SE, Table 1 shows the GA parameters used.

The SRP functions used as instances set are shown in Table 2, and it were taken from [8].

The Mean Root Squared Error (MRSE), Eq. 3, was used as objective function to evaluate the performance on the candidate solutions.

The grammar used for the SRP is shown in Grammar 2.

It was performed 33 independently runs for each instance with each MP. It was taken the median result to make the statistical comparison.

A statistical test was applied to discern about the correct test to be applied.

$$MRSE = \sqrt{\frac{\sum_{i=1}^N (x_i - y_i)^2}{N}} \tag{3}$$

where:

- N is the number of data points.
- x corresponds to the obtained value.
- y is the real value.

```

(start) ::= (expr)
(expr) ::= ((expr)(op)(expr)) | (pre-op)((expr)) | (var)
(var) ::= X | Y | 1.0
(pre-op) ::= sin | cos | exp | log
(op) ::= + | - | / | *
    
```

Grammar 2 Grammar used for the Symbolic Regression Problem [8]

Table 1 Parameter setup used

Parameter	Value
Population size	100
Dimensions	100
Function calls	250,000
Selection method	Random
Crossover	2 points
Mutation rate	1.0
Elitism rate	0.30

Table 2 Symbolic regression functions [8] used as instances set

Function	Fit cases
$F_1 = X^3 + X^2 + X$	20 random points $\in [-1, 1]$
$F_2 = X^4 + X^3 + X^2 + X$	
$F_3 = X^5 + X^4 + X^3 + X^2 + X$	
$F_4 = X^6 + X^5 + X^4 + X^3 + X^2 + X$	
$F_5 = \sin(X^2) \cos(X) - 1$	
$F_6 = \sin(X) + \sin(X + X^2)$	
$F_7 = \log(X + 1) + \log(X^2 + 1)$	20 random points $\in [0, 2]$
$F_8 = \sqrt{X}$	20 random points $\in [0, 4]$
$F_9 = \sin(X) + \sin(Y^2)$	200 random points $\in [-1, 1]$, $X \in [-1, 1]$, $Y \in [-1, 1]$
$F_{10} = 2 \sin(X) \cos(Y)$	

5 Results

Table 3 shown the median obtained by each instance with an MP.

A Shapiro-Wilk test [23] was performed with the results obtained to know if the data belong to a Normal distribution [24]. The Shapiro-Wilk test gives a p -value of 1.201e-05 that indicates that the data don't come from a Normal distribution.

The Shapiro-Wilk test shows that the data doesn't belong to a Normal distribution. Considering this, a Friedman non-parametric test [25] was applied to know if there is a significant difference between the results obtained by the MPs.

The p -value obtained by the Friedman non-parametric test is 0.067.

Table 3 Obtained results for each MP at each function

	Depth-First	Breadth-First	piGE
F_1	0.0476	0.0653	0.0532
F_2	0.0443	0.1170	0.0572
F_3	0.0986	0.1062	0.0709
F_4	0.0851	0.1948	0.0747
F_5	0.0374	0.0379	0.0291
F_6	0.0262	0.0355	0.0402
F_7	0.0092	0.0096	0.0120
F_8	0.0354	0.0591	0.0249
F_9	0.0291	0.0301	0.0147
F_{10}	0.0256	0.0221	0.0221

6 Conclusions and Future Work

In the present paper, three Mapping Processes were applied to ten instances of the Symbolic Regression Problem. The obtained results were compared with a Friedman non-parametric test with the purpose of know if there is a statistical difference between the studied Mapping Processes.

Derived from the statistical test, we could conclude that there is no statistical evidence to discern about the performance of the three studied Mapping Processes for Grammatical Evolution applied to the Symbolic Regression Problem. This implies that is possible to use any Mapping Process.

As future work, it is proposed to determine the algorithmic complexity for each MP to know if one of them can be easily applied. There exist other MP, like the Univariate Model-Based Grammatical Evolution (UMBGE), that can be applied to the Symbolic Regression Problem and can compare it with the results obtained.

Acknowledgements The authors want to thank National Council for Science and Technology of Mexico (CONACyT) through the scholarship for postgraduate studies: 703582 (B. Zuñiga) and the Research Grant CÁTEDRAS-2598 (A. Rojas), the León Institute of Technology (ITL), and the Guanajuato University for the support provided for this research.

References

1. Koza, J.R.: Genetic Programming On the Programming of Computers by Means of Natural Selection. Massachusetts Institute of Technology, Cambridge (1998)
2. Rich, C., Waters, R.C.: Automatic programming: myths and prospects. *Computer* **21**, 40–51 (1988)

3. Igwe, K., Pillay, N.: A study of genetic programming and grammatical evolution for automatic object-oriented programming: a focus on the list data structure. In: Pillay, N., Engelbrecht, A.P., Abraham, A., du Plessis, M.C., Snášel, V., Muda, A.K. (eds.) *Advances in Nature and Biologically Inspired Computing*, pp. 151–163. Springer, Cham (2016)
4. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. In: *An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. MIT Press (1992)
5. Ryan, C., Collins, J., Neill, M.O.: Grammatical evolution: evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) *Genetic Programming*, vol. 1391, pp. 83–96. Springer, Berlin, Heidelberg (1998)
6. Brabazon, A., O’Neill, M., McGarraghy, S.: *Natural Computing Algorithms*. Natural Computing Series, 1st edn. Springer, Berlin, Heidelberg (2015)
7. Sotelo-Figueroa, M.A., Soberanes, H.J.P., Carpio, J.M., Huacuja, H.J.F., Reyes, L.C., Soria-Alcaraz, J.A.: Improving the bin packing heuristic through grammatical evolution based on swarm intelligence. *Math. Probl. Eng.* (2014)
8. Sotelo-Figueroa, M.A., Hernández-Aguirre, A., Espinal, A., Soria-Alcaraz, J.A., Ortiz-López, J.: Symbolic regression by means of grammatical evolution with estimation distribution algorithms as search engine. In: *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, vol. 749, pp. 169–177. Springer (2018)
9. O’Neill, M., Brabazon, A.: Grammatical differential evolution. In: *IC-AI*, pp. 231–236 (2006)
10. Quiroz-Ramírez, O., Espinal, A., Ornelas-Rodríguez, M., Rojas-Domínguez, A., Sánchez, D., Puga-Soberanes, H., Carpio, M., Espinoza, L.E.M., Ortíz-López, J.: Partially-connected artificial neural networks developed by grammatical evolution for pattern recognition problems. In: *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Application*, pp. 99–112. Springer, Cham (2018)
11. Hemberg, E.A.P.: An exploration of grammars in grammatical evolution. Ph.D. thesis, University College Dublin (2010)
12. Nicolau, M., Agapitos, A.: Understanding grammatical evolution: Grammar design. In: *Handbook of Grammatical Evolution*, pp. 23–53. Springer (2018)
13. Fagan, D., O’Neill, M., Galván-López, E., Brabazon, A., McGarraghy, S.: An analysis of genotype-phenotype maps in grammatical evolution. In: *European Conference on Genetic Programming*, pp. 62–73. Springer, Berlin, Heidelberg (2010)
14. O’Neill, M., Ryan, C.: Grammatical evolution. *IEEE Trans. Evo. Comput.* **5**, 349–358 (2001)
15. Fagan, D., O’Neill, M.: *Analysing the Genotype-Phenotype Map in Grammatical Evolution*. Ph.D. thesis, University College Dublin (2013)
16. O’Neill, M., Brabazon, A., Nicolau, M., Garraghy, S.M., Keenan, P.: π grammatical evolution. In: Deb, K. (ed.) *Genetic and Evolutionary Computation—GECCO 2004*, pp. 617–629. Springer, Berlin, Heidelberg (2004)
17. Dempsey, I., O’Neill, M., Brabazon, A.: *Foundations in Grammatical Evolution for Dynamic Environments*. Vol. 194 of *Studies in Computational Intelligence*, 1st edn. Springer, Berlin, Heidelberg (2009)
18. Hugosson, J., Hemberg, E., Brabazon, A., O’Neill, M.: Genotype representations in grammatical evolution. *Appl. Soft Comput.* **10**(1), 36–43 (2010)
19. McKay, R.I., Hoai, N.X., Whigham, P.A., Shan, Y., O’Neill, M.: Grammar-based genetic programming: a survey. *Gen. Program. Evol. Mach.* **11**, 365–396 (2010)
20. Backus, J.W., Bauer, F.L., Green, J., Katz, C., McCarthy, J., Perlis, A.J., Rutishauser, H., Samelson, K., Vauquois, B., Wegstein, J.H., van Wijngaarden, A., Woodger, M.: Revised report on the algorithm language algol 60. *Commun. ACM* **6**, 1–17 (1963)
21. Fagan, D., Murphy, E.: Mapping in grammatical evolution. In: *Handbook of Grammatical Evolution*, pp. 79–108. Springer (2018)
22. White, D.R., McDermott, J., Castelli, M., Manzoni, L., Goldman, B.W., Kronberger, G., Jaskowski, W., O’Reilly, U.-M., Luke, S.: Better GP benchmarks: community survey results and proposals. *Gen. Program. Evol. Mach.* **14**, 3–29 (2012)
23. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). *Biometrika* **52**, 591 (1965)

24. Soria-Alcaraz, J.A., Sotelo-Figueroa, M.A., Espinal, A.: Statistical comparative between selection rules for adaptive operator selection in vehicle routing and multi-knapsack problems. In: *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, pp. 389–400. Springer (2018)
25. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **1**, 3–18 (2011)

Optimization and Evolutionary Algorithms

A Survey of Hyper-heuristics for Dynamic Optimization Problems



Teodoro Macias-Escobar, Bernabé Dorronsoro, Laura Cruz-Reyes, Nelson Rangel-Valdez and Claudia Gómez-Santillán

Abstract Dynamic optimization problems have attracted the attention of researchers due to their wide variety of challenges and their suitability for real-world problems. The application of hyper-heuristics to solve optimization problems is another area that has gained interest recently. These algorithms can apply a search space exploration method at different stages of the execution for finding high quality solutions. However, most of the proposed works using these methodologies do not focus on the development of hyper-heuristics for dynamic optimization problems. Despite that, they arise as very appropriate methods for dynamic problems, being highly responsive and able to quickly adapt to any possible changes in the problem environment. In this paper, we present a brief study of the most salient previously proposed hyper-heuristics to solve dynamic optimization problems, and classify them, taking into consideration the complexity of their low-level heuristics. Then, we identify some of the most important research areas that have been vaguely explored in the Literature yet.

Keywords Dynamic optimization · Hyper-heuristics · Dynamic optimization problems · Low-level heuristics

T. Macias-Escobar (✉)

Tecnológico Nacional de México, Instituto Tecnológico de Tijuana, Tijuana, Mexico
e-mail: teodoro_macias@hotmail.com

T. Macias-Escobar · B. Dorronsoro

Universidad de Cádiz, Cadiz, Spain

e-mail: bernabe.dorronsoro@uca.es

L. Cruz-Reyes · N. Rangel-Valdez · C. Gómez-Santillán

Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Ciudad Madero, Mexico

e-mail: cruzreyeslaura@gmail.com

N. Rangel-Valdez

e-mail: rangel.nelson1980@gmail.com

C. Gómez-Santillán

e-mail: cggs71@hotmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_33

1 Introduction

Optimization problems may present changes in the problem environment as time progresses. These alterations can be related to objective functions, constraints, decision variables or any other elements. These problems are defined as dynamic optimization problems (DOPs). Several DOPs benchmarks have been previously proposed, adding a new challenge for optimization algorithms.

The application of heuristics has been gaining interest among researchers lately. Especially due to their capability of quickly producing good-quality solutions, an important factor when solving DOPs. There is also interest in the use of hyper-heuristics. These methods focus on the dynamic selection or generation of good heuristics or metaheuristics to solve the problem. This allows hyper-heuristics to use the most suitable method for the current state of both the search and the problem.

One of the most common applications of dynamic hyper-heuristics is solving scheduling and programming problems, such as the dynamic job shop scheduling problem (DJSSP). Several hyper-heuristics have been proposed to solve DJSSP instances. Reviews related to these problems have been presented in [8, 33].

While static hyper-heuristics have been previously analyzed and classified in different reviews [10–12]. There is no general review of dynamic hyper-heuristics, further than those focused on solving DJSSP. This paper presents a brief study of the most relevant existing hyper-heuristics to solve different DOPs.

The remaining of this paper is organized as follows. Section 2 presents a background regarding the main topics reviewed in this work. Section 3 provides a brief explanation of some of the most relevant previously proposed hyper-heuristics to solve DOPs that use problem-specific heuristics as low-level heuristics (LLH). Section 4 presents a similar review but focusing on hyper-heuristics that use metaheuristics as LLHs. Section 5 brings a discussion about dynamic hyper-heuristics and the possible future research. Section 6 presents the conclusion of this survey.

2 Background

In this section, we present a brief explanation and definitions of dynamic optimization problem and hyper-heuristic, which are the main topics related to this work.

2.1 *Dynamic Optimization Problem*

A dynamic optimization problem (DOP) can be defined as finding a set of decision variables that satisfies a set of restrictions while trying to obtain the best possible result for an objective function that changes over time. Formally, minimization DOPs can be defined as:

$$\min f(\vec{x}, t) \text{ s.t. } g(\vec{x}, t) > 0, h(\vec{x}, t) = 0 \tag{1}$$

where x is a vector of decision variables, f is the objective function to be optimized, t represents the current time step. Lastly g and h are the set of inequality and equality constraints, respectively. Two new factors must be considered for DOPs with respect to static problems: how often changes occur in an environment (change frequency) and the strength of those changes (change severity).

Multi-objective DOPs (DMOPs) have a vector of objective functions to be solved. Considering minimization problems, DMOPs can be formally defined as:

$$\min F(\vec{x}, t) = \{f_1(\vec{x}, t), f_2(\vec{x}, t), \dots, f_m(\vec{x}, t)\}, \quad f_i: R^n \rightarrow R \tag{2}$$

where F is a vector of objective functions to be optimized. A common issue in DMOPs is that the objectives are in conflict, which makes setting a single solution as optimal difficult. A group of non-dominated individuals, defined as the Dynamic Pareto Optimal Set (POS(t)*), is set as solutions, and can be formally defined as:

$$\text{POS}(t) = \{\vec{x} | \nexists f(\vec{y}, t) \preceq f(\vec{x}, t), \vec{y} \in V\} \tag{3}$$

where V is the feasible solution search space. These solutions define the Dynamic Pareto Optimal Front (POF(t)*), which is the Pareto Optimal solution set with respect to the objective space for time step t :

$$\text{POF}(t) = \{f(\vec{x}, t) | \nexists f(\vec{y}, t) \preceq f(\vec{x}, t), \vec{y} \in V\} \tag{4}$$

2.2 Adaptation of Heuristic to Problem Changes

Environmental changes could affect the quality of the solutions found by optimization algorithms, lowering their quality or even making them unfeasible. A change in the optimal solution might provoke that heuristics require more time to reach the area where the new optimal solution is located. To deal with these issues, several works in the Literature have added two methods in the heuristics proposed to tackle these problems:

Change detection method. Perceives a change in the problem environment. Several change detection approaches have been previously proposed, such as setting a predefined number of iterations before a change, re-evaluation of a solutions [16] or the use of a diversity threshold [48].

Change response method: Adaptation to a change to avoid loss in quality or unfeasible solutions. According to [2], these methods can be classified by different approaches: diversity maintenance, memory, prediction, parallelization or dividing a DOP into several static optimization problems.

A classification and deeper studies for dynamic single- and multi-objective optimization can be found in [2, 32]. However, dynamic heuristics and metaheuristics are not the only stochastic methods to solve DOPs. Other approaches based on hyper-heuristics, whose nature allows them performing the change detection and response methods implicitly, have also been previously considered.

2.3 *Hyper-heuristic*

Taking into consideration the “No Free Lunch” theorem [50] there is not any algorithm better than any other one for the whole class of optimization problems. However, it could be possible that the combined application of a set of algorithms may lead to better solutions for a wider set of problems.

A hyper-heuristic can be defined as a high-level methodology that can generate or select heuristics (or metaheuristics), named low-level heuristics (LLHs), to solve hard computational problems [10]. While heuristics and metaheuristics act directly on the solution search space, a hyper-heuristic only acts in the heuristic search space and does not tackle the problem directly. Focusing on finding the most suitable LLH for the current state of the problem.

Hyper-heuristics attempt to use the strengths of their LLHs to cover their respective flaws. In theory, this means that they could obtain better solutions than those these heuristics could individually achieve. The application of a set of LLHs also means that a wider variety of problems could be solved.

2.4 *Classification of Hyper-heuristics*

Hyper-heuristics can be classified in terms of two main characteristics, according to [10]: the learning process (feedback) and the nature of the heuristics search space. The former focuses on how the information and knowledge regarding the performance of the LLHs and the problem are handled. Hyper-heuristics can be classified as *non-learning*, *online learning* (gathering information during execution) or *offline learning* (knowledge obtained from a set of training instances).

The second dimension focuses on how the LLHs are defined: selection, where LLHs are chosen from a set of heuristics; and generation, where LLHs are created using available components. LLHs are classified in two types: heuristics that generate new solutions and heuristics that perturbate solutions. Table 1 shows how the dynamic hyper-heuristics reviewed in this work are classified for both dimensions.

According to [9], selection solution-perturbative hyper-heuristic can be divided in two components: heuristic selection method and move acceptance. The first element defines which LLH will be applied on the problem after analyzing its current state while the second element defines the criteria to accept the new solution.

Table 1 Classification of dynamic hyper-heuristics based on their nature of heuristic search space, feedback and LLH complexity

Heuristic search space	Generation	Several hyper-heuristics for DJSSP mentioned in [8]
	Selection	[5, 13, 19, 20, 25–29, 34, 37, 38, 41–48]
Feedback	Online	[5, 13, 19, 20, 25–28, 37, 38, 41–44, 46–48]
	Offline	[42–44]
	No learning	[25–27, 29, 34, 41, 43, 45–47]
LLH complexity	Problem-specific	[13, 19, 20, 25–29, 34, 37, 38, 41–44]
	Metaheuristics	[5, 45–48]

The complexity of the LLHs is not limited, as they could be problem-specific heuristics or metaheuristics. Hyper-heuristics and LLHs can even be equally complex. However, they are still differentiated by the search spaces they focus on.

3 Dynamic Hyper-heuristics with Problem-Specific Low-Level Heuristics

An initial investigation of hyper-heuristics to solve dynamic optimization problems is presented in [34]. In this study, a greedy hyper-heuristic is used to solve a set of single-objective DOPs with different combinations of change frequencies and severity levels. Six different LLHs are used, bit-flip mutation with five different mutation rates and the Davis' bit hill-climbing method [15]. The experimental results confirmed that different problem environment changes regarding frequency or severity require different change adaptation approaches and that hyper-heuristics are capable to adapt to these changes as they can choose which heuristic is appropriate to use according to the current problem conditions.

In [26, 27], the authors propose a set of selection solution-perturbative hyper-heuristics to solve the Moving Peaks Benchmark (MPB) under different change severities and frequencies, with peak heights and widths constantly changing. A total of 35 hyper-heuristics are tested, each methodology using a specific combination of heuristic selection method and acceptance criterion. The heuristic selection methods used are Simple Random, Greedy, Choice Function, Random Permutation Descent [14] and Reinforcement Learning [22]. Meanwhile, the acceptance criteria used are All Moves, Only Improving, Improving and Equal, Exponential Monte Carlo with Counter [1], Simulated Annealing [3], Simulated Annealing with Reheating [4] and Great Deluge [23]. These heuristic selection methods are compared against a hyper-mutation based single point search selection method. The results show that Choice Function-Improving and Equal hyper-heuristic had a better ranking among all other algorithms performance-wise.

A posterior study [28] from the same authors propose a hyper-heuristic using the ant colony optimization algorithm [17] as a heuristic selection method. This method, along with two different versions of the Choice Function and Reinforcement Learning are combined with the Improving and Equal acceptance criterion and their performance is compared by solving MPB instances. The ant-based selection method had a better performance among all methods, especially on problems with high change severity and frequency.

A selection hyper-heuristic using a memory-based approach is hybridized with an evolutionary algorithm in [25, 41]. In this approach, the best individuals from the memory population, obtained by using an evolutionary algorithm (EA), and the search population, obtained by the hyper-heuristic, are inserted into an external memory. Whenever a change is detected, the external memory and the memory population are merged with the best individuals being inserted into the memory population and reevaluated along with the search population. This hybrid approach is tested on MPB and dynamic Generalized Assignment Problem (DGAP) [31] instances using problem-specific LLHs. Five heuristic selection methods (Simple Random, Random Descent, Random Permutation, Random Permutation Descent and Choice Function) are combined with three different move acceptance criteria (All Moves, Only Improving and Monte Carlo) for a total of 15 hyper-heuristics tested in this work and compared to a memory/search algorithm [7]. The hybrid approach significantly outperforms the memory/search algorithm performance-wise as the Choice Function and Random Permutation Descent selection methods, both using Only Improving acceptance criterion, were the most effective algorithms when combined with an EA.

HH-PBIL2 [42, 43] and HH-EDA2 [44] are hybrid algorithms that combine a bi-population version of the population based incremental learning algorithm (SPBIL2) [52] and a selection hyper-heuristic. The population is divided into two subsets, one assigned to SPBIL2 (P_1) and the other to the hyper-heuristic (P_2). Each subpopulation is given a probability vector, P_1 is initialized with the central probability vector. An offline learning process is done prior the execution to determine a set of probability vectors for P_2 , each vector is given an initial equal score and one is chosen randomly. Each subpopulation is independently sampled using the respective probability vector. The best solutions from both samples are inserted into a vector (B). Then, the best fitness value from the whole population or P_2 is taken as information for the heuristic selection method. P_1 learns from B and mutates. A heuristic selection method chooses as P_2 a probability vector obtained in the offline learning process. Experiments were carried out using Simple Random, Random Descent, Random Permutation, Random Permutation Descent, Reinforcement Learning and Ant-based Selection as heuristic selection methods. All Moves was used as the move acceptance criterion.

Other experiments were also carried on. Different change adaptation methods, involving resetting the probability vectors were tested as well as different online and offline approaches of the framework. The XOR DOP generator [51] was used to create dynamic environments on a set of static problems. Random permutation based hyper-heuristics provided the best results while restarting P_1 to a central probability vector and keeping P_2 unmodified was the best resetting scheme.

Hyper-heuristics have also been proposed to solve DOPs with real-life like environments. The Open Car Optimization Problem (TORCS) [30] is an open source car racing simulator that provides car dynamics. A Simple Random-Improving and Equal hyper-heuristic to tackle TORCS problems was presented in [29]. This methodology used a vector of standard deviation values for Gaussian mutation as its set of LLHs and its performance was compared to a genetic algorithm (GA) and a particle swarm optimization approach. The results displayed the potential of selection hyper-heuristics to handle dynamic and noisy environments, such as TORCS.

The dynamic vehicle routing problem (DVRP) [24] incorporates dynamic elements to VRP by adding new problem-related information at any time. EH-DVRP [19] is an online learning selection hyper-heuristic that uses an evolutionary approach to generate and evolve combinations of constructive, repairing and noise LLHs, which are applied in sequence to construct and improve partial routes. EH-DVRP is tested using several large instances that simulate real-life changing environments. EH-DVRP exhibited capability of self-adaptation to handle dynamic environments by performing an adequate selection of a combination of LLHs and producing results better than heuristic-based methods and competitive results against state-of-the-art algorithms.

A hyper-heuristic using a gene expression programming framework, named GEP-HH, is presented in [37] to solve DVRP instances. GEP-HH evolves a population of individuals, decoding them to generate heuristic selection methods and acceptance criteria. This allows GEP-HH to adapt to a problem environment by generating high-level heuristics to handle a problem on its current state. It uses a memory mechanism to maintain solution diversity. The heuristic selection method ranks the LLHs, applying them from best to worst ranked until the current solution cannot be improved. A new obtained solution replaces the current one if there is an outperformance or if the move acceptance criterion accepts it. After all LLHs have been applied, they are reevaluated and ranked again. Every certain number of iterations, GEP-HH generates a new heuristic selection process and acceptance criteria.

A variant of GEP-HH is presented in [38]. The dynamic multi-armed bandit algorithm with extreme-value based reward [18] performs the LLH selection. The gene expression algorithm framework focuses only on move acceptance criteria generation. GEP-HH displayed the capability of online-learning generation hyper-heuristics to obtain high-quality results for DOPs by creating problem-specific LLHs and move acceptance criteria.

In [20] a selection GA-based hyper-heuristic is proposed to solve dynamic instances of the Multilevel Capacitated Lot Sizing Problems with Linked Lot Sizes (MLCLSP-L). The proposed approach uses a matrix notation that assigns to each machine a certain LLH to be applied for each period. The LLH permutations with the best quality are kept and used as parents to generate new permutations. All Moves is used as the acceptance criterion. However, the genetic algorithm would stop if no improvement is found after a certain number of generations. The quality of the obtained schedules had over a 96% similarity with respect to the optimal solution in less than three minutes.

A tabu search based selection hyper-heuristic, named binary exponential back off (BEBO) proposed in [35] is used in [13] on a dynamic environment by solving a gully pot maintenance route scheduling problem. It focuses on improving a route by minimizing the failure probability of all gully pots during the maintenance period. BEBO uses a set of problem-specific route related and schedule related LLHs and Only Improving is used as the acceptance criterion. The proposed strategy outperformed real-world gully pot maintenance approaches that were being utilized by the time this study was presented.

4 Metaheuristics as Low-Level Heuristics

Problem-specific heuristics can be useful alternatives over exact techniques when their application is excessively expensive both computationally and timewise. However, this can present an issue when trying to solve problems with different properties or characteristics. Metaheuristics are beyond that restriction. Since these methods are problem-independent, they could be applied to almost any kind of problem.

This does not mean that a metaheuristic can perform well for all problems, as we must take into consideration the “No Free Lunch” theorems [50], which establish that if an algorithm performs well for a subset of problems it will pay off by having poor performance for another subset. However, metaheuristics have a high usability. As mentioned in Sect. 2, the complexity of low-level heuristics is not limited. Therefore, an LLH set could have a group of metaheuristics.

To our knowledge, the first attempt to use metaheuristics as LLHs in a hyper-heuristic is shown in [48]. The Adaptive Hill Climbing (AHC) is a local search (LS) strategy used as a component of a dynamic genetic-algorithm based memetic algorithm named AHMA. AHC uses two hill climbing methods: a greedy crossover-based hill climbing (GCHC) and a steepest mutation-based hill climbing (SMHC). GCHC takes an elite individual (the solution with the best fitness) and a roulette-wheel selected individual from a population as parents to generate an offspring using uniform crossover, replacing the elite individual if it has better fitness. SMHC mutates several random bits from an elite individual. The mutated solution replaces the elite individual if it has better fitness. AHC assigns an equal probability to each HC. Then, it randomly chooses and executes one step LS operation using the selected method. The improvement (or deterioration) degree of the selected individual is updated. This process is repeated until the all LS steps for the current generation finish. Then, the probability of each HC method is recalculated taking in consideration its respective total improvement during the current generation.

In AHMA, AHC is applied before the change detection and adaptation methods were applied. AHC has also been used on a binary version of the weighted superposition attraction algorithm (bWSA) proposed in [6]. In this case, AHC is applied after the change detection and adaptation methods are executed. The performance of AHMA and bWSA to solve problems with dynamic environments set by the XOR

DOP generator [51] proved to be more effective alternatives to solve several binary-encoded DOPs in comparison to several GA, particle swarm optimization (PSO) and firefly algorithm strategies.

The application of a hyper-heuristic as a main strategy is first shown in [45], to our knowledge. A set of metaheuristics specialized on solving dynamic optimization problems was used as LLHs for the Heterogeneous Meta-Hyper Heuristic (HMHH) [21] to solve MPB-generated instances. HMHH starts with a population of candidate solutions and a random LLH is assigned to each solution. Each LLH is applied for a certain number of sampling iterations, the new solutions are evaluated, and the heuristic selection method assigns a new LLH to each solution according to its criteria. This process is repeated until the stopping condition is met. In this study, the authors proposed the use of the Simple Random heuristic selection method and All Moves as the acceptance criterion. HMHH is tested on several instances with quasi-static, progressive, abrupt and chaotic environment and compared against the LLHs executed individually. The performance of HMHH was superior than the rest of the algorithms for all environments, except quasi-static.

The study of HMHH is extended in [46] by adding a Roulette wheel and a Tournament selection based heuristic selection methods. Two versions of each hyper-heuristic are proposed, one sharing global stated information between heuristics and other without information sharing. The results presented the relevance of global sharing information as the best results were obtained using this technique. It is also implied that there is a correlation between the selection pressure of a heuristic selection method and the change frequency and severity of a DOP with respect to the quality of the solutions. Heuristic selection methods with lower selection pressure, such as Simple Random has a better performance when working on more chaotic environments. In [47] HMHH is tested using as heuristic selection methods a fixed selection, two ant colony-based methods based on pheromone concentration and fitness improvement, and a frequency improvement selection, which sets the probability of each heuristic according on the how frequently it improves the fitness of a solution. HMHH outperforms all LLHs run individually on a single population and several hyper-heuristics with fixed and random speciation. However, particle swarm optimization heuristics outperformed all algorithms when a population is divided on subpopulations. This line of research shows that using metaheuristics as LLHs is not only feasible, but also offers the possibility to tackle a wider array of DOPs efficiently.

A study made in [5] presents a hyper-heuristic with three population-based metaheuristics being used as LLHs to solve dynamic multi-dimensional knapsack problems. A Roulette Wheel-based heuristic selection method is used. Each LLH is assigned a selection probability, promoting the selection of high-improving LLHs. All Moves is used as the move acceptance criterion as the new population is always updated. A diversity threshold functions as a change detection method while Triggered Random Immigrants [48] along with a change in the solution representation

size act as change adaptation methods. This hyper-heuristic shows an important difference over other previously proposed works, as AHC, another hyper-heuristic, is used as a local search strategy within the process of the proposed hyper-heuristic. This lets us understand the flexibility of hyper-heuristics regarding their application to solve DOPs.

5 Discussion and Future Research Areas

In this section, we present areas and topics that in our belief might have a higher impact on the field of dynamic optimization and require deeper investigation. We also discuss possible future research. Our discussion focuses on generation hyper-heuristics to solve problems different than DJSSP, dynamic multi-objective optimization, the complexity of low-level heuristics and the application of fitness landscape analysis methods for problem and algorithm information gathering.

5.1 Generation Hyper-heuristics on Non-job-Shop Problems

As this survey shows, the application of hyper-heuristics to solve dynamic optimization problems has been previously proposed. However, when solving problems different than the dynamic job shop scheduling problem (DJSSP), most of the works in the Literature focus on selection hyper-heuristics. This means that a set of pre-defined LLHs are used. While this is helpful to limit the available LLH set and avoid having an excessively large heuristic search space, the capability of designing problem-specific heuristics might be beneficial when solving a group of instances with similar characteristics. To our knowledge, GEP-HH is the only hyper-heuristic that uses a heuristic generation process to solve DOPs different than the DJSSP. However, it focuses on generating heuristic selection and move acceptance methods and not LLHs. Therefore, the application of generation hyper-heuristic to create or select LLHs to solve other DOPs is a field that has been briefly researched and further investigation is necessary.

5.2 Solving Dynamic Multi-objective Optimization Problems

Most of the dynamic hyper-heuristics in the Literature focus on solving single-objective DOPs. This presents a lack of proposals to solve DMOPs. Perhaps, the only exception are the studies presented to solve DJSSP instances, as most of them have multiple objective functions. However, this is not the case for most of the non-job shop related DMOPs. This situation provides a potential research. Most real-life problems change constantly and have multiple objectives. Taking in consideration the “Not Free

Lunch” theorems, it is understandable to believe that hyper-heuristics could obtain high-quality solutions for DMOPs. Previously proposed hyper-heuristics could be adapted to solve DMOPs using LLHs focused on multi-objective optimization or fitness aggregation methods.

5.3 Complexity of Low-Level Heuristics

As mentioned in Sect. 2.4, even complex approaches, such as metaheuristics, can be used as an LLH. This allows hyper-heuristics to work on a wider array of problems in comparison to LLH sets formed of problem-specific heuristics. Also, metaheuristics can explore the solution search space more thoroughly, as each LLH has its own search process. As there is practically no need to adapt metaheuristics to each problem, they can be worked as a black box, allowing researchers to focus on the hyper-heuristic itself. However, the computational effort and time cost that metaheuristics might take in comparison to problem-specific heuristics could be discouraging when solving heavy-resource-demanding problems.

DMOPs are usually difficult problems, as they present not only a changing environment but also several objective functions that are usually in conflict. Since DMOPs can be more complex than single-objective DOPs, it would be desirable to use a more sophisticated approach to solve them. This presents an opportunity to introduce the application of dynamic metaheuristics as LLHs.

5.4 Fitness Landscape Analysis

The quality of the data collected is crucial to perform an adequate LLH selection. Founding this decision only on fitness values can be deceiving, as this could provoke stalling in local optima. Environment changes in DOPs could make reaching other areas in the solution search space harder. Fitness Landscape Analysis (FLA) methods evaluate the difficulty of an optimization problem by identifying characteristics about its environment, such as its smoothness, ruggedness and neutrality.

The incorporation of FLA methods into a heuristic selection method is an attractive alternative. A study made in [36] presents several FLA methods to analyze DOPs. This offers an opportunity to incorporate these methods into a dynamic hyper-heuristic for future research.

Evolvability, which evaluates the capacity of an individual to evolve into fitter solutions, is inserted into heuristic selection methods of static hyper-heuristics in [39, 40]. Recently, [49] proposed the concept of Population Evolvability, which can handle a population and obtain information regarding both problem and algorithm by analyzing the probability of obtaining a fitter population and the improvement degree. The authors proposed its capability as an algorithm selection task method. Although it has only been tested on static optimization problems.

6 Conclusions

The application of hyper-heuristics has been proposed and studied widely for solving static optimization problems. However, studies about their feasibility and application to solve DOPs has only been started since last decade. This brings an opportunity for researchers to investigate novel areas of dynamic optimization.

This survey had the objective to provide the reader with a general understanding of how the current hyper-heuristics have been applied to tackle DOPs, their classification, components and the advantages that they offer over other optimization algorithms that focus directly on the solution search space.

Most hyper-heuristics proposed in the Literature focus either on single-objective DOPs or DJSSP while using problem-specific heuristics as LLHs. Therefore, using metaheuristics as LLHs and applying hyper-heuristics to solve DMOPs are two research areas that offer new, unsolved challenges. Also, the inclusion of problem-information gathering measures, such as FLA methods, in hyper-heuristics presents another possible advantage over other algorithms.

This work also sought to encourage researchers to get introduced in this area, expanding the current research scope regarding hyper-heuristics and their application on DOPs. As a final remark, we present two suggestions to take in consideration when working with hyper-heuristics. First, it is advised to be well-informed of the used LLHs. It is recommended, if possible, to test all candidate LLHs individually before defining the LLH set. This will allow the hyper-heuristic to work with good LLHs to produce even better solutions. Lastly, researchers should carefully define the heuristic selection method and the move acceptance criteria, as this could drastically affect the performance of the hyper-heuristic.

Acknowledgements This work was supported by the project TecNM 6308.17-P and the following CONACyT projects; Consolidation National Lab under project 280712; Cátedras CONACyT under Project 3058; and CONACyT National Grant System under Grant 465554; Spanish MINECO and ERDF under contracts TIN2014-60844-R (SAVANT project) and RYC-2013-13355, and the University of Cadiz (contract PR2018-056).

References

1. Ayob, M., Kendall, G.: A monte carlo hyper-heuristic to optimise component placement sequencing for multi head placement machine. In: Proceedings of the International Conference on Intelligent Technologies, InTech, vol. 3, pp. 132–141, Dec 2003
2. Azzouz, R., Bechikh, S., Said, L.B.: Dynamic multi-objective optimization using evolutionary algorithms: a survey. In: Recent Advances in Evolutionary Multi-objective Optimization, pp. 31–70. Springer, Cham (2017)
3. Bai, R., Kendall, G.: An investigation of automated planograms using a simulated annealing based hyper-heuristic. In: Metaheuristics: Progress as Real Problem Solvers, pp. 87–108. Springer, Boston (2005)

4. Bai, R., Blazewicz, J., Burke, E.K., Kendall, G., McCollum, B.: A simulated annealing hyper-heuristic methodology for flexible decision support. Technical Report, School of CSiT, University of Nottingham, UK (2007)
5. Baykasoğlu, A., Ozsoydan, F.B.: Evolutionary and population-based methods versus constructive search strategies in dynamic combinatorial optimization. *Inf. Sci.* **420**, 159–183 (2017)
6. Baykasoğlu, A., Ozsoydan, F.B.: Dynamic optimization in binary search spaces via weighted superposition attraction algorithm. *Expert Syst. Appl.* **96**, 157–174 (2018)
7. Branke, J.: Memory enhanced evolutionary algorithms for changing optimization problems. In: Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99. vol. 3, pp. 1875–1882. IEEE (1999)
8. Branke, J., Nguyen, S., Pickardt, C.W., Zhang, M.: Automated design of production scheduling heuristics: a review. *IEEE Trans. Evol. Comput.* **20**(1), 110–124 (2016)
9. Bilgin, B., Özcan, E., Korkmaz, E.E.: An experimental study on hyper-heuristics and exam timetabling. In: International Conference on the Practice and Theory of Automated Timetabling, pp. 394–412. Springer, Berlin (2006)
10. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Woodward, J.R.: A classification of hyper-heuristic approaches. In: Handbook of Metaheuristics, pp. 449–468. Springer, Boston (2010)
11. Burke, E.K., Gendreau, M., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Qu, R.: Hyper-heuristics: a survey of the state of the art. *J. Oper. Res. Soc.* **64**(12), 1695–1724 (2013)
12. Burke, E. K., Hyde, M. R., Kendall, G., Ochoa, G., Özcan, E., & Woodward, J. R. (2018). A classification of hyper-heuristic approaches: revisited. In: Handbook of Metaheuristics, vol. 272, p. 453
13. Chen, Y., Cowling, P., Polack, F., Remde, S., Mourdjis, P.: Dynamic optimisation of preventative and corrective maintenance schedules for a large scale urban drainage system. *Eur. J. Oper. Res.* **257**(2), 494–510 (2017)
14. Cowling, P., Kendall, G., Soubeiga, E.: A hyperheuristic approach to scheduling a sales summit. In: International Conference on the Practice and Theory of Automated Timetabling, pp. 176–190. Springer, Berlin (2000)
15. Davis, L.: Bit-climbing, representational bias, and test suite design. In: Proceedings of the 4th International Conference on Genetic Algorithm, pp. 18–23 (1991)
16. Deb, K., Rao U.B., Karthik, S.: Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. In: International Conference on Evolutionary Multi-criterion Optimization, pp. 803–817. Springer, Berlin (2007)
17. Dorigo, M., Stützle, T.: *Ant Colony Optimization*. MIT Press, Cambridge (2004)
18. Fialho, Á.: Adaptive operator selection for optimization. Doctoral dissertation, Université Paris Sud-Paris XI (2010)
19. Garrido, P., Riff, M.C.: DVRP: a hard dynamic combinatorial optimisation problem tackled by an evolutionary hyper-heuristic. *J. Heuristics* **16**(6), 795–834 (2010)
20. Gökçe, M.A., Beygo, B., Ekmekçi, T.: A hyperheuristic approach for dynamic multilevel capacitated lot sizing with linked lot sizes for APS implementations. *J. Yaşar Univ.* **12**(45), 1–13 (2017)
21. Grobler, J., Engelbrecht, A.P., Kendall, G., Yadavalli, V.S.S.: Alternative hyper-heuristic strategies for multi-method global optimization. In: 2010 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, IEEE, July 2010
22. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement learning: a survey. *J. Artif. Intell. Res.* **4**, 237–285 (1996)
23. Kendall, G., Mohamad, M.: Channel assignment in cellular communication using a great deluge hyper-heuristic. In: Proceedings. 12th IEEE International Conference on Networks (ICON 2004), vol. 2, pp. 769–773. IEEE, Nov 2004
24. Kilby, P., Prosser, P., Shaw, P.: Dynamic VRPs: a study of scenarios, pp. 1–11. Technical Report, University of Strathclyde (1998)

25. Kiraz, B., Topcuoglu, H.R.: Hyper-heuristic approaches for the dynamic generalized assignment problem. In: 2010 10th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 1487–1492, IEEE, Nov 2010
26. Kiraz, B., Uyar, A.Ş., Özcan, E.: An investigation of selection hyper-heuristics in dynamic environments. In: European Conference on the Applications of Evolutionary Computation, pp. 314–323. Springer, Berlin (2011)
27. Kiraz, B., Etaner-Uyar, A.Ş., Özcan, E.: Selection hyper-heuristics in dynamic environments. *J. Oper. Res. Soc.* **64**(12), 1753–1769 (2013)
28. Kiraz, B., Etaner-Uyar, A.Ş., Özcan, E.: An ant-based selection hyper-heuristic for dynamic environments. In: European Conference on the Applications of Evolutionary Computation, pp. 626–635. Springer, Berlin (2013)
29. Köle, M., Etaner-Uyar, A.Ş., Kiraz, B., Özcan, E. (2012.). Heuristics for car setup optimisation in torcs. In: 2012 12th UK Workshop on Computational Intelligence (UKCI), pp. 1–8, IEEE, Sept 2012
30. Loiacono, D., Cardamone, L., Lanzi, P.L.: Simulated car racing championship competition software manual (2011)
31. Martello, S., Toth, P.: Knapsack problems: algorithms and computer implementations. Wiley-Interscience series in discrete mathematics and optimization. (1990)
32. Nguyen, T.T., Yang, S., Branke, J.: Evolutionary dynamic optimization: a survey of the state of the art. *Swarm Evol. Comput.* **6**, 1–24 (2012)
33. Nguyen, S., Mei, Y., Zhang, M.: Genetic programming for production scheduling: a survey with a unified framework. *Complex. Intell. Syst.* **3**(1), 41–66 (2017)
34. Ozcan, E., Uyar, S.E., Burke, E.: A greedy hyper-heuristic in dynamic environments. In: Proceedings of the 11th Annual Conference Companion on Genetic and Evolutionary Computation Conference: Late Breaking Papers, pp. 2201–2204. ACM, July 2009
35. Remde, S., Dahal, K., Cowling, P., Colledge, N.: Binary exponential back off for tabu tenure in hyperheuristics. In: European Conference on Evolutionary Computation in Combinatorial Optimization, pp. 109–120. Springer, Berlin (2009)
36. Richter, H.: Dynamic fitness landscape analysis. In: Evolutionary Computation for Dynamic Optimization Problems, pp. 269–297. Springer, Berlin (2013)
37. Sabar, N.R., Ayob, M., Kendall, G., Qu, R.: Automatic design of a hyper-heuristic framework with gene expression programming for combinatorial optimization problems. *IEEE Trans. Evol. Comput.* **19**(3), 309–325 (2015)
38. Sabar, N.R., Ayob, M., Kendall, G., Qu, R.: A dynamic multiarmed bandit-gene expression programming hyper-heuristic for combinatorial optimization problems. *IEEE Trans. Cybern.* **45**(2), 217–228 (2015)
39. Soria-Alcaraz, J.A., Ochoa, G., Sotelo-Figueroa, M.A., Burke, E.K.: A methodology for determining an effective subset of heuristics in selection hyper-heuristics. *Eur. J. Oper. Res.* **260**(3), 972–983 (2017)
40. Soria-Alcaraz, J.A., Espinal, A., Sotelo-Figueroa, M.A.: Evolvability metric estimation by a parallel perceptron for on-line selection hyper-heuristics. *IEEE Access.* **5**, 7055–7063 (2017)
41. Topcuoglu, H.R., Ucar, A., Altin, L.: A hyper-heuristic based framework for dynamic optimization problems. *Appl. Soft Comput.* **19**, 236–251 (2014)
42. Uludağ, G., Kiraz, B., Etaner-Uyar, A.Ş., Özcan, E.: A Framework to Hybridize PBIL and a Hyper-heuristic for Dynamic Environments. In: International Conference on Parallel Problem Solving from Nature, pp. 358–367. Springer, Berlin (2012)
43. Uludag, G., Kiraz, B., Etaner-Uyar, A.S., Ozcan, E.: Heuristic selection in a multi-phase hybrid approach for dynamic environments. In: UKCI, pp. 1–8, Sept (2012)
44. Uludağ, G., Kiraz, B., Etaner-Uyar, A.Ş., Özcan, E.: A hybrid multi-population framework for dynamic environments combining online and offline learning. *Soft. Comput.* **17**(12), 2327–2348 (2013)
45. van der Stockt, S., Engelbrecht, A.P.: Analysis of hyper-heuristic performance in different dynamic environments. In: 2014 IEEE Symposium on Computational Intelligence in Dynamic and Uncertain Environments (CIDUE), pp. 1–8. IEEE, Dec 2014

46. van der Stockt, S., Engelbrecht, A.P.: Analysis of global information sharing in hyper-heuristics for different dynamic environments. In: 2015 IEEE Congress on Evolutionary Computation (CEC), pp. 822–829. IEEE, May 2015
47. van der Stockt, S.A., Engelbrecht, A.P.: Analysis of selection hyper-heuristics for population-based meta-heuristics in real-valued dynamic optimization. *Swarm Evol. Comput.* (2018)
48. Wang, H., Wang, D., Yang, S.: A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems. *Soft. Comput.* **13**(8–9), 763–780 (2009)
49. Wang, M., Li, B., Zhang, G., Yao, X.: Population evolvability: dynamic fitness landscape analysis for population-based metaheuristic algorithms. *IEEE Trans. Evol. Comput.* (2017)
50. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997)
51. Yang, S., Yao, X.: Experimental study on population-based incremental learning algorithms for dynamic optimization problems. *Soft. Comput.* **9**(11), 815–834 (2005)
52. Yang, S., Yao, X.: Population-based incremental learning with associative memory for dynamic environments. *IEEE Trans. Evol. Comput.* **12**(5), 542–561 (2008)

The Dynamic Portfolio Selection Problem: Complexity, Algorithms and Empirical Analysis



Daniel A. Martínez-Vega, Laura Cruz-Reyes,
Claudia Guadalupe Gomez-Santillan, Fausto Balderas-Jaramillo
and Marco Antonio Aguirre-Lam

Abstract In the real world, problems usually involve a characteristic that is not typically considered when experimenting algorithmically; this characteristic is the change concerning time. The parameters included in many issues are not static, but dynamic, they are altered over time. Until a few years ago due to, among other things, the limited computational power to which one had access, it was very complicated for some researchers to be able to deal with problems in a dynamic way. This chapter makes a comparison between some metaheuristics when working on the issue of dynamic project portfolio selection. Also, it approaches the correspondence between the Dynamic project portfolio selection and the well-known Knapsack problem, as well as makes the demonstration, step by step, that the first one is a NP-hard problem.

Keywords Knapsack · Dynamic portfolio · Dynamic allocation of resources · DMOCcell · DNSGA-II · DSPEA2

D. A. Martínez-Vega · L. Cruz-Reyes (✉) · C. G. Gomez-Santillan · F. Balderas-Jaramillo ·
M. A. Aguirre-Lam
Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Mexico City, Mexico
e-mail: lauracruzreyes@itcm.edu.mx

D. A. Martínez-Vega
e-mail: adalbertovega@gmail.com

C. G. Gomez-Santillan
e-mail: claudia.gomez@itcm.edu.mx

F. Balderas-Jaramillo
e-mail: fausto.balderas@itcm.edu.mx

M. A. Aguirre-Lam
e-mail: marco.aguirre@itcm.edu.mx

D. A. Martínez-Vega
Tecnológico Nacional de México, Instituto Tecnológico de Tijuana, Tijuana, Mexico

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_34

1 Introduction

The dynamic part of some problems is a fundamental factor for a good solution, one of these problems is the portfolio of projects.

When working with the Project Portfolio Selection (PPS) problem dynamically, it is necessary to monitor and adjust the actions periodically, understanding by actions how the money is distributed to finance the projects that make up the portfolio [1].

In practice, heuristics models have limited avail owing to they do not consider some things like the essential dynamic character of portfolio processes [2]. In this chapter, an extension of the previous work [3] is presented, and it shows a broader investigation into state of the art. Up to our knowledge there is no detailed verification of the complexity of the PPS problem, in this chapter we carry out one, demonstrating that the PPS problem is of the NP-hard class by transforming in both directions this problem and the Knapsack problem 0/1 (KP) that has been demonstrated in [4] which is a problem of the same kind. The purpose of the demonstration is to specify the relationship between these problems and allow a future work to take advantage of the algorithms that have been proposed to solve them.

In this chapter, an analogy is made between the primary problem addressed and the knapsack problem 0/1, as well as that of the multidimensional knapsack problem (MKP).

Also, a comparison is made between three dynamic algorithms, generated based on the literature: Dynamic Multi-Objective Cellular (DMOCCell) [5], Dynamic Non-dominated Sorting Genetic Algorithm II (DNSGA-II) [6] and Dynamic Strength Pareto Evolutionary Algorithm 2 (DSPEA2) [7]. We experimentally demonstrate the benefits of our proposal and leave open the possibility that its study will apply to solve large-scale multidimensional problems and some related problems as Knapsack.

Finally, this document consists of six sections in the following order: Introduction, state of the art, problems selection of project portfolio, demonstration of the complexity of the project portfolio selection problem, experimental study and conclusions.

2 State of the Art

An exhaustive survey of the literature was carried out with the purpose of analyzing those works that had focused on the solution of the Dynamic portfolio selection problem. To our knowledge, there is a limited amount of work focused on this problem, which generates a large research space for the present and future. Among the jobs that were analyzed, the following describes those that present characteristics that make them stand out from the rest and that can serve as a guide for future research.

Uncertain mean-variance model for dynamic project portfolio selection problem with divisibility [8]. Article by Li, Wang, Yan, and Zhao, published in 2018, addresses the issue of dynamic portfolio selection of projects with divisibility (partial support).

In that chapter, the fact that an investment made in a project must be regarded as with the passage of time. First because once invested, 100% of the investment is not recovered in case of not continuing with the project, and second because if a project stops being supported, a certain percentage of reimbursement must be obtained that can be invested in other projects. The authors propose a mathematical model of mixed whole programming called UDPPSD_R.

Management of dynamic project portfolio [9]. Article published in 2014 by Fiala, Arlt, and Marketa, in that, an Analytical Network Processing model (ANP) with clusters (of projects, resources, criteria and time) is proposed, which tries to solve the managing of dynamic project portfolio problem. The document suggests a hybrid procedure for the adjustment of time-dependent priorities. The procedure is based on a combination of functions of comparison by pairs and exponential smoothing; each period is applied because the synergy between projects changes with the passage of time and the best combination of them must be recalculated. The methodology is verified in the projects of an engineering company.

Optimal dynamic portfolio selection for projects under a competence development model [10]. Gutjahr presented this work in 2011, he proposes a mathematical model for a dynamic portfolio selection, in which each of the projects is classified, and the accumulated time allocated to each of these classes is what integrates the project portfolio. The primary objective is to maximize the total time invested in each of the project classes since it is known in advance the total time required to complete each of them and what is sought is to achieve the highest percentage of completion of all the projects to obtain the particular more significant benefit. That work considers the increase and decrease of the capacities and abilities of the personnel in charge of the realization of the projects as it happens in the real world. The tests were done based on a real-world situation provided by the company with mixed capital origin E-Commerce Competence Center Austria (EC3).

A Framework for Dynamic Project Portfolio Management [11]. Work done by Ei-Lin and Hsi-Mei in 2007, that work proposes a dynamic framework of the portfolio of projects of NPD, which tries to emulate the periodic review of a portfolio in practice. It considers especially the uncertainty included in the main problem, the complex interactions between the projects and the possibility of making control decisions during the development of the project. A process called Stage-gate is used, which, supported by a decision tree approach for decision making, determines by submitting to a set of filters which projects will be financed in the portfolio. Subsequently, a process of risk control and changes in the portfolio is carried out, where a constant detection of changes is carried out, which if found, produce immediate actions, doing new projects to be incorporated into the portfolio, which some already it is included leave or modify the support which is receiving.

As can be observed, each of these works has a particularity very different from the rest and that resembles a lot to reality, from our point of view, those characteristics made them stand out in literature. The rest of the analyzed works despite not having contained traits that we considered of great significance does not mean that for other

authors they could not have it. For this chapter, as well as for future work on DPPS, up to now, these are the documents that could provide a plus to the way of dealing with the problem.

3 Project Portfolio Selection Problems

In this section, a basic definition of the PPS problem is first given. Subsequently, the description of the dynamic problem of the project portfolio is made, in this case, the dynamic part is contributed by the dynamic allocation of resources attached problem.

3.1 *Dynamic Project Portfolio Selection Problem*

Both in the public and private sectors, a task of management with great importance is to evaluate a set of projects that compete for financial support, to select those that contribute the maximum benefit to the organization. This subset constitutes a project portfolio [12]. PPS's mathematical model in detail is found in [13].

Over time, the problem of the project portfolio has been merged with various problems, which increases its complexity while allowing the problem to be studied in a way that is closer to reality. For example, it has been addressed along with the problem of scheduling [14], the problem of synergy [15], among others.

Several problems have been studied immersed in the PPS problem. Therefore, more than one has given the characteristic of being dynamic, that is, to address the changes that could arise over time, such as it happens in the real thing. In this case, the attached problem that includes the dynamism in the portfolio problem is that of the dynamic allocation of resources.

The problem of dynamic allocation of resources seeks to permanently monitor those variables that could be affected over time, for example, the costs of the projects, the available budgets, and so on. Thus, when detecting a change, the necessary adjustments are applied so that the problem can be controlled while obtaining consistent data.

In this section, the mathematical model used to address the problem of Dynamic project portfolio selection (DPPS) is explained.

Let T periods to analyze, in each of these periods, there are N projects of interest that meet certain minimum requirements of acceptability to be supported. Each project has associated a region A , an area G and a cost $C(t)$, the latter may vary over time:

$$\begin{aligned} A &= (a_1, a_2, \dots, a_k), \\ G &= (g_1, g_2, \dots, g_r), \\ C(t) &= (c_1(t), c_2(t), \dots, c_N(t)), \end{aligned}$$

where $c_j(t)$ is an amount of money that fully satisfies the budget requirements of the j project at the time t [16].

Below are the equations belonging to the DPPS problem.

Let $X = (x_1, x_2, \dots, x_N)$, the set of N projects where:

$$x_i = \begin{cases} 1, & \text{if the } i\text{th project is funded} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Fitness function:

$$\text{Max}_{X \in R_F} (Z(X)), \quad (2)$$

$$Z(X) = (z_1, z_2, z_3, \dots, z_p), \quad (3)$$

$$z_j(X) = \sum_{i=1}^N x_i f_{j,i}(t) \quad \forall_{t,j}, \quad (4)$$

Subject to:

$$\left(\sum_{i=1}^N x_i c_i(t) \right) \leq B(t) \quad \forall_t, \quad (5)$$

$$B_{l_{\min}}(t) \leq B_l(t) \leq B_{l_{\max}}(t) \quad \forall_{t,l}, \quad (6)$$

$$B_l(t) = \sum_{i=1}^N x_i c_i(t) a_{i,l} \quad \forall_{t,l}, \quad (7)$$

$$B_{r_{\min}}(t) \leq B_r(t) \leq B_{r_{\max}}(t) \quad \forall_{t,r}, \quad (8)$$

$$B_r(t) = \sum_{i=1}^N x_i c_i(t) g_{i,r} \quad \forall_{t,r}. \quad (9)$$

The evaluation of the projects is one of the most complex tasks; this considers what each project contributes to the fitness function. The level of contribution (benefit) of each project x_i to the different objectives can be represented by the vector $f_{x_i}(t) = (f_{x_i,1}(t), f_{x_i,2}(t), \dots, f_{x_i,p}(t))$; which is called the benefit vector for project i in the period t , considering P objectives.

Let the matrix $F(t)$ of dimension $N \times p$ be the profit matrix (Table 1), where p is the total number of objectives, N is the total number of projects and t , is the period. Each row represents the benefit vector for the i th project.

Table 1 Profit matrix F for the period t

Objectives	Projects			
	1	2	...	N
1	$f_{1,1}(t)$	$f_{1,2}(t)$...	$f_{N,1}(t)$
2	$f_{2,1}(t)$	$f_{2,2}(t)$...	$f_{N,2}(t)$
...
p	$f_{p,1}(t)$	$f_{p,2}(t)$...	$f_{N,p}(t)$

Let $B(t)$ be the total amount of financial resources available for distribution to different projects in the period t . Since each project has a cost $c_i(t)$, any project portfolio must comply with the budget constraint in Eq. 5.

Equation 6 assumes as possible that there are budgetary restrictions for each investment area. So, if $B_l(t)$ is the budget dedicated to area l in the period t , and there is a minimum budget $B_{l_{min}}(t)$ and a maximum budget $B_{l_{max}}(t)$ established by period.

The budget constraint by the area that each portfolio must fulfill is given by Eq. 7, where a_{i_l} is a binary variable that indicates whether project i belongs to the area l .

On the other hand, each project benefits a region, and as with the areas, there is a minimum budget $B_{r_{min}}(t)$ and a maximum budget $B_{r_{max}}(t)$ per established region in each period, where the budget by region for each portfolio is given by Eq. 8.

In Eq. 9, g_{i_r} is a binary variable that indicates whether project i belongs to region r or it does not.

The quality of a portfolio X depends on the benefits of its projects and is represented by the quality vector $Z(X)$, whose components are at the same time quality values in relation to each of p objectives of the projects (Eqs. 2, 3 and 4).

Let R_F be the feasible portfolio space, the solution of the portfolio selection problem is to find one or more portfolios satisfying (Eq. 2). That is, the only accepted solutions will be those that meet the constraints (5)–(9).

3.2 NP-Hard Subproblems

Several problems can be related to PPS, one of them is the Knapsack problem (KP). It is a problem in combinatorial optimization: It is a set of elements, each with a weight and a value, from which a decision maker must select the subset that occupies the least space during the most significant benefit. Its name comes from the problem faced by someone who is restricted by a knapsack with limited space and must fill it with the most valuable items.

The study of the KP has taken place for more than a century, with early works dating as far back as 1897 [17]. The name “knapsack problem” is associated with the mathematician Tobias Dantzig [18] and refers to the common problem of storing items of higher value or utility without exceeding the space available.

The KP has had several formulations over time, two of them widely studied in the literature are described below.

Knapsack 0/1. The most common formulation of the KP. This problem considers n items which must be packed in a knapsack of capacity c . Each item i has associated a profit p_i and a weight w_i , and the objective is to maximize the profit sum of the items within knapsack while the weight sum does not exceed c , KP is NPC [19].

Multidimensional Knapsack. Gavish and Pirkul gave this variation in [20], this problem raises the following difference concerning the KP, the capacity restrictions are not given only concerning a total, but to different sums that belong to each of the sections in which the knapsack is divided.

3.3 *Demonstration of PPS Complexity*

Algorithmic complexity represents the number of computational resources (temporal) that an algorithm needs to solve a problem and therefore allows to determine its efficiency.

The NP-completeness theory provides several methods to prove that a problem is so complex that it fits into a select group of people recognized for their complexity to the extent that they have surprised scientists for years, these problems are cataloged as Nondeterministic Polynomial time Complete (NPC).

Definition of the decision problem version of PPS

Given the projects p_1, p_2, \dots, p_n , a portfolio of projects is a subset of k projects ($1 \leq k \leq n$) that meet a series of budget constraints.

Input: A set of projects p_1, p_2, \dots, p_n , with their respective costs and benefits, and a budget constraint T .

Question: Is there a project portfolio such that its benefit is greater than a given constant?

The internal representation of the problem

Two data structures are used in the algorithm: one to store the given projects, their characteristics (cost and benefit per objective) and another structure to store the restrictions.

Let n be the number of total projects.

An array of binary data X is used to store each solution, each row of this matrix represents a portfolio of projects x_i , where a 1 is stored in case the project is included in the portfolio or a 0 otherwise.

The restrictions are stored in an independent structure R , which will be used only to verify that a project portfolio is feasible.

Complexity of PPS

Theorem: The PPS decision problem is NPC.

To demonstrate that a problem belongs to the NPC class, the following actions must be taken:

Step I. *Prove that the problem is NP:*

1. Define a data structure to represent the candidate solutions.

$$x[i] \text{ where: } i = 1, 2, 3, \dots, n$$

2. Construct a random algorithm of complexity $O(n)$ that generates a candidate solution.

```
for  $i = 1$  to  $n$   
 $x[i] = (i)$ 
```

3. Construct a deterministic algorithm of complexity $O(n)$ to verify that each candidate solution meets the specifications in the problem.

```
for  $i = 0$  to  $n$  // Verify that  $x[i]$  is a portfolio of size  $n$   
  if ( $x[i] == 1$ )  
     $budget += projects[i][0]$   
  if ( $budget > total\_budget$ ) // Verify the budget restriction  
    return false  
  return true; // Portfolio  $x$  is feasible
```

With the previous thing, it is verified that the PPS problem belongs to the category NP.

Step II. *Validate that the PPS problem belongs to the NPC category*

A problem P' known to be NPC is selected, a reduction function is generated from P' to P and a function of inverse reduction (P to P'). In the case of the PPS problem, the Knapsack problem was used as P' [21] due to the similarity in its internal structure (Knapsack $<_p$ Project portfolio selection).

Let M be a feasible vector of decision binary variables in the search space (knapsack), which stores a subset of elements (objects), each element in M has a benefit (q) and an associated weight (w), in addition, there is a maximum weight as a restriction associated with the problem which indicates the weight capacity of the knapsack. The objective is to find the subset of elements in M that maximize the total benefits (Eq. 10) without exceeding the load capacity C (Eq. 11). The decision problem version of KP is the following: does there exist M such that

$$\sum_{j=1}^k q_{i,j} m_j \geq K \quad \forall_i, \tag{10}$$

for any positive integer constant K .

Subject to:

$$\sum_{j=1}^k w_j m_j \leq C. \tag{11}$$

Equation 10 represents the benefits to be maximized. w_i in Eq. 11 is the weight associated to the item i , on the other hand m_i is a binary variable that represents the element i in M which will have the value of one if it has been introduced to the knapsack, or zero otherwise, this restriction indicates that the sum of all the weights of the packages included in each knapsack should not exceed the maximum cost C , otherwise said solution will be infeasible.

On the other hand, PPS problem pretend to find the subset of elements in X (project portfolio), where each element has a benefit (b) and an requires a cost (c), in addition, there is a maximum budget as a restriction associated with the problem which indicates the financing capacity of the project portfolio. The objective is to find the subset of elements in X that maximize the total benefits (Eq. 12) without exceeding the budget B (Eq. 13). The decision problem version of PPS is the following: does there exist X such that

$$\sum_{i=1}^N f_{j,i} x_i \geq K \quad \forall_j, \tag{12}$$

for any positive integer constant K .

Subject to:

$$\sum_{i=1}^N x_i c_i \leq B. \tag{13}$$

Table 2 shows the correspondence between the PPS and KP problems, which indicates that if there is a portfolio X , there is also a knapsack M that does the same with the KP. Conversely, the PPS problem can become the KP problem, look

Table 2 Correspondencia between decision problems: PPS to KP

Variable	Equation	Description	Correspond to		
			Variable	Equation	Description
c_i	13	Cost per project	w_i	11	Weight per item
x_i	12, 13	Project of portfolio	m_i	10, 11	Item of Knapsack
$f_{j,i}$	12	Benefit	$q_{i,j}$	10	Benefit
B	13	Budget	C	11	Capacity

at the elements of Table 2 in reverse. Therefore, the PPS decision problem is NPC. Besides, The PPS optimization problem is NP-hard since its decision problem version is NP-complete.

3.4 Solution Algorithms

For the experimental part of this chapter three dynamic metaheuristics were used, which were implemented based on the original algorithms of each of them, modifying them according to how the literature indicates that the algorithms work for dynamic problems [22]. Next, a brief description of each of these algorithms is made.

MOCeII. It's a cell-type algorithm which uses an external file to store the non-dominated individuals found during the search. The most relevant feature of MOCeII concerning the other existing cellular approaches is the feedback of individuals from the file to the population [5].

NSGA-II. It is a genetic algorithm proposed by Deb in the year 2000, which has the properties of a fast non-dominated sorting procedure, an approach without parameter and an elitist strategy. It is one of the most outstanding genetic algorithms for the high quality of solutions that it offers especially for small and medium scale problems [6].

SPEA2. It is an improved elitist multi-objective evolutionary algorithm that uses an enhanced fitness assignment strategy unlike its predecessor (SPEA) as well as new techniques for file truncation and density-based selection [7].

4 Experimentation and Result

In this section, we present the case of study, and the results of the testing carried out. This experimentation consisted of solving a set of instances of the DPPS problem, generated by its code.

The characteristics of the equipment used for this experimentation is the following: a processor Intel Inside i3 2.1 GHz, 6 GB of memory RAM and Windows 10 O.S.

The comparison is based on the error between the results of the metaheuristics DMOCeII, DNSGA-II, and DSPEA2; the parameters used in the experimental configuration of each metaheuristic were taken from the literature, specifically from [23].

The stop criterion was the number of evaluations of the objective function, for this experiment it was 25,000.

Table 3 shows the instances used for the experimentation of this work. The table has three columns, the first one (on the left side) stores the names of the instances used, in the middle are the number of projects to be managed, and the third one (on the right side) is the number of periods to be solved.

Table 3 Instances

Name	Number of projects	No. of years to calculate
DPPS_20p_5y	20	5
DPPS_20p_10y	20	10
DPPS_20p_20y	20	20
DPPS_100p_5y	100	5
DPPS_100p_10y	100	10
DPPS_100p_20y	100	20

4.1 Results

For each metaheuristic were made 30 runs of each instance described in Table 3. Figures 1 and 2 show the results and the average error of the solutions obtained.

As can be observed in all cases, the metaheuristics DMOCcell, DNSGA-II and DSPEA2 present very similar results, however, as the complexity of the instances increases, considering the number of projects that are handled and the number of years or periods that are calculated with respect to the dynamic part, DNSGA-II begins to present better results than the other pair of algorithms. The above is reflected in Figs. 1 and 2, that is, both in the average of the results and the error rate.

Giving greater weight to this experiment, a statistical analysis of the error rate was performed, the *p-value* used was 0. The tests conducted were the Friedman non-parametric test. In cases where significant differences were found in the Friedman

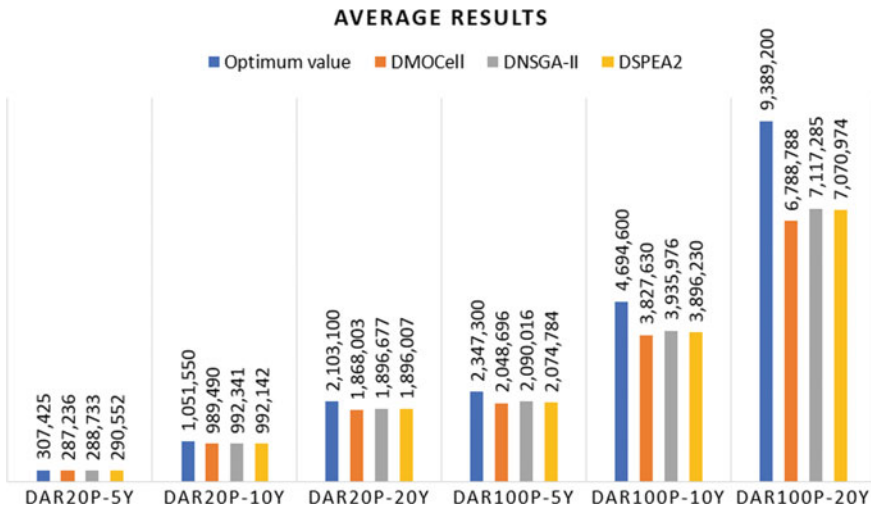


Fig. 1 Average results of 30 runs in each metaheuristic

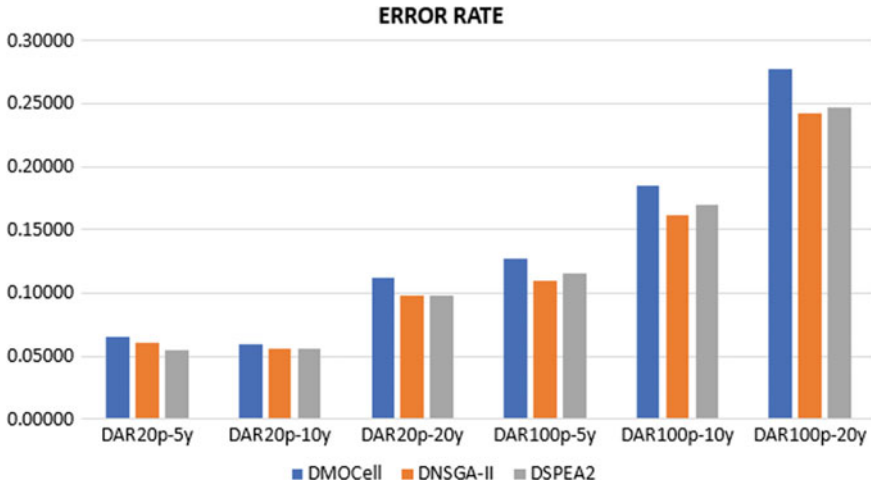


Fig. 2 The error rate of 30 runs in each metaheuristic

test, we proceeded to apply the Post hoc Holm test, which is widely used in the scientific community.

The null hypothesis used to Friedman test was (H_0): “the mean of the results of two or more algorithms is the same and a significance level of 0.05”.

In the case of post hoc Holm test, we establish the null hypothesis (H_0): “the mean of the results of each pair of algorithms compared is equal and with a significance level of 0.05”.

In all cases analyzed (all instances) the H_0 of the Friedman test was rejected, which indicates that the difference between the performance of the algorithms for each instance is significant.

We proceeded to apply the post hoc test to determine the cause of this significant difference, the results where H_0 was accepted are shown in Table 4.

Based on the above data we infer that there is a significant difference in performance between the performance of the algorithms compared for the instances used in this experiment. Besides, in five of the six cases analyzed, the DSPEA2 and

Table 4 Results of post hoc tests

Instance	Comparison	Statistic
20p10y	DSPEA2 versus DMOCcell	1.1
	DMOCcell versus DNSGA-II	1
	DSPEA2 versus DNSGA-II	0.1
20p20y	DSPEA2 versus DNSGA-II	0.6
10p5y	DSPEA2 versus DNSGA-II	1.7
10p20y	DSPEA2 versus DNSGA-II	1.1

DNSGA-II algorithms showed similar performance, obtaining better results than DMOCcell.

5 Conclusions and Future Work

In this chapter, the result of a state-of-the-art investigation focused on the dynamic project portfolio problem is presented. From this research derives what is given in Sect. 2, a summary based on the point of view of the authors of those works that are considered to have significant relevance and particularities that make them different from the rest of the works in state of the art focused on this topic.

Because the verification of the complexity of the PPS problem to our knowledge is not in state of the art, beyond comments that only assume that it is NP-hard type, in Sect. 3.3 this demonstration is made in detail based on the format by Garey and Johnson [17]. This demonstration was carried out to specify the relationship between the KP and PPS problems, so that in future works the solution capacity of their corresponding algorithms can be extended.

Also, an experiment is presented for which instances of the DPPS problem were used in which the dynamic part is provided by the DAR problem coupled with the PPS problem. The instances used are small, medium and large scale, taking as a reference to the number of projects used and reported by other authors in the literature. The result of this experiment shows that among the compared algorithms, despite having a significant difference in the quality of their solutions for the instances used, it is concluded that one of the algorithms cannot be determined to be the best everyone. For these cases, the algorithm NSGA-II obtained the best results in 5 of the 6 test instances.

About the performance of the algorithms is not absolute, but it shows the feasibility of our proposal is not perfect; it should be noted that these results are for the set of test instances used in this work. With the purpose of obtaining more meaningful conclusions in future work, it must be done more significant experimentation with a higher number of larger instances regarding the number of objectives, the number of projects and periods.

References

1. Santos, C.P.A., Lopez-Sanchez, I.A.: U.S. Patent Application No. 14/485,339 (2014)
2. Pajares, J., López, A., Araúzo, A., Hernández, C.: Project portfolio management, selection, and scheduling. Bridging the gap between strategy and operations. In: XIII Congreso de Ingeniería de Organización, pp. 1421–1429, April 2009
3. Martínez-Vega, D.A., Cruz-Reyes, L., Gomez-Santillán, C., Rangel-Valdez, N., Rivera, G., Santiago, A.: Modeling and project portfolio selection problem enriched with dynamic allocation of resources. In: Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications, pp. 365–378. Springer, Cham (2018)

4. Hansen, P.: Bicriterion path problems. In: *Multiple Criteria Decision-Making Theory and Application*, pp. 109–127. Springer, Berlin (1980)
5. Nebro, A.J., Durillo, J.J., Luna, F., Dorronsoro, B., Alba, E.: Design issues in a multiobjective cellular genetic algorithm. In: *International Conference on Evolutionary Multi-criterion Optimization*, pp. 126–140. Springer Berlin (2007)
6. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.A.M.T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
7. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: improving the strength pareto evolutionary algorithm (2001)
8. Li, X., Wang, Y., Yan, Q., Zhao, X.: Uncertain mean-variance model for dynamic project portfolio selection problem with divisibility. In: *Fuzzy Optimization and Decision Making*, pp. 1–20 (2018)
9. Fiala, P., Arlt, J., Arltova, M.: Management of dynamic project portfolio. *Int. J. Innov. Manage. Technol.* **5**(6), 455 (2014)
10. Gutjahr, W.J.: Optimal dynamic portfolio selection for projects under a competence development model. *OR Spectrum* **33**(1), 173–206 (2011)
11. Ei-Lin, H., Hsi-Mei, H.: A framework for dynamic project portfolio management. Master's degree thesis, National Chiao Tung University, Hsinchu, Taiwan, Republic of China (2007)
12. Nebro, A., Luna, F., Dorronsoro, B., Durillo, J. Un algoritmo multiobjetivo basado en búsqueda dispersa. In: *Quinto Congreso Español de Metaheurísticas, Algoritmos Evolutivos y Bioinspirados (MAEB 2007)*, pp. 175–182 (2007)
13. Martínez-Vega, D.A., Cruz-Reyes, L., Gomez-Santillan, C., Rangel-Valdez, N., Rivera, G., Santiago, A.: Modeling and project portfolio selection problem enriched with dynamic allocation of resources. In: *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, pp. 365–378. Springer, Cham (2018)
14. Martínez-Vega, D.A., Cruz-Reyes, L., Rangel-Valdez, N., Santillán, C.G., Sánchez-Solís, P., Villafuerte, M.P.: Project portfolio selection with scheduling: an evolutionary approach. *Int. J. Comb. Optim. Probl. Inform.* **10**(1), 25–31 (2019)
15. Rivera, G., Gómez, C.G., Fernández, E.R., Cruz, L., Castillo, O., Bastiani, S.S.: Handling of synergy into an algorithm for project portfolio selection. In: *Recent Advances on Hybrid Intelligent Systems*, pp. 417–430. Springer, Berlin (2013)
16. Jain, H., Deb, K.: An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization. In: *International Conference on Evolutionary Multi-criterion Optimization*, pp. 307–321. Springer Berlin (2013)
17. Garey, M.R., Johnson, D.S.: *Computers and intractability: a guide to the theory of np-completeness (series of books in the mathematical sciences)*, ed. Computers and Intractability, 340 (1979)
18. Kellerer, H., Pferschy, U., Pisinger, D.: *Knapsack problems 2004* (2003)
19. Pisinger, D.: A minimal algorithm for the 0-1 knapsack problem. *Oper. Res.* **45**(5), 758–767 (1997)
20. Gavish, B., Pirkul, H.: Efficient algorithms for solving multi constraint zero-one knapsack problems to optimality. *Math. Program.* **31**(1), 78–105 (1985)
21. Bartlett, M., Frisch, A.M., Hamadi, Y., Miguel, I., Tarim, S.A., Unsworth, C.: The temporal knapsack problem and its solution. In: *International Conference on Integration of Artificial Intelligence (AI) and Operations Research (OR) Techniques in Constraint Programming*, pp. 34–48. Springer, Berlin (2005)
22. Deb, K., Karthik, S.: Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. In: *International Conference on Evolutionary Multi-criterion Optimization*, pp. 803–817. Springer, Berlin (2007)
23. Durillo, J.J., Nebro, A.J.: jMetal: a Java framework for multi-objective optimization. *Adv. Eng. Softw.* **42**(10), 760–771 (2011)

A Novel Dynamic Multi-objective Evolutionary Algorithm with an Adaptable Roulette for the Selection of Operators



Héctor Joaquín Fraire Huacuja, Eduardo Rodríguez del Angel, Juan Javier González Barbosa, Alejandro Estrada Padilla and Lucila Morales Rodríguez

Abstract In this chapter, the optimization of Dynamic Multi-Objective Problems (DMOP) is approached. To solve this kind of problems several evolutionary algorithms with a static selection of operators are reported in the literature. In this work, a new evolutionary algorithm with that an online operator selector is proposed. The operator choice is guided by a self-adapting roulette that modifies the probabilities of usage for each operator. The evolutionary algorithm proposed follows the classical generational scheme of an evolutionary algorithm, but each offspring is constructed by selecting an operator from an operator's pool based on a probability regulated by the roulette. A series of experiments were done to assess the performance of the proposed algorithm that includes a set of state-of-the-art algorithms, a set of standard instances and statistical hypothesis tests to support the conclusions.

Keywords Dynamic multi-objective optimization · Evolutionary algorithm · Adaptive algorithm

1 Introduction

The optimization of Multi-objective problems in many of the real-world applications frequently involves changes in its objectives, constraints, and parameters over time.

H. J. F. Huacuja (✉) · E. R. del Angel · J. J. G. Barbosa · A. E. Padilla · L. M. Rodríguez
Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Mexico
e-mail: automatas2002@yahoo.com.mx

E. R. del Angel
e-mail: rodela22@hotmail.com

J. J. G. Barbosa
e-mail: jjgonzalezbarbosa@hotmail.com

A. E. Padilla
e-mail: aestrada1993@hotmail.com

L. M. Rodríguez
e-mail: lmoralesrdz@gmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_35

This kind of problems is named Dynamic Multi-objective Optimization Problems (DMOP). The development of solution methods for these problems is a recent and active research area. A common approach to solve these problems is evolutionary algorithms.

The main contribution of this work is a new dynamic multi-objective evolutionary algorithm. It implements a self-adapting roulette for the dynamic selection of the operators that regulates the usage probability of each operator. The roulette operator rewards the operator with the highest efficiency rate of the last generation, incrementing its distribution on the roulette for the next generation and punish the operator with the lowest performance. To the best of our knowledge, this first time that an evolutionary algorithm, based on the selection of operators regulated by a self-adapting probability distribution, is used to solve dynamic multi-objective optimization problems.

The operators included in the algorithm are taken from state-of-the-art algorithms: NSGA-II and Differential Evolution. The performance of the proposed algorithm is compared against the dynamic multi-objective versions of those algorithms in a selected DMOP set that includes problems with two and three objectives.

The structure of this chapter is as follows. First, the definitions related to the field of Dynamic Multi-objective Optimization (DMO) are introduced in Sect. 2. Section 3 describes the proposed Dynamic Multi-Objective Evolutionary Algorithm with an Adaptable Roulette for the Selection of Operators and gives an in-depth explanation of the proposed adaptable roulette operator. Section 4 briefly describes the classic algorithms from the state of the art chosen for comparison. Section 5 presents the experiments performed. Finally, the obtained results are summarized in Sect. 6. This chapter ends with the main conclusions of this work.

2 Dynamic Multi-objective Optimization Problem Definitions

In this section, the most relevant definitions for Dynamic Multi-objective Optimization (DMO) field are presented.

A DMO problem (DMOP) is defined as the problem of finding a vector of decision variables $\vec{x}(t)$, that satisfies a restriction set and optimizes a function vector $\vec{F}(\vec{x}, t)$ that changes over time t [1]. The minimization problem can be formally defined as:

$$\begin{aligned} \text{Min } \vec{F}(\vec{x}, t) &= \{f_1(\vec{x}, t), f_2(\vec{x}, t), \dots, f_M(\vec{x}, t)\} | \vec{x} \in X^n \\ \text{s.t. } g(\vec{x}, t) &> 0, h(\vec{x}, t) = 0 \end{aligned} \quad (1)$$

where g and h represent the set of inequality and equality constraints. The decision vectors satisfying the constraints and variable bounds constitute a feasible solution.

The objectives included in a DMOP are usually conflicting among each other, so it is not possible to make a direct comparison. Hence, the dynamic Pareto dominance

is defined as one comparison method between different vector solutions for a DMOP. A decision vector $\vec{x}_i(t)$ is said to dominate other vector $\vec{x}_j(t)$ at time t , if both the following conditions are true:

1. The solution $\vec{x}_i(t)$ is no worse than $\vec{x}_j(t)$ in all objectives.
2. The solution $\vec{x}_i(t)$ is strictly better than $\vec{x}_j(t)$ in at least one objective.

The symbol $<$ represents the Pareto dominance relation of a minimization problem.

At a given time t the optimal Pareto Front $PF(t)^*$ is composed by the set of Pareto optimal solutions concerning the objective space defined for the time t such that

$$PF(t)^* = \left\{ \vec{F}_i(t)^* \mid \nexists \vec{F}_j(t) < \vec{F}_i(t)^*, \vec{F}_j(t) \in F^M \right\} \tag{2}$$

For any DMOP where the time change in the optimization problem is gradual in t , a time window is defined as the number of iterations T_t , to track the optimal Pareto Front before the problem definition changes [1]. The problem definition maintains to be constant during the time window. The change frequency is defined as the number of changes that a time window produces on the DMOP, the smaller the time window, the more changes are produced on the problem (big change frequency). Oppositely, a large time window produces fewer changes.

3 Dynamic Multi-objective Evolutionary Algorithm with an Adaptable Roulette for the Selection of Operators

In this section, the proposed evolutionary algorithm for DMOP's is described. The main features of the algorithm can be summarized as:

- The algorithm applies a different set of operators to generate the offspring solutions during each iteration. The operators used are: sbx crossover, differential evolution crossover, polynomial mutation and uniform mutation.
- The operators are dynamically chosen using a self-adaptable roulette that modifies the probability of each operator during the algorithm execution, based on the performances of each of the operators during previous generations.
- The algorithm uses a set of sentry solutions to maintain a continuous search process to identify changes in the definition of the problem to be optimized. This technique is proposed in [1] and is used in different algorithms [2-4].

The use of a set of operators in tandem with an adaptive selection controller seeks to guide the search towards solutions that maintain diversity and quality during the execution of the algorithm even after having identified changes in the problem.

The continuous need to correctly identify the changes in the definition of the DMOPs is the main justification for using the third mentioned feature.

3.1 Main Algorithm

Algorithm 1 includes the pseudo-code of the proposed algorithm. The code of the algorithm is composed of three main steps.

Step 1 Initialization. Firstly, it creates an Initial random-generated population (line 2), then the algorithm defines the sentry indexes, the algorithm uses a set of five sentry solutions distributed evenly throughout the population (line 3).

Before the start of the main loop, the sentry solutions are evaluated, and objective values are stored in an array (line 4). Finally, the probability for selecting each of the operators due to the roulette is initialized evenly distributed (line 5).

Step 2 Main loop. The algorithm uses the solution construction scheme of a classical population evolutionary algorithm, where a set of solutions is produced during each of the generations (lines 12 through 19), which will join the current best population, result of the previous generation (line 20).

The new population (size $2n$) will be sorted taking two criteria (line 21): Pareto ranking, and Crowding Distance. The resulting population will be reduced to its size of “ n ” elements by discarding the worst ranked solutions (line 22). The remaining population will be used during the next generation for the construction of the next solutions until the stopping criterion is reached. Each generation, the sentry solutions are evaluated again, and the objective values are stored in an array (line 24).

Step 2.1 Search for changes in the DMOP definition. At the beginning of each generation, before the recombination operators are applied, the sentinel solutions are evaluated (line 7) and stored in a different matrix called *NewSentryValues*, so they can be compared with the matrix with the old stored values (line 8). If the definition of the problem does not change between two consecutive generations, the values will be the same in both arrangements, but if a change occurs, differences between the arrangements will be found. When a change is detected, a mechanism of hypermutation is used to introduce diversity in the solutions. Also, the probability for selecting each of the operators due to the roulette is evenly assigned.

Algorithm 1 *Dynamic Multi-Objective Evolutionary Algorithm with an Adaptable Roulette*

Input: Problem to solve, the stopping criterion, population size and the set of recombination operators (Operators).

Output: The computed Pareto front approximation for the considered problem.

Step 1 Initialization

```

1  Initialize(Problem, Operators)
2  Population ← RandomInitialPopulation()
3  SentryPopulation ← SelectSentryIndexes()
4  OldSentryValues ← Evaluate(SentryPopulation)
5  Roulette.AssignUniformDistribution()

```

Step 2 Main loop

```

6  for actualGen = 1 to GenerationCount do
    Step 2.1 Search for changes in the DMOP definition
7    NewSentryValues ← Evaluate(SentryPopulation)
8    if NewSentryValues ≠ OldSentryValues then
9      Population ← ApplyDynamicOperators(Population)
10     Roulette.AssignUniformDistribution()
11    end if
    Step 2.2 Reproduction
12    for j = 1 to PopulationSize do
13      Parents ← TournamentSelection(Population)
14      OperatorToUse ← Roulette.ChooseOperator()
15      IncreaseOperatorUsage(OperatorToUse)
16      OffspringSolution ← ApplyOperator(OperatorToUse, Parents)
17      Evaluate(OffspringSolution)
18      OffspringPopulation ← AddSolution(OffspringSolution)
19    end for
    Step 2.3 Population and Roulette Update
20    NewPopulation ← Merge(Population, OffspringPopulation)
21    RankingAndCrowdingSort(NewPopulation)
22    Population ← RemoveSolutions(NewPopulation)
23    Roulette.UpdateRoulette(Population, actualGen, OperatorUsage)
24    OldSentryValues ← Evaluate(SentryPopulation)
25  end for
Step 3 Output
26  return Population

```

Step 2.2 Reproduction. The process of building each of the new solutions begins by selecting their parents from the population through a tournament (line 13). Subsequently, the operator to be used is randomly selected, using the distribution of percentages regulated by the roulette (line 14) and it must be evaluated (line 16) before it can be placed into the offspring population (line 17). The operators used in the algorithm are enlisted in Table 1.

When the roulette selects an operator, the number of times it has been used is increased (line 15). This process continues until the offspring population is filled.

Step 2.3 Population and roulette update. The offspring population and the current best population will be merged into a new population of size $2n$ elements. To reduce the number of elements back to a size of n elements, the new population is sorted using the criteria mentioned in the description of step 2.

Table 1 Operators used by the algorithm

<p>SBX crossover Algorithm: NSGA-II [5]</p> $s_1 = \frac{1}{2}(p_1 + p_2) - \frac{1}{2}\beta(p_2 - p_1)$ $s_2 = \frac{1}{2}(p_1 + p_2) + \frac{1}{2}\beta(p_2 - p_1)$	<p>Polynomial mutation Algorithm: NSGA-II [5]</p> $p' = \begin{cases} p + \overline{\delta_L}(p - x_i^{(L)}), & \text{for } u \leq 0.5 \\ p + \overline{\delta_R}(x_i^{(U)} - p), & \text{for } u > 0.5 \end{cases}$
<p>DE crossover: rand/2/bin Algorithm: Differential Evolution [6]</p> $s = p_e + f(p_a + p_b - p_c - p_d)$	<p>Uniform mutation Algorithm: OMOPSO [7]</p> $x_i = \begin{cases} x_i, & \text{for } u \leq 0.5 \\ x'_i, & \text{for } u > 0.5 \end{cases} \quad x'_i \in [L_i, U_i]$

Section 3.2 includes an in-depth description of the roulette update scheme and the pseudo-algorithm of the function used by the proposed algorithm to update each operator percentage (line 23 of the proposed algorithm).

3.2 Roulette Update Scheme

Algorithm 2 includes the pseudo-code of the proposed update scheme of the self-adapting roulette. The update scheme has three main features: (1) Calculate each operator efficiency rate, (2) Verify the stagnation of the roulette distribution, and (3) Modify the operator usage percentage.

In the current generation, the operator efficiency rate is the ratio between the times that an operator was used to build a solution with Pareto ranking zero, and the total times that the operator was used (Eq. 3).

$$Efficiency\ Rate = \frac{Times\ Operator\ Used\ Correctly}{Total\ Operator\ Usage} \% \tag{3}$$

Step 1 Calculate operator efficiency ratio. An operator is considered to be used correctly if and only if the following two characteristics are met: (1) The solution belongs to the Pareto zero ranking, and (2) The solution was produced during the current generation. Lines 1 through 6 describes the process that records the number of times that each operator was used correctly at any given generation.

Step 2 Update roulette distribution. The mechanism for updating the roulette includes a restart function, in case there is a stagnation in the efficiency of the operators to generate solutions that are incorporated into the best Pareto fronts. If during k -generations, zero-front solutions are not incorporated, the distribution of the probabilities of the operators will be uniformly reassigned (lines 10 and 11).

If the operator does not identify a stagnation, the mechanism proceeds to reward and punish the operators with the best and worst efficiency rate (Fig. 1), respectively

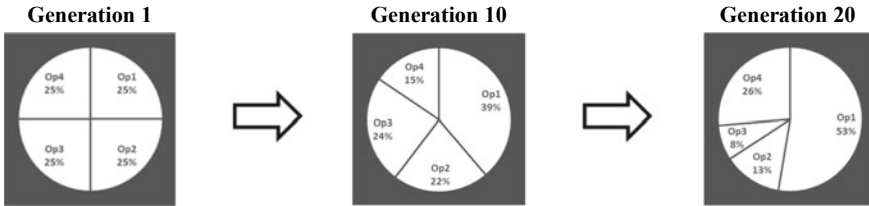


Fig. 1 Roulette update example

(lines 12 through 26).

To ensure that no operator obtains the entire distribution of the roulette or that no operator completely disappears, two boundary variables were included, *Max Percentage* and *MinPercentage*.

Step 2.1 Reward operator. After calculating the efficiency rate of all operators in each generation and sorting them from best to worst performance (line 13), the update mechanism will seek to reward the operator that obtained the best performance (lines 14 through 19) by increasing its percentage using an additive parameter ϕ .

To prevent an operator from monopolizing the entire roulette, before increasing its percentage it is verified that the result is not greater than the maximum percentage allowed (line 15), in case it exceeds the maximum limit the reward will be assigned to the operator with the next best performance, this process is performed iteratively until finding an operator whose increase is feasible.

Step 2.2 Punish operator. The process of punishing operators works in a way equivalent to step 2.1, but instead of increasing the probability of selecting an operator in roulette, the operator with the worst performance is punished by reducing his probability of selection by subtracting ϕ from his percentage of selection.

To ensure that no operator is eliminated in its entirety from the roulette, in case the result of subtracting ϕ from its current selection percentage is lower than the minimum percentage allowed, the punishment will be assigned to the operator with the next worst performance, the process will continue iteratively until finding an operator whose decrease is feasible.

Algorithm 2 Update Roulette Scheme

Input: Population, actual generation and an array with each operator usage count.

Function: Calculate the efficiency rate of each of the operators, check for stagnation and update the distribution of the roulette.

Function UpdateRoulette(Population, actualGen, OperatorUsage)

Step 1 Calculate Operator Efficiency Ratio

```

1  for i = 1 to PopulationCount do
2    if Population[i].BuiltInGen == actualGen &&
      Population[i].SolutionFront == 0 then
3      OperatorUsedCorrectly ← GetOperatorUsed(Population[i])
4      IncreaseOperatorEfficiency(OperatorUsed)
5    end if
6  end for
7  for i = 1 to OperatorCount do
8    effRate[i] ← OperatorUsedCorrectly[i] / OperatorUsage[i]
9  end for

```

Step 2 Update Roulette Distribution

```

10 if VerifyStagnation() == true do
11   AssignUniformDistribution()
12 else
13   SortOperatorsByEfficiency()
14   Step 2.1 Reward operator
15   for i = 1 to OperatorCount do
16     if Operator[i].Percentage +  $\phi$  < MaxPercentage do
17       Operator[i].Percentage ← Operator[i].Percentage +  $\phi$ 
18       break
19     end if
20   end for
21   Step 2.2 Punish operator
22   for i = OperatorCount to 1 do
23     if Operator[i].Percentage -  $\phi$  > MinPercentage do
24       Operator[i].Percentage ← Operator[i].Percentage -  $\phi$ 
25       break
26     end if
27   end for

```

4 Algorithms in the Comparison

4.1 DNSGA-II

Dynamic Non dominated Sorting Genetic Algorithm II (DNSGA-II) proposed in [5] is a modification made to the NSGA-II algorithm to handle dynamic problems, seeking to guide the search process through operators that promote diversity. In each generation, a small set of randomly selected solutions is re-evaluated. This reevaluation itrys to determine changes in the objective function or the constraints used in

the previous generations. When a change is detected, the algorithm re-evaluates the rest of the population. Two versions of the proposed DNSGA-II were suggested:

- **DNSGA-II-A:** To maintain diversity the algorithm replaces $\zeta\%$ of the population with new randomly created solutions.
- **DNSGA-II-B:** To maintain diversity the algorithm replaces $\zeta\%$ of the population with mutated solutions from the original population.

Experimental results show that the first version (with replacement using new random solutions) outperforms on DMOPs where its definition changes drastically over time, while the second version (with replacement using mutated solutions) performs well on problems that have a smooth change in the problem definition.

4.2 Differential Evolution

The Differential Evolution algorithm [6] is an instance of an evolutionary algorithm, that involves maintaining a population subjected to an iterative process of recombination, evaluation, and selection of solutions.

The recombination process builds a new mutated solution calculating the weighted difference between two or more randomly selected solutions from the population by adding the difference to another solution member. Equation 4 shows the most popular variation of the differential evolution mutation operator *rand/1/bin*, that operator constructs each element of the mutated solution x_m by recombining the elements of three different solutions (x_{s1}, x_{s2}, x_{s3}) .

$$x_m = x_{s1} + f(x_{s2} - x_{s3}) \quad (4)$$

After the mutated solution x_m has been built, it is combined with an index solution x_i of the population using elements of both solutions randomly selected by a crossover parameter *CR* to build a trail solution x_t (Eq. 5).

$$x_t = \begin{cases} x_m & \text{if } rand \leq CR \\ x_i & \text{if } rand > CR \end{cases} \quad (5)$$

The trial solution is compared against the index solution and the algorithm proceeds to replace the index solution for the solution that has the best fitness value of both.

5 Experimental Setup

The experiment goal was to compare the proposed algorithm performance regards to the performance of three algorithms of the state of the art. To measure the performance of each algorithm the epsilon, hypervolume, inverted generational distance and overall spread metrics were used. A set of twelve dynamic multi-objective optimization problems were solved independently thirty times with each algorithm. For each metric, the median and interquartile range were determined.

Finally, two statistical hypothesis tests were conducted to determine the significance of the observed differences. Table 2 contains a brief description of each algorithm and the parameters used. Table 3 shows the DMOP set choose for the experiment. Table 4 describes each performance metric.

Table 2 Algorithm description and parameters

DNSGA-II-A. Genetic Algorithm. <i>Operators:</i> Sentry population (5 distributed-solutions), Replacement method (random replacement), SBX crossover (Probability = 0.9 and distribution index = 5.0) and Polynomial mutation (probability = 1/population Size and distribution index = 20.0)
DNSGA-II-B. Genetic Algorithm. <i>Operators:</i> Sentry population (5 distributed-solutions), Replacement method (replacement using mutation), SBX crossover (Probability = 0.9 and distribution index = 5.0) and Polynomial mutation (probability = 1/population Size and index = 20.0)
Differential evolution. Genetic Algorithm. <i>Operator:</i> Sentry population (5 distributed-solutions), Replacement method (replacement using mutation), variant (rand/2/bin, CR = 0.5, f = 0.9)
Proposed algorithm. Genetic Algorithm. <i>Operators:</i> Sentry population (5 distributed-solutions), Replacement method (replacement using mutation), SBX crossover (Probability = 0.9 and distribution index = 5.0), Differential Evolution Crossover variant (rand/2/bin, CR = 0.5, f = 0.9), Polynomial mutation (probability = 1/population Size and distribution index = 20.0) and uniform mutation (probability = 1/population Size and perturbation = 0.5)

Table 3 Dynamic problems used on the comparison

FDA1 [1]	FDA5 [1]	DCTP2 [8]
FDA2 [1]	dMOP1 [9]	DCTP3 [8]
FDA3 [1]	dMOP2 [9]	DCTP4 [8]
FDA4 [1]	ZJZF8 [10]	DCTP5 [8]

Two objectives Unconstrained DMOP: FDA1, FDA2, FDA3, dMOP1 and dMOP2

Two objectives Constrained DMOP: DCTP2, DCTP3, DCTP4, and DCTP5

Three objectives DMOP: FDA4, FDA5, and ZJZF8

Table 4 Performance Measures

Epsilon (ϵ^+) [10]: The ϵ -metric is used to compare approximation sets. It measures the factor by which an approximation set is worse than another for to all objectives
Hypervolume (HV) [11]: The hypervolume or S-metric (first referred to as the size of the space covered) measures how much of the objective space is dominated by a non-dominated set
Inverted generalized distance (IGD) [12]: Measures the convergence of the approximated set towards the true <i>Pareto Optimal Front</i> (POF). The IGD measures the gap between the optimal PF and the optimized one
Overall spread (OS) [13]: Measures the extent of the front covered by the Pareto front approximation but it does not take into account the distribution of points along the Pareto front

6 Results

Tables 5, 6, 7 and 8 shows the median \bar{x} and interquartile range *IQR*, as measures of central tendency and statistical dispersion, of the four considered performance measures for all the evaluated DMOPs. In the tables, the algorithm with the best performance is highlighted using a dark gray background while the algorithm with second best performance has a light gray background.

Table 5 Median and IQR of the 4 algorithms over 30 independent runs in terms of ϵ^+

Problem	DNSGA-II-A	DNSGA-II-B	Dif evol	Adaptable roulette
FDA1—Gen 1000	8.79E-01 _{3.8E-02} ↑	1.94E-02 _{1.3E-03} ↑	1.72E-02 _{7.6E-03} ↑	8.24E-03 _{7.6E-03}
FDA2—Gen 1000	9.91E-01 _{4.5E-03} ↑	2.20E-01 _{4.5E-02} ↑	2.24E-01 _{4.2E-02} ↑	6.11E-02 _{2.6E-02}
FDA3—Gen 1000	1.05E-00 _{8.5E-02} ↑	1.89E-01 _{3.2E-01} ↓	1.87E-01 _{1.4E-02} ↓	4.32E-01 _{7.1E-04}
DCTP2—Gen 400	1.05E-00 _{1.0E-02} ↑	3.63E-02 _{1.2E-02} ↑	4.24E-02 _{1.7E-02} ↑	2.57E-02 _{7.1E-04}
DCTP3—Gen 400	1.00E-00 _{1.7E-02} ↑	1.14E-01 _{2.4E-02} ↑	1.17E-01 _{3.5E-02} ↑	9.52E-02 _{7.6E-03}
DCTP4—Gen 400	5.94E-01 _{5.3E-01} ↑	8.51E-01 _{6.9E-01} ↑	2.22E-01 _{9.4E-02} ↑	1.67E-01 _{6.7E-02}
DCTP5—Gen 400	1.00E-00 _{7.1E-01} ↑	8.84E-02 _{9.1E-01} ↑	6.37E-02 _{2.8E-02} ↑	5.97E-02 _{1.2E-02}
dMOP1—Gen 400	9.23E-03 _{2.0E-01} ↑	6.65E-03 _{1.7E-03} ↑	6.52E-03 _{7.6E-04} ↑	5.87E-03 _{4.5E-04}
dMOP2—Gen 400	9.98E-01 _{1.0E-01} ↑	1.10E-01 _{1.8E-01} ↑	7.67E-02 _{7.2E-02} ↓	8.19E-02 _{7.6E-02}
FDA4—Gen 1000	1.08E-00 _{6.6E-02} ↑	1.14E-01 _{1.5E-02} ↓	1.02E-01 _{8.2E-03} ↓	1.23E-01 _{2.2E-02}
FDA5—Gen 1000	2.27E-01 _{7.5E-05} ↑	1.39E-01 _{4.5E-02} —	1.03E-01 _{6.0E-02} ↓	2.21E-01 _{4.1E-02}
ZJZF8—Gen 1000	7.37E-01 _{1.4E-01} ↑	1.69E-01 _{3.5E-02} ↑	1.78E-01 _{5.5E-02} ↑	1.48E-01 _{3.1E-02}

Table 6 Median and IQR of the 4 algorithms over 30 independent runs in terms of *HV*

Problem	DNSGA-II-A	DNSGA-II-B	Dif evol	Adaptable roulette
FDA1—Gen 1000	9.78E-02 _{3.4E-02} ↑	6.74E-01 _{3.7E-03} ↑	6.55E-01 _{3.8E-03} ↑	6.60E-01 _{4.1E-04}
FDA2—Gen 1000	3.98E-03 _{4.5E-03} ↑	1.92E-01 _{2.1E-03} ↑	1.93E-01 _{2.2E-03} ↑	2.02E-01 _{2.6E-04}
FDA3—Gen 1000	3.74E-02 _{7.8E-02} ↑	6.27E-01 _{5.2E-03} ↓	6.26E-01 _{1.4E-02} ↑	5.22E-01 _{6.7E-04}
DCTP2—Gen 400	4.32E-01 _{1.0E-01} ↑	4.63E-01 _{4.4E-03} ↑	4.61E-01 _{4.7E-03} ↑	4.63E-01 _{1.5E-03}
DCTP3—Gen 400	1.56E-01 _{1.7E-01} ↑	4.72E-01 _{2.2E-02} ↑	4.72E-01 _{3.0E-02} ↑	4.81E-01 _{1.1E-02}
DCTP4—Gen 400	3.12E-01 _{2.4E-01} ↑	3.96E-01 _{2.9E-02} —	1.45E-01 _{2.3E-01} ↑	3.75E-01 _{4.9E-02}
DCTP5—Gen 400	3.76E-01 _{2.4E-02} ↑	4.19E-01 _{5.7E-03} —	3.98E-01 _{4.0E-01} ↑	4.06E-01 _{9.3E-03}
dMOP1—Gen 400	4.17E-01 _{1.6E-02} ↑	4.37E-01 _{1.7E-02} ↑	4.27E-01 _{5.1E-05} ↑	4.87E-01 _{1.4E-05}
dMOP2—Gen 400	3.81E-02 _{7.0E-02} ↑	3.74E-01 _{8.7E-02} ↓	3.41E-01 _{1.0E-01} ↑	3.44E-01 _{3.9E-02}
FDA4—Gen 1000	0.00E-00 _{0.0E-00} ↑	3.99E-01 _{8.9E-03} —	3.96E-01 _{7.5E-03} ↑	3.99E-01 _{9.6E-03}
FDA5—Gen 1000	1.00E-00 _{3.6E-04} ↓	7.93E-01 _{2.2E-02} ↑	7.92E-01 _{2.7E-02} ↑	9.33E-01 _{4.8E-02}
ZJZF8—Gen 1000	1.83E-02 _{4.8E-03} ↑	2.75E-03 _{1.5E-04} ↓	4.08E-01 _{2.3E-02} —	4.01E-01 _{1.8E-02}

To assess the statistical confidence in every pairwise comparison between the proposed algorithm and the three-classic state of the art algorithms, the Wilcoxon signed-rank test was applied at 95% confidence level.

In the tables, the symbol ↑ is used when the Adaptable Roulette Algorithm is significantly better than the corresponding algorithm, for the opposite the symbol ↓ is used. When the performance difference is not statistically significant the symbol — is used. Finally, the Friedman statistical test was applied to analyze the overall performance of the algorithm across all considered problems.

Table 9 shows the algorithms according to their Friedman Ranking for each of the performance metrics. In each column, algorithms are ordered from the best to the worst performance.

7 Conclusions and Future Work

In this work, we have proposed a new multi-objective evolutionary algorithm, called Dynamic multi-objective evolutionary algorithm with an adaptable roulette for the

Table 7 Median and IQR of the 4 algorithms over 30 independent runs in terms of *IGD*

Problem	DNSGA-II-A	DNSGA-II-B	Dif evol	Adaptable roulette
FDA1—Gen 1000	1.60E-02 _{1,0E-03} ↑	2.40E-04 _{6,8E-05} ↑	2.27E-04 _{7,6E-05} ↑	1.46E-04 _{7,6E-06}
FDA2—Gen 1000	1.08E-02 _{5,8E-04} ↑	5.17E-04 _{1,1E-04} ↑	5.19E-04 _{1,7E-04} ↑	1.65E-04 _{1,9E-05}
FDA3—Gen 1000	1.45E-02 _{2,8E-03} ↑	1.19E-03 _{1,9E-04} ↓	1.11E-03 _{3,4E-03} ↑	4.22E-03 _{4,1E-06}
DCTP2—Gen 400	3.86E-02 _{5,6E-04} ↑	5.91E-04 _{1,2E-04} ↑	5.79E-02 _{1,7E-02} ↑	4.67E-04 _{6,7E-05}
DCTP3—Gen 400	2.24E-01 _{2,8E-02} ↑	1.92E-02 _{5,4E-03} ↑	2.05E-02 _{4,1E-03} ↑	1.69E-02 _{2,5E-03}
DCTP4—Gen 400	1.43E-01 _{2,1E-01} ↑	2.99E-02 _{6,9E-03} ↑	1.56E-01 _{1,3E-01} ↑	2.59E-02 _{2,7E-03}
DCTP—Gen 400	3.00E-02 _{1,6E-03} ↑	1.97E-03 _{6,1E-04} ↑	2.38E-03 _{2,8E-02} ↑	1.59E-03 _{2,2E-04}
dMOP1—Gen 400	1.35E-04 _{2,1E-06} -	1.36E-04 _{1,7E-06} ↑	1.36E-04 _{2,7E-06} ↑	1.34E-04 _{7,5E-07}
dMOP2—Gen 400	5.91E-02 _{1,8E-01} ↑	2.23E-03 _{5,0E-03} ↓	4.38E-03 _{6,9E-03} ↑	4.19E-03 _{2,3E-03}
FDA4—Gen 1000	4.15E-02 _{1,6E-03} ↑	2.22E-03 _{1,2E-04} ↓	2.28E-03 _{6,4E-05} ↑	2.25E-03 _{5,2E-05}
FDA5—Gen 1000	2.46E-02 _{3,6E-03} ↑	1.43E-02 _{2,2E-03} ↓	1.46E-02 _{1,2E-03} -	1.50E-02 _{1,3E-03}
ZIZF8—Gen 1000	1.83E-02 _{4,8E-03} ↑	2.75E-03 _{1,5E-04} ↓	2.88E-03 _{4,5E-04} -	2.87E-03 _{3,8E-04}

Table 8 Median and IQR of the 4 algorithms over 30 independent runs in terms of *OS*

Problem	DNSGA-II-A	DNSGA-II-B	Dif evol	Adaptable roulette
FDA1—Gen 1000	1.08E-01 _{6,2E-02} ↑	9.98E-01 _{3,7E-03} ↑	9.97E-01 _{7,5E-03} ↑	1.00E-00 _{4,7E-04}
FDA2—Gen 1000	7.62E-03 _{2,5E-03} ↑	7.82E-01 _{4,2E-02} ↑	7.95E-01 _{4,5E-02} ↑	9.48E-01 _{2,6E-02}
FDA3—Gen 1000	7.20E-04 _{1,2E-02} ↑	5.43E-01 _{4,1E-02} ↑	5.44E-01 _{7,7E-01} ↓	1.93E-01 _{2,4E-04}
DCTP2—Gen 400	1.13E-00 _{1,2E-04} ↑	1.00E-00 _{4,1E-02} —	1.02E-00 _{2,1E-02} —	1.01E-00 _{7,5E-03}
DCTP3—Gen 400	9.11E-04 _{2,1E-03} ↑	9.91E-01 _{1,6E-01} ↑	9.93E-01 _{7,6E-03} ↑	9.96E-01 _{4,3E-03}
DCTP4—Gen 400	2.48E-02 _{1,4E-01} ↑	8.06E-01 _{2,4E-01} ↑	2.39E-02 _{1,0E-00} ↑	1.00E-00 _{1,6E-01}
DCTP5—Gen 400	0.00E-00 _{0,0E-00} ↑	9.98E-01 _{8,3E-02} ↑	9.13E-01 _{9,2E-01} ↑	1.01E-00 _{1,2E-02}
dMOP1—Gen 400	9.92E-01 _{3,2E-01} ↑	1.00E-00 _{0,0E-00} —	1.00E-00 _{0,0E-00} ↑	1.00E-00 _{0,0E-00}
dMOP2—Gen 400	1.09E-02 _{3,1E-02} ↑	9.78E-01 _{3,4E-02} ↑	7.51E-01 _{2,8E-01} ↑	9.90E-01 _{3,9E-02}
FDA4—Gen 1000	1.28E-00 _{4,9E-01} ↓	9.74E-01 _{2,2E-02} ↑	2.33E-00 _{8,8E-01} ↑	9.99E-01 _{4,8E-02}
FDA5—Gen 1000	1.94E-05 _{5,5E-05} ↑	5.08E-02 _{3,1E-02} ↑	5.96E-02 _{2,9E-02} ↓	3.43E-02 _{1,5E-02}
ZJZF8—Gen 1000	7.78E-00 _{5,5E-00} ↓	1.53E-00 _{3,9E-01} ↑	1.72E-00 _{4,1E-01} ↑	1.95E-00 _{2,6E-01}

Table 9 Friedman’s ranking comparison

<i>e</i> ⁺	HV		IGD		OS		
Adaptable roulette	1.967	Adaptable roulette	1.787	Adaptable roulette	1.627	Adaptable roulette	2.051
DNSGA-II-B	2.570	DNSGA-II-B	2.430	Dif evol	1.914	Dif evol	2.697
Dif evol	2.913	Dif evol	2.946	DNSGA-II-B	2.719	DNSGA-II-B	2.872
DNSGA-II-A	3.411	DNSGA-II-A	3.837	DNSGA-II-A	3.897	DNSGA-II-A	3.462

selection of operators. The proposed algorithm contains different operators, commonly used independently by the other algorithms, but applying them following the probability defined by a self-adapting roulette, that changes according to the effective rate of each operator, rewarding the operators with good performance while diminishing the impact of the operators that do not contribute with good solutions.

For the epsilon, hypervolume, inverted generalized distance and overall spread metrics, the proposed algorithm outperforms with a significance of 95% to the other algorithms in a 66.66%, 42%, 60% and, 60% respectively.

Experimental results also show that crossover operators are a good starting point for the optimization process but are outperformance by the mutation operators on

the late generations. In future works, we plan to evaluate each operator performance and experiment with other operator set using another operator-selection scheme.

References

1. Farina, M., Deb, K., Amato, P.: Dynamic multiobjective optimization problems: test cases, approximations, and applications. *IEEE Trans. Evol. Comput.* **8**(5), 425–442 (2004)
2. Greeff, M., Engelbrecht, A.P.: Solving dynamic multi-objective problems with vector evaluated particle swarm optimisation. In: 2008 IEEE Congress on Evolutionary Computation, CEC (IEEE World Congress on Computational Intelligence), pp. 2917–2924 (2008)
3. Helbig, M., Engelbrecht, A.P.: Population-based metaheuristics for continuous boundary-constrained dynamic multi-objective optimisation problems. *Swarm Evol. Comput.* **14**, 31–47 (2014)
4. Azzouz, R., Bechikh, S., Said, L.B.: A dynamic multi-objective evolutionary algorithm using a change severity-based adaptive population management strategy. *Soft Comput.* 1–22 (2015)
5. Deb, K., Rao, N.U.B., Karthik, S.: Dynamic multi-objective optimization and decision-making using modified NSGA-II: a case study on hydro-thermal power scheduling. In: International Conference on Evolutionary Multi-Criterion Optimization, pp. 803–817 (2007)
6. Storn, R., Price, K.: Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *J. Glob. Optim.* **11**(4), 341–359 (1997)
7. Sierra, M.R., Coello, C.A.C.: Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance. In: International Conference on Evolutionary Multi-criterion Optimization, pp. 505–519 (2005)
8. Azzouz, R., Bechikh, S., Said, L.B.: Multi-objective optimization with dynamic constraints and objectives: new challenges for evolutionary algorithms. In: Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation, pp. 615–622 (2015)
9. Goh, C.K., Tan, K.C.: A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *IEEE Trans. Evol. Comput.* **13**(1), 103–127 (2009)
10. Li, H., Zhang, Q.: A multiobjective differential evolution based on decomposition for multiobjective optimization with variable linkages. In: Parallel Problem Solving from Nature-PPSN IX, pp. 583–592. Springer, Berlin (2006)
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 182–197 (2002)
12. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., Da Fonseca, V.G.: Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans. Evol. Comput.* **7**(2), 117–132 (2003)
13. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE Trans. Evol. Comput.* **3**(4), 257–271 (1999)

Combinatorial Designs on Constraint Satisfaction Problem (VRP)



Juan A. Montesino-Guerra, Héctor Puga, J. Martín Carpio,
Manuel Ornelas-Rodríguez, A. Rojas-Domínguez and Lucero Ortiz-Aguilar

Abstract The constraint satisfaction problems (CSP) often show great complexity and require a combination of heuristic methods and combinatorial search to be solved in a reasonable time. Therefore, they are of particular importance in the area of intelligent systems. A proposal of a methodology for solving CSP problems is presented, in which the characteristics of combinatorial designs based on algebraic structures, such as Mutually Orthogonal Latin Squares, are exploited in the search for solutions (answers) to a CSP problem. The proposal and the set of heuristics associated with the combinatorial design are evaluated, looking for the pair of heuristics with the best performance in the set of artificial instances of the vehicle routing problem (VRP). The results show the usefulness of the combinatorial designs to find solutions that resolve artificial instances and support the feasibility to extend its application on instances of the state-of-the-art and later on different problem domains.

Keywords Constraint satisfaction problem · CSP · Combinatorial designs · Latin squares · Mutually orthogonal latin squares · MOLS · Metaheuristics · Iterated local search

J. A. Montesino-Guerra · H. Puga · J. M. Carpio (✉) · M. Ornelas-Rodríguez ·
A. Rojas-Domínguez · L. Ortiz-Aguilar
División de Estudios de Posgrado e Investigación, Tecnológico Nacional de México, Instituto
Tecnológico de León, León, Guanajuato, Mexico
e-mail: jmcarpio61@hotmail.com

J. A. Montesino-Guerra
e-mail: adolfo.montesino@hotmail.com

H. Puga
e-mail: pugahector@yahoo.com

M. Ornelas-Rodríguez
e-mail: mornelas67@yahoo.com.mx

A. Rojas-Domínguez
e-mail: alfonso.rojas@gmail.com

L. Ortiz-Aguilar
e-mail: ldm_oa@hotmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_36

1 Introduction

The theory of combinatorial designs has been used extensively in very different areas of computing. The combinatorial designs have properties that are inherent to many difficult problems and therefore can be useful in addressing these problems. Combinatorial designs and studies on Combinatorics intersect in a natural way, for example, in the case of finite spaces, such as graphics and the nature of their manipulation [1, 2].

Combinatorial design theory is based on questions about whether it is possible to organize elements of a finite set in subsets, so that certain properties are satisfied in a “balanced” way. There are many and varied types of designs; as an example, one can find the designs of incomplete blocks, t -designs, designs balanced by pairs, and those that are of interest for this investigation, orthogonal Latin squares [3].

A Latin square of order n is a matrix of order $n \times n$ whose terms are elements of any set S of size n , so that each row and each column contain all the elements of S . These elements cannot be repeated more than once at each column and a given row [5, 6]. In Latin squares we can notice many interesting areas of research, among them, the Latin Sub-squares and the Mutually Orthogonal Latin Squares (MOLS), just to name a few. Latin squares also have a variety of different practical applications, for example, they can be used to encode messages, design tournaments or formation of statistical designs and construct error correction codes [4, 5].

In the domain of Constraint Satisfaction Problems (CSP), these have an innate connection between the structure of the problem and the difficulty in solving it. This structure is often a product of the representation of the design of this. It is important the design that represents the problem, even if it only represents an instance of the problem, since this is essential for the solution of the problem itself.

Combinatorial problems with restrictions arise in many areas of Computer Science, Artificial Intelligence and Operations Research, where simplified models are often used to formalize real-life problems, involving finding a grouping, arrangement or assignment of a discrete and finite set of objects that satisfy the given conditions. It is therefore of interest to design a method/model that solves the problem as best as possible.

This paper presents a method to apply combinatorial designs, specifically Latin squares, to a CSP, namely the Vehicle Routing Problem (VRP). In the proposed methodology, combinatorial structures (MOLS) are designed for different artificial instances of the VRP, with different levels of complexity, to analyze through their evolution the performance of these structures in the search for solutions.

2 Concepts

In this Section, we focus to concepts about Constraint satisfaction problem, Combinatorial designs, Latin squares, Vehicle routing problem and the metaheuristic that we will use in this work.

2.1 Constraint Satisfaction Problems

The satisfaction restriction problem has an approach with which it is possible to pose, in a natural way, combinatorial problems that are of importance in artificial intelligence and other research areas. The objective of a constraint satisfaction problem is to find an assignment of values to a given set of variables subject to constraints; these values can be assigned simultaneously to certain specific subsets of variables [6].

It is known that the problem of the satisfaction of restrictions is in general an NP-complete problem [7]. An instance of a *constraint satisfaction problem* consists of [6].

- a finite set of variables, V ;
- a finite domain of values, D ;
- a finite set of constraints $\{C_1, C_2, \dots, C_q\}$;
each constraint C_i is a pair (s_i, R_i) , where:
 - s_i is a tuple of variables of length m_i , called the ‘constraint scope’; and
 - R_i is an m_i -ary relation over D , called the ‘constraint relation’.

“For each constraint, (s_i, R_i) , the tuples in R_i indicate the allowed combinations of simultaneous values for the variables in s_i . The length of s_i , and of the tuples in R_i , is called the ‘arity’ of the constraint. In particular, unary constraints specify the allowed values for a single variable, and binary constraints specify the allowed combinations of values for a pair of variables” [6].

2.2 Combinatorial Design

The theory of combinatorial design raises questions as if it is possible to organize elements of a finite set in subsets so that certain properties of “equilibrium” are satisfied (in our context this will serve as constraints). The types of designs that exist, to mention a few examples, include balanced designs of incomplete blocks, t-designs, designs balanced by pairs, orthogonal Latin squares and many more. Questions such as “Is there a design of a specific type?” are the most fundamental in this area of research. Current design theory includes existence results, as well as non-existence results of such designs. However, there are still open problems related to the existence of certain types of designs [3].

Design theory has tools for analysis such as linear algebra, groups, rings as well as fields, and number theory, in addition to combinatorics. The basic concepts of design theory are simple, but the mathematics used to study designs are diverse, robust and ingenious [3].

2.2.1 Definition

The combinatorial designs systematically study the selection of objects according to specific rules, or, in an equivalent way, the incidence relations between certain objects and certain subsets of these objects [8].

A combinatorial design is [8]:

$$D = (V, B)$$

where:

$V = \{x_1, x_2, \dots, x_n\}$, called set of *variety elements*,

$B = \{b_1, b_2, \dots | b_i \cdot V\}$, are subsets of the varieties called *design blocks*.

The number of varieties of a design D is the product of $|V||B|$ where by $v = |V|$ and the $b = |B|$ [8].

A design is, then, a general system of incidence that tells us when an element (variety) $x_i \in V$, is in a certain subset (block) $b_i \in B$. A useful way to represent a combinatorial design is from what is called its incidence matrix [8].

$$A = (a_{ij})_{v \times b}, \quad a_{ij} = \begin{cases} 1, & \text{if } x_i \in B_j \\ 0, & \text{if } x_i \notin B_j \end{cases}$$

For example:

$$A = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

is the incidence matrix of a design $D = (V, B)$, with set of varieties $V = \{1, 2, 3, 4, 5\}$ and set of blocks: $B = \{\{4\}, \{4\}, \{2, 4\}, \{2, 3, 4\}\}$

It is important to note that the possibility that B has repeated blocks is admitted. When this is not the case, it is said that the design is simple. It is said that a design $D = (V, B)$ can have uniformity and regularity as a characteristic to be able to create a particular combinatorial design [8].

2.2.2 Latin Squares

A Latin square is a regular combinatorial design of order n is a matrix of order $n \times n$ whose terms are elements of any set S of size n , so that each row and each column contain all the elements of S , for $S = \{1, 2, 3\}$ [8].

For example, the matrix:

```

1 2 3
2 3 1
3 1 2
    
```

It is a Latin square of order 3.

Each row and each column of a Latin square is a permutation of the elements of S . It is also clear that a Latin square is a complete symmetric design with n repeated blocks, each equal to S [8].

Two Latin squares of size n are equivalent if it is possible to generate one of the other by means of a permutation of the symbols. For example, each pair of squares of size n shown below are equivalent [8]:

```

1 2      2 1
2 1      1 2
1 2 3    3 2 1
2 3 1    2 1 3
3 1 2    1 3 2
    
```

Two Latin squares $A = (a_{ij})$ and $B = (b_{ij})$ of size n are *orthogonal* if the n^2 ordered pairs $(a_{ij}, b_{ij}) \in A \times B$ are all different [8].

2.2.3 CSP that Can Be Modeled as Combinatorial Designs

On the algebraic structure of combinatorial problems [6] *The general combinatorial problem* (GCP) is a decision problem with Instance:

A pair of similar finite relational structures, (S_1, S_2) .

Question: Is there a homomorphism from S_1 , to S_2 ?

For any instance GCP $P = (S_1, S_2)$ a homomorphism from S_1 to S_2 will be called a solution of P .

An instance of the CSP [2, 3] consists of a set of variables V , a domain of D values, and a list of constraints $C(S_1), C(S_2), \dots, C(S_m)$, where each S_i is an ordered subset of V and the constraint $C(S_i)$ is a set of tuples specifying the combinations of allowed values for the variables in S_i . The question is, if there is an assignment of values of D for the variables in V in such a way that each restriction is satisfied [6]. This can be expressed as an instance of GCP:

$$\{\{V, S_1, S_2, \dots, S_m\}, \{D, C(S_1), C(S_2), \dots, C(S_m)\}\}$$

2.3 Vehicle Routing Problem

The vehicle routing problem (VRP) is a problem of combinatorial and integer optimization programming that asks, “What is the optimal set of routes for a fleet of vehicles to travel and deliver to a given set of customers?”. Generalizes the well-known Traveler Salesman Problem (TSP), See Fig. 1 [9].

It is known that the VRP is NP-hard because it includes the Traveling Salesman problem as a special case [10].

This is one of the most studied combinatorial optimization problems in Operations Research. To solve this problem, we must determine the order in which to visit a set of customers, each with a known demand for a product, using a fleet of vehicles. The routes of the vehicles start and end in a depot and have a capacity, which limits the total demand to cover of all the customers served in each route. The transport costs of the distances of the resulting delivery routes should be minimized. To solve instances of realistic size in a feasible time, a variety of heuristics have been successfully developed over the past decades.

3 Proposed Design Methodology

A CSP can be considered as an algebraic structure from which a combinatorial design is generated [6]. In our methodology a MOLS design is “evolved” and thus used to search the space of modeled problem solutions consisting of three phases and some preparations, where instances of the problem to be modeled are obtained. See Fig. 2.

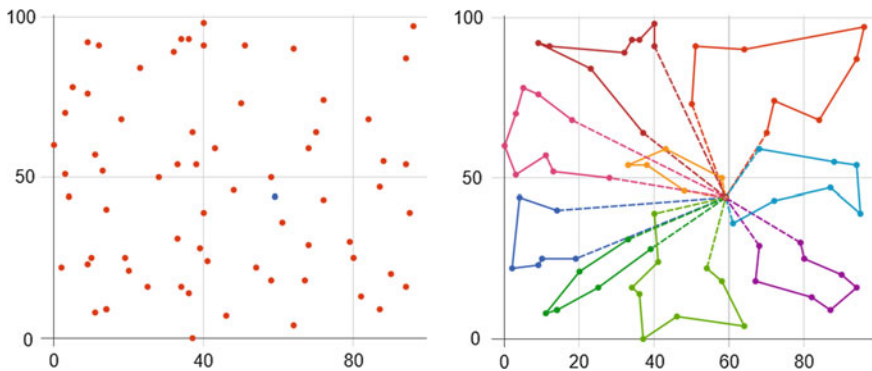


Fig. 1 Instance example with 68 delivery points and 9 trucks with a capacity of 100 (Unsolved/Solved) [11]

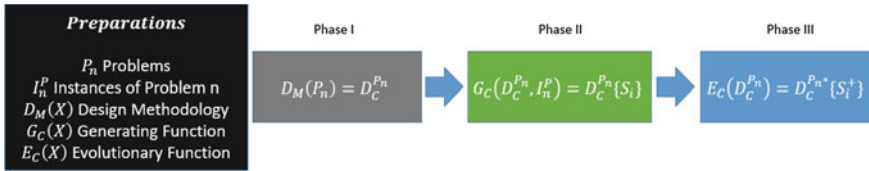


Fig. 2 Methodology of combinatorial design with three phases. $D_C^{P_n}$ is a Combinatorial Design related to a given problem, and S_i is an initial solution for that same problem

3.1 Phase I

As explained, the fundamental components of a combinatorial design is a set of *design blocks* and a set of *varieties*, remembering that these *varieties* exist within the *design blocks*.

As the first stage of the design, in the case of the VRP, the set of varieties $V = \{1, 2, 3, \dots, n\}$, where n is the number of *cities* or *delivery points*. And the *design blocks* are subsets of cities traveled by each truck, which are the *different solution routes*. In this particular case, there cannot be more than one type of *variety (delivery point)* within each *design block* (route of a truck) because of the restriction that only one time can be passed per delivery point.

An instance of VRP contains, among other data, the dimension of the problem (the number of delivery points), the coordinates per point, the demand per point, type of distance used, the node designated as a deposit. See following example of the instance of Set A of Augerat, 1995 [11] :

NAME: A-n45-k7. COMMENT: (Augerat et al., Min no of trucks: 7, Best value: 1146)
 TYPE: CVRP. DIMENSION: 45. EDGE_WEIGHT_TYPE: EUC_2D
 NODE_COORD_SECTION

1	61	99
2	95	7
...
44	7	25
45	35	35

DEPOT_SECTION

1

With this, we can generate for this instance a MOLS of 7×7 , the size is given by the restriction that the number of trucks must be equal to 7. This square is labeled with the respective numbers keeping the properties of a MOLS, where there should not be more than one variety per row and per column. In addition, a letter of the alphabet is added to the entire line on the label. With this we obtain an alpha-numeric label, with which we will identify the “box”, located in the matrix, as shown in Fig. 3.

Fig. 3 MOLS with alpha-numeric label

7B	8B	5B	6B	3B	4B	1B
6C	5C	8C	7C	2C	1C	4C
5D	6D	7D	8D	1D	2D	3D
4E	3E	2E	1E	8E	7E	6E
3F	4F	1F	2F	7F	8F	5F
2G	1G	4G	3G	6G	5G	8G
1H	2H	3H	4H	5H	6H	7H

3.2 Phase II

We proceed to “fill” each of these labeled “boxes”, so that all the nodes of the instance are assigned to a box. For this example, a sequential filling was done until all the nodes have been assigned. Other types of filling can be used, as the one described in Sect. 3.5.

In addition, a colored column of the MOLS is shown, in order to represent each of the necessary routes for this problem, we can see that the route of the first column contains the route $R = \{1, 8, 15, 22, 29, 36, 43\}$, in the same way each route (column) contains its own initial route without repeating any node within it, thus complying with the restrictions established up to this moment, having a feasible and consistent initial solution.

3.3 Phase III

We have everything ready to begin the process of finding solutions from this initial MOLS. The search process is done by manipulating the labels. For this example, we have the initial square shown in Fig. 5a.

Taking the numbers of a whole row and exchanging them for the numbers of another row, one can see how the labels change positions. Both in the horizontal and vertical order, therefore disturbing the content and order of the routes modeled by column (see Fig. 5b).

Going back to the initial MOLS example in Fig. 4, and applying the label modification, one can clearly see (Fig. 6) the effect mentioned in the previous paragraph.

It is evident that the route $R_1 = \{1, 8, 15, 22, 29, 36, 43\}$, is different to the route $R_1^* = \{7, 9, 15, 22, 29, 36, 43\}$, so will have different distances of travel than the initial ones. The same modification effect can be observed in each of the other routes (columns). It is worth mentioning that different types of modifications can be made to the labels to achieve different effects, which will be dealt with in detail in Sect. 3.4. With this type of modifications, the state search is performed to minimize the fitness function for this problem.

3.4 Heuristics K-Opt

The set of heuristics used to create a solution methodology is a fundamental part of any approach in the search for a solution proposal.

Heuristics are a systemic strategy, which immediately or gradually generates alternatives, seeking that these are positive innovations [12]. Then the heuristics are principles that contribute to the reduction of search in the problem-solving activity [13]. These oriented to the work of computation can support in the search and systemic investigation of a space of solutions.

One of these heuristics is known as k-opt whose main idea is to take two relationships or unions and reorder them within the same set that contains them [14, 15].

Fig. 4 MOLS filled with initial VRP solution

7B	1B	2B	3B	4B	5B	6B
1	2	3	4	5	6	7
6C	7C	1C	2C	3C	4C	5C
8	9	10	11	12	13	14
5D	6D	7D	1D	2D	3D	4D
15	16	17	18	19	20	21
4E	5E	6E	7E	1E	2E	3E
22	23	24	25	26	27	28
3F	4F	5F	6F	7F	1F	2F
29	30	31	32	33	34	35
2G	3G	4G	5G	6G	7G	1G
36	37	38	39	40	41	42
1H	2H	3H	4H	5H	6H	7H
43	44	45				

(a)								(b)							
7B	8B	5B	6B	3B	4B	1B		6B	5B	8B	7B	2B	1B	4B	
6C	5C	8C	7C	2C	1C	4C		7C	8C	5C	6C	3C	4C	1C	
5D	6D	7D	8D	1D	2D	3D		5D	6D	7D	8D	1D	2D	3D	
4E	3E	2E	1E	8E	7E	6E		4E	3E	2E	1E	8E	7E	6E	
3F	4F	1F	2F	7F	8F	5F		3F	4F	1F	2F	7F	8F	5F	
2G	1G	4G	3G	6G	5G	8G		2G	1G	4G	3G	6G	5G	8G	
1H	2H	3H	4H	5H	6H	7H		1H	2H	3H	4H	5H	6H	7H	

Fig. 5 Mutually orthogonal Latin Squares, a Initial, b Perturbed

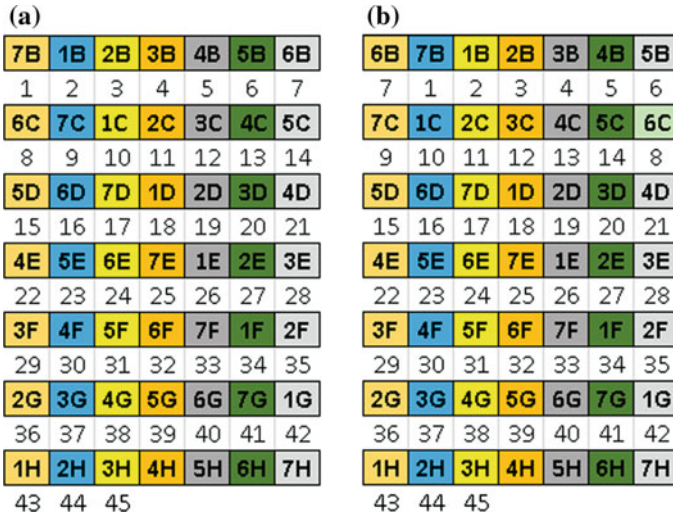


Fig. 6 Modification in the content of the MOLS by manipulating labels

3.4.1 Perturbative Heuristics Applied to Latin Squares

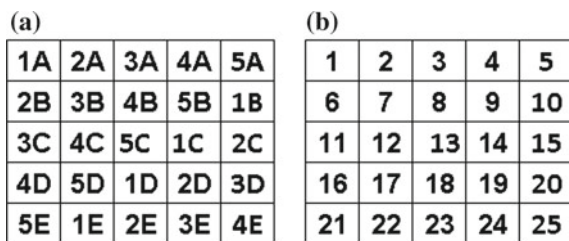
For decades, many algorithms have been developed to find the solution to different problems, finding good solutions but not necessarily optimal solutions. Not all optimization problems can be solved using exact methods, that is, it is not always possible to find the optimal solution with acceptable computational efforts in a reasonable time. So, the adaptation and proposal of measures that support the search for solutions that have simple designs are well received in the areas of optimization.

Following are some of the heuristics based on k-opt for Latin squares; only the first heuristic is explained graphically for illustrative purposes, starting from an initial MOLS (A) of 5×5 [regular 5-design] (Fig. 7).

Heuristic K-opt 1/(A). Swap the numbers of a given row with the numbers of another row; these numbers can be chosen contiguously or randomly. As shown in Fig. 8.

Heuristic K-opt 2/(B). Swap the letters of a given row, with the letters of another chosen row; these rows can be chosen contiguously or randomly.

Fig. 7 a Identifiers of A
b Content of A



1A	2A	3A	4A	5A
2B	3B	4B	5B	1B
3C	4C	5C	1C	2C
4D	5D	1D	2D	3D
5E	1E	2E	3E	4E

2A	3A	4A	5A	1A
1B	2B	3B	4B	5B
3C	4C	5C	1C	2C
4D	5D	1D	2D	3D
5E	1E	2E	3E	4E

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

2	3	4	5	1
7	8	9	10	6
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Fig. 8 Heuristic K-opt 1/(A)

Heuristic K-opt 3/(C). Exchange a given number with another chosen number; these numbers can be chosen contiguously or randomly.

Heuristic K-opt 4/(D). Exchange the numbers of a given column with the numbers of another chosen column; this can be done from right to left or from left to right. The numbers can be chosen contiguously or randomly.

Heuristic K-opt 5/(E). Exchange the letters of a given column with the letters of another column; this can be done from right to left or from left to right. The letters can be chosen contiguously or randomly.

3.5 Initial Solution

The classic idea of a greedy algorithm can be explained as follows [16]:

Define an appropriate potential function for problem $f(A)$ over the set of potential solutions A , this function is commonly known as *Fitness Function*.

Initialize $A = 0$, and increase the set of solutions A by adding, in each stage, an element that maximizes (or minimizes) the value of $f(A \cup \{x\})$, until $f(A)$ reaches the maximum value (or minimum) according to the objective of the problem. An example of the type of distance minimization is shown in Algorithm 1.

Algorithm 1 Greedy Algorithm

Require: $A = 0, S[M]$

- 1: Sort $S[M]$ in increasing order by distance $w(x)$
 - 2: **for each** $x \in S[M]$, taken in increasing order by distance $w(x)$ **do**
 - 3: **if** $A \cup \{x\} \in I[M]$
 - 4: $A \leftarrow A \cup \{x\}$
 - 5: **end if**
 - 6: **end for**
 - 7: **return** A
-

3.6 Metaheuristics

Metaheuristics, in its original definition, are methods of solution that orchestrate an interaction between local improvement procedures and higher-level strategies to create a process capable of escaping local optima and conduct a robust search over a space of solutions. Over time, these methods have also come to include all procedures that employ strategies to overcome the trap of local optimality in complex solution spaces, especially procedures that use one or more neighborhood structures as a means to define admissible movements from one solution to another, or to construct or destroy the solutions in the constructive and destructive processes [17, 18].

In this work, by *metaheuristics*, we are referring to an iterative master process that guides and modifies the operations of subordinate heuristics to efficiently produce high quality solutions. The metaheuristic used in in this work is briefly presented below.

3.6.1 Iterated Local Search

In the Iterative Local Search (ILS), the initial solution has proven to be fundamental. From this initial state iteratively constructs a sequence of related solutions by means of the integrated heuristics in its phases [17]. In summary, this algorithm has an “external” stage in which it performs a perturbation with some exploratory heuristic, and an “internal” perturbation stage in which it applies some heuristic to the solution with an intensification approach. The ILS algorithm is described in detail in [18].

Algorithm 2 Iterated Local Search (ILS)

```

1:  $S_0 = \text{GenerateInitialSolution}$ 
2:  $S^* = \text{ImprovementStage}(S_0)$ 
3: while ! $\text{StopCondition}$  do
4:    $S^{*'} = \text{SimpleRandomPerturbation}(S^*)$ 
5:    $S^{*'} = \text{ImprovementStage}(S^{*'})$ 
6:   if  $f(S^{*'}) < f(S^*)$  then
7:      $S^* = S^{*'}$ 
8:   end if
9: end while
10: return  $S^*$ 

```

4 Experiments and Results

In this section, the experiments and the result of the performance of the heuristics applied to the MOLS that models the VRP are reported. The characteristics of the instances used are also described.

Table 1 Artificial Instances

Instance	N	Noise	Optimum
7×7	49	5/10/20/40/80	Yes
8×8	64	5/10/20/40/80	Yes
9×9	81	5/10/20/40/80	Yes
10×10	100	5/10/20/40/80	Yes
15×15	225	5/10/20/40/80	Yes

4.1 Test Instances

Artificial instances of VRP, whose design methodology is explained in depth by Soria and Ortiz in [19, 20] were employed. In such instances, uniform random points without repetition are generated as the optimal solution of the instance, with which a distance matrix is constructed, leaving the rest of the intersections between pairs of points occupied by noise (non-optimal distances). Instances of 49, 64, 81, 100 and 225 points were generated, including the deposit point. Each of these instances has 5 versions for the difficulty value that is added by means of the noise (25 instances in total), where the higher this value is, the greater the difficulty of the instance. Table 1 shows the information about these instances.

4.2 Experimental Design

Given a set of tools for a problem, it is important to prove which of these is appropriate for the work required, verifying that they distinguish their effect from each other achieving desirable results. Therefore, to test this methodology with its set of proposed heuristics, it is important to test the performance of the heuristics for the set of instances of the VRP.

To test the methodology, 25 experiments were generated, one for each pair of heuristics. Each experiment was run over 25 VRP test instances with a minimization approach. For each instance, 33 runs were made to obtain the minimum path of each experiment, and then the median of the 33 experiments was obtained to use it as a representative in the nonparametric statistical tests. The 33 runs were performed with all the possible pairs of heuristics, the first element of the pair is the heuristic assigned to the exploration while the second element is the heuristic assigned to the intensification of the local search. The initial configuration of parameters for the ILS is reported in Table 2. The number of *function calls* is used as stopping criterion for this algorithm.

Although there are several pairs of heuristics that achieve good performance, this information is not enough to know the significance of the difference in performance between the different pairs that solve the instances of the problem. Thus, a non-parametric statistical test was applied.

Table 2 Configuration for iterate local search

	ILS
Exploratory crossover	1
Local search iterations	10
Stop criteria (functions Call)	100,000

Table 3 Omnibus tests with level of significance $\alpha = 0.01$ [21]

	Friedman	Friedman AR	Quade
Statistic	314.0385689	376.5765013	18.60402682
<i>P</i> – Value	2.47195E–52	4.73527E–65	4.88167E–54

To determine if there are statistically significant differences in performance in terms of the best routes found, the Friedman, Friedman Aligned-Ranks and Quade omnibus test [21] was applied among all pairs of heuristics. Table 3 shows the results of the omnibus test using the median as a statistical representative, where h_o : There are no differences in the performance of the heuristics and h_a : There are differences between heuristics, where the p value is lower than the significance value $\alpha = 0.01$ (indicating that there is not enough evidence to accept h_o).

Taking as a control methodology the heuristic [4–4], because it has the lowest rank in the three tests, post hoc tests with a value of $\alpha = 0.01$ were performed. Tables 4, 5 shows the corresponding z values and the p values [21]. In the tests in pairs for all the test cases, the p value is lower than α , so there are differences between the behavior of the heuristics.

5 Conclusions and Future Work

In this work, a combinatorial design methodology based on algebraic structures has been presented to look for solutions to specific artificial instances of the VRP problem, which is a class of CSP. In the methodology an algebraic structure is applied to implicitly establish problem restrictions. In the problem presented in this article, the algebraic structure used were the Mutually Orthogonal Latin Squares (MOLS) and movements are proposed that preserve the orthogonality of the Latin squares, as well as the constraints of the VRP. The exploration of the solution space was carried out by defining heuristics through movements in the algebraic structure instead of directly in the instance. The analysis of the results of the experimentation carried out shows that in two of the three tests, the performance of a pair of heuristics applied in the proposed structure had statistically significant differences. These partial results suggest the possibility of using combinatorial design methodologies based on algebraic structures for the application of heuristics that improve the exploration of the search space in CSP problems.

Table 4 Post-hoc test with level of significance $\alpha = 0.01$ [21]

Control (4-4)	1-1	1-2	1-3	1-4	1-5		
Friedman							
Z	169.72	169.72	37.423	19.152	99.503		
P-value	0.0	0.0	0.0	0.0	0.0		
Friedman AR							
Z	120.66	120.66	4.1931	1.5829	9.3926		
P-value	0.0	0.0	2.7512E-05	0.113441956	0.0		
Quade							
Z	2.2222	2.2222	0.7275	0.5183	1.4862		
P-value	0.026265086	0.026265086	4.67E-01	0.604184081	0.137212266		
Control (4-4)	2-1	2-2	2-3	2-4	2-5	3-1	3-2
Friedman							
Z	169.72	169.72	40.285	28.838	88.716	29.498	33.241
P-value	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Friedman AR							
Z	120.66	120.66	3.6794	2.6731	8.0822	2.8723	3.8052
P-value	0.0	0.0	0.00023371	0.007514794	6.66134E-16	0.004074953	0.000141647
Quade							
Z	2.2222	2.2222	0.7598	0.4210	1.2394	0.5839	0.5969
P-value	0.026265086	0.026265086	0.447328729	0.673734166	2.15E-01	0.559257128	0.550534534

Table 5 Post-hoc test with level of significance $\alpha = 0.01$ [21]

Control (4-4)	3-3	3-4	3-5	4-1	4-2	4-3	4-5
Friedman							
Z	34.782	30.159	90.697	31.259	14.529	29.939	89.376
P-value	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Friedman AR							
Z	3.6270	2.4005	8.4911	2.3481	1.0063	2.3900	8.5540
P-value	0.000286661	0.016369458	0.0	0.018866515	0.314245581	0.016844271	0.0
Quade							
Z	0.3771	0.5482	1.1114	0.6131	0.4063	0.6767	1.1715
P-value	0.706049895	0.583501621	0.266354765	0.539805993	0.684501919	0.498576871	0.241395619
Control (4-4)	5-1	5-2	5-3	5-4	5-5		
Friedman							
Z	114.69	95.320	63.840	62.739		103.24	
P-value	0.0	0.0	0.0	0.0		0.0	
Friedman AR							
Z	11.877	9.4240	5.0317	5.0841		10.472	
P-value	0.0	0.0	4.85979E-07	3.69215E-07		0.0	
Quade							
Z	1.5949	1.3756	0.8594	1.0095		1.3626	
P-value	0.110724709	0.16893517	3.90E-01	3.13E-01		0.173002517	

As future work, the experimentation and analysis of the methodology in other variants of the VRP and other CSP problems is proposed.

Acknowledgements The authors acknowledge the support provided by the National Council of Science and Technology of Mexico (CONACYT), through the Postgraduate Scholarships: 446105 (J. A. Montesino), 446106 (L. de M. Ortiz) and the Research Grant CÁTEDRAS-2598 (A. Rojas). We also wish to thank the National Technological Institute of Mexico for providing facilities for our Doctoral studies in Computer Sciences (J.A. Montesino and L. de M. Ortiz).

References

1. Colborne, C.J., Van Oorschot, P.C.: Applications of combinatorial designs in computer science. *ACM Computing Surveys (CSUR)*, June 1989 **21**(2), pp. 223–250. New York, NY, USA (1988)
2. Bose, R.C.: On the construction of balanced incomplete block designs. *Ann. Eugen.* **9**, 358–399 (1939) Raj Chandra Bose
3. Stison, D.R.: *Combinatorial Designs, Constructions and Analysis*. Springer, New York (2004). ISBN 0-387-95487-2
4. Keedwell, D., Dénes, J.: *Latin Squares and their Applications*, 2nd edn. Elsevier. ISBN 978-0-444-63555-6 (2015)
5. Tang, J.: *Latin Squares and Their Applications*. N.p.: n.p. (2009). Web. 1 Ago. 2018. <http://www.elscorcho.50webs.com/latinsquares.pdf>
6. Jeavons, P.: On the algebraic structure of combinatorial problems. *Theor. Comput. Sci.* **200**(1), 185–204 (1998)
7. Mackworth, A.K.: Consistency in networks of relations. *Artif. Intell.* **8**, 99–118 (1977)
8. Comellas, F., Fábrega, J., Sánchez, A., Serra, O.: *Matemática Discreta*. ED UPC. Translation by Andreu, M. (2001) ISBN 84-8301-456-4
9. Dantzig, G.B., Ramser, J.H.: The truck dispatching problem. *Manag. Sci.* **6**(1), 80–91 *INFORMS*, Oct 1959
10. Garey, M.R., Johnson, D.S.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, New York (1979)
11. CVRPLIB (07–02–2019) Capacitated Vehicle Routing Problem Library. Recovered from <http://vrp.atd-lab.inf.puc-rio.br/index.php/en/>
12. Polya, G.: *How to Solve It: A New Aspect of Mathematical Method*, 2nd edn. Princeton University Press, Nov 1971
13. Tonge, F.M.: The use of heuristic programming in management science 151; a survey of the literature. *Manag. Sci.* **7**, 231, 237, Apr 1961
14. Misevicius, A.B.A.: Combining 2-opt, 3-opt and 4-opt with k-swap-kick perturbations for the traveling salesman problem. In *Information Technologies 2011: Proceedings of the 17th International Conference on Information and Software Technologies, IT 2011*, pp. 45, 50. Technologija, Kaunas, Lithuania, Kaunas (2011)
15. Croes, G.A.: A method for solving traveling salesman problems. *Oper. Res.* **6**, 791–812 (1958)
16. Du, D.Z., Ko, K.I., Hu, X.: Greedy strategy. In: *Design and Analysis of Approximation Algorithms*. Springer Optimization and Its Applications, vol 62. Springer, New York, NY (2012)
17. M.R.C.: *Simulation in Business and Economics*, 1st edn. Prentice Hall, Englewood Clis (1969)
18. Michel, G., Jean-Yves, P.: *Handbook of Metaheuristics (International Series in Operations Research & Management Science)*. Springer, Jan 2003
19. Ortiz-Aguilar, L.M.: *Diseño de horarios de alumnos y maestros mediante técnicas de Soft Computing, para una Institución Educativa*. Master's thesis, Instituto Tecnológico de León (2016)

20. Soria-Alcaraz Jorge, A., Carpio, M., Puga, H., Sotelo-Figueroa, M.: Methodology of design: a novel generic approach applied to the course timetabling problem. In: Melin, P., Castillo, O. (eds.) *Soft Computing Applications in Optimization, Control, and Recognition, Studies in Fuzziness and Soft Computing*, vol. 294, pp. 287–319. Springer, Berlin, Heidelberg (2013)
21. Derrac, J., Garcia, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **1**(1), 3–18 (2011)

Comparative Analysis of Multi-objective Metaheuristic Algorithms by Means of Performance Metrics to Continuous Problems



Carlos Juárez-Santini, J. A. Soria-Alcaraz, M. A. Sotelo-Figueroa and Esteves-Jiménez Velino

Abstract Multi-objective problems (MOP) are an important type of real-world problem that involves more than one objective that must be optimized simultaneously. There exists a wide variety of metaheuristic algorithms aimed to work on that type of problems, however, the selection of which algorithm should be used to a given MOP depends on the expertise of the researcher. This decision is not straightforward and usually means to use extra computational resources to try different multi-objective metaheuristics even before to try to solve the interested domain. In the state of the art, there exists several metrics to compare and contrasts the performance of two or more given multi-objective algorithms. In this work, we use these metrics to compare a set of well-known multi-objective metaheuristics over the continuous CEC 2009 benchmark with the objective to give the interested researcher useful information to properly select a multi-objective algorithm.

Keywords Multi-objective algorithms · Multi-objective metrics · CEC 2009 benchmark

1 Introduction

The multi-objective problems are a common type of problem. These problems are based on not produce a single solution but a set of solutions or Pareto front. Each solution is the result of an optimization process for two or more objectives which used

C. Juárez-Santini · J. A. Soria-Alcaraz (✉) · M. A. Sotelo-Figueroa · E.-J. Velino
División de Ciencias Económico-Administrativas, Departamento de Estudios Organizacionales,
Universidad de Guanajuato, Guanajuato, Mexico
e-mail: jorge.soria@ugto.mx

C. Juárez-Santini
e-mail: ca.juarezsantini@ugto.mx

M. A. Sotelo-Figueroa
e-mail: masotelo@ugto.mx

E.-J. Velino
e-mail: velino@ugto.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_37

the same set of variables. In the state of the art exist a wide number of algorithms that had been proposed to solve this type of problems. The efficacy of these algorithms is measured in terms of their performance over the problem to be solved using several metrics. A metric is a real value that measures a specific characteristic of a given Pareto front. For example, diversity of the given solutions, the volume of the area generated by the Pareto front etc. These multi-objective metrics allows the researcher to compare the quality of an observed Pareto solution set obtained by a multi-objective optimization method [1]. This allows the researcher to select the algorithm whit the desired performance.

In this work, we used four well-known algorithms of the state of the art. Namely, NSGA-II, MOPSO, MOEA/D and MOEA/D-DRA. These algorithms were tested with a set of metrics: Hypervolume, Generational Distance, Inverted Generational Distance, Spread and the Additive Unitary Epsilon Indicator. Both, algorithms and metrics, were executed using their implementations from the jMetal framework. As for benchmark, we use CEC 2009 competition problems to test our algorithms with fixed function points. In order to find statistical significance in our experimentation, we use the Shapiro-Wilk Test, ANOVA one-way Test, Pairwise T-test and a Boxplot graphic to visualize the performance of the four algorithms mentioned above over the suite of problems CEC 2009 competition.

The paper is organized as follows. Section 2 presents the problems definitions as well as important concepts related to Multi-objective optimization problems, Pareto Front metrics, and Statistical Analysis. Section 3 presents the solution approach and its justification. Section 4 contains the experimental set-up, results, their analysis, and discussion. Finally, Sect. 5 contains our conclusions and describes some future work.

2 Background

In this Section, we describe the multi-objective algorithms selected as well as the used metrics and some characteristics of the CEC 2009 competition problems.

2.1 *Multi-objective Algorithms*

This kind of algorithms has usually implemented to optimize the multi-objective problems because their essential capacity to escape from local optima, their extensibility in the multi-objective problems and their ability to incorporate the handling of linear and non-linear inequality and equality constraints in a straightforward way [2].

2.1.1 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

In this paper, we use the NSGA-II algorithm based on [3] which is an improved version of the NSGA algorithm [4] where authors made an improvement to counteract the computational complexity that NSGA had. In NSGA-II the authors proposed remove that shared parameter and replace it by a crowded-comparison operator. This operator works as a process selection guide in sundry algorithm stage toward an extended optimal Pareto Front and also they added elitism to make the selection. Algorithm 1 shows the implementation used [3, 5].

Algorithm 1: NSGA-II

Start

- 1: $R_t = P_t \cup Q_t$
- 2: $F = \text{fast-non-dominated-sort}(R_t)$
- 3: $P_{t+1} = \emptyset$ and $i = 1$
- 4: **while** $|P_{t+1}| + |F_i| \leq N$ **do**
- 5: *crowding-distance-assignment* (F_i)
- 6: $P_{t+1} = P_{t+1} \cup F_i$
- 7: $i = i + 1$
- 8: **end while**
- 9: $\text{Sort}(F_i \prec_n)$
- 10: $P_{t+1} = P_{t+1} \cup F_i[1:(N - |P_{t+1}|)]$
- 11: $Q_{t+1} = \text{create-new-population}(P_{t+1})$
- 12: $t = t + 1$

End

2.1.2 Multi-objective Particle Swarm Optimization (MOPSO)

This algorithm was proposed in [6] and we used a version proposed in [7] which is it based on Pareto dominance and an elitist selection through crowding factor, authors incorporated two mutation operators: uniform mutation and non-uniform mutation. The uniform mutation refers to variability range allowed for each decision variable is kept constant over generations and the non-uniform mutation has as characteristic variability range allowed for each decision variable decrease over time. Finally, the authors added the ϵ -dominance concept that is the final size of the external file where store the non-dominated solutions. Algorithm 2 shows an abstract of the implementation used.

Algorithm 2: MOPSO

Require: Initialize Swarm P_i , Initialize Leaders L_i .

- 1: Send L_i to ε -file
 - 2: $crowding(L_i)$, $g = 0$
 - 3: **while** $g < g_{max}$ **do**
 - 4: **for** $i = 1$ each particle:
 - 5: Select leader
 - 6: Fly
 - 7: Mutation
 - 8: Evaluate
 - 9: Update p_{best}
 - 10: **end for**
 - 11: Update L_i
-

2.1.3 Multi-objective Evolutionary Algorithm Based on Decomposition (MOEA/D)

The most algorithms multi-objective evolutionary algorithms (MOEAD) are based on Pareto dominance [8]. In these algorithms, the utility of each individual solution is mainly determined by Pareto dominance relationship with other found solutions in the search before. MOEA/D is based on decomposition (i.e. aggregations) [9] which means that an optimization problem is decomposed into different subproblems of simple optimization or mono-optimization and then it applies a method based on population to optimize the subproblems simultaneously with the main goal, each of these problems is the aggregation of all objectives. Neighborhood relations between these subproblems are defined on the function of measures between their aggregation coefficient vectors; each subproblem is optimized in MOEA/D using information mainly from its neighbors' subproblems [10]. In this paper, we used the algorithm proposed in [11] where it is required to decompose the multiobjective problem with your own considered and they used Tchebycheff approach that is an aggregation function [12]. Algorithm 3 details the implementation used from Jmetal framework.

Algorithm 3: MOEA/D**Require:**

Each generation t , MOEA/D with *Tchebycheff* approach maintains:

- A population N points $x^1, \dots, x^N \in \Omega$, where x^i is the current solution of i -th subproblem.
- FV^1, \dots, FV^N , where FV^i is F -value of x^i , that is: $FV^i = F(x^i)$ each $i = 1, \dots, N$.
- $z = (z_1, \dots, z_m)^T$, where z_i is the best value found so far the objective f_i .
- An external population (EP) which is used to store the non-dominated solutions found during the search.

Input:

1: A problem multiobjective to solve.

2: $a \leftarrow$ a stopping criterion.

3: $N \leftarrow$ Number of the subproblems considered in MOEA/D

4: A uniform spread of N weight vectors: $\lambda^1, \dots, \lambda^N$

5: $T \leftarrow$ Number of the weight vectors in the neighborhood of each weight vector.

Output: external population (EP)

6: Initialize:

1. Set $EP = 0$.
2. Calculate the Euclidian distances between any two weight vectors and then work out the T closest weight vectors. For each $i = 1, \dots, N$, set $B = \{i_1, \dots, i_T\}$, where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are T weight vectors closest to λ^i .
3. Generate an initial population x^1, \dots, x^N randomly or by a specific-problem method. Set $FV^i = F(x^i)$.
4. Initialize $z = (z_1, \dots, z_m)^T$ by a specific-problem method.

7: Update:**for 1 N do**

1. *Crossover*: Select two index j, k randomly from $B(i)$, and generate a new solution y from x^j and x^k by using genetic operators.
2. *Mutation*: Apply a specific problem repair/improve heuristic on y to produce y' .
3. *Update z*:
 - for each** $j = 1, \dots, m$
 - if** $z_j < f_j(y')$, **then** $z_j = f_j(y')$.
 - end for**
4. *Update of Neighboring Solutions*:
 - for each** index $j \in B(i)$,
 - if** $g^{te}(y' | \lambda^j, z) \leq g^{te}(x^j | \lambda^j, z)$, **then** $x^j = y'$ $FV^j = F(y')$.
 - end for**
 - // check in page. 715 in [11] more details
5. *Update EP*:
 - Remove from EP all vectors dominated by $F(y')$.
 - Add $F(y')$ in EP **if no** vector in EP dominate $F(y')$.

end for

2.1.4 MOEA/D with Dynamic Resource Allocation (MOEA/D-DRA)

MOEA/D-DRA is an improvement proposed in [9] to the algorithm MOEA/D. The authors proposed this improvement given that in previous versions of MOEA/D reviewed, it was found that all subproblems are treated by equal, this means that each one of the subproblems used similar or approximately the same amount of computation effort. Therefore, the authors explain that these subproblems can have different computational difficulties, so it is reasonable to assign different amounts of computational effort to different subproblems. Their improvement consist to define and compute an *utility* π^i for each subproblem i with this computational efforts are distributed to the subproblems in function to their utilities. Algorithm 4 shows the implementation used from Jmetal framework.

Algorithm 4: MOEA/D-DRA

1: Initialize:

1. Calculate the Euclidian distances between any two weight vectors and then work out the T closest weight vectors. For each $i = 1, \dots, N$, set $B = \{i_1, \dots, i_T\}$, where $\lambda^{i_1}, \dots, \lambda^{i_T}$ are T weight vectors closest to λ^i .
2. Generate an initial population x^1, \dots, x^N by uniform randomly sampling from search space.
3. Initialize $z = (z_1, \dots, z_m)^T$ by setting $z_i = \min\{f_i(x^1), f_i(x^2), \dots, f_i(x^N)\}$.
4. Set $gen = 0$ and $\pi^i = 1$ for all $i = 1, \dots, N$.

2: Select of Subproblems for Search: The index of subproblems whose objectives are MOP individual objectives f_i are selected to from initial I . By using 10 -tournament selection based on π_i , select other $\lfloor \frac{N}{5} \rfloor - m$ index and add them to I .

3: for each $i \in I$, do:

1. *Selection of Mating/Update Range:*
Uniformly randomly generate a number $rand$ from $(0,1)$ then set:

$$P = \begin{cases} B(i), & \text{if } rand < \delta; \\ \{1, \dots, N\}, & \text{otherwise.} \end{cases}$$
2. *Reproduction:*
Set $r_1 = i$ and randomly select two indexes r_2 and r_3 from P , and then, generate a solution \bar{y} from x^{r_1} , x^{r_2} and x^{r_3} by a *DE operator*, and then perform a mutation operator on \bar{y} with probability p_m to produce a new solution y .
3. *Repair:*
if an element of y is out of the boundary of Ω , its value is reset to be a randomly selected value inside the boundary.

4. *Update z*:
 - for** each $j = 1, \dots, m$
 - if** $z_j < f_j(y')$, **then** $z_j = f_j(y')$.
5. *Update of Solutions*:
 - Set $c = 0$ and then do the following:
 - a) **if** $c = n_r$ or P is empty, go to **point 9**.
 - otherwise**, randomly pick an index j from P .
 - b) **if** $(y|\lambda^j, z) \leq g(x^j|\lambda^j, z)$, **then** $x^j = y$, $FV^j = F(y)$ and $c = c + 1$.
 - c) Delete j from P and to go (a).
 - end for**
- 4: **Stopping Criteria**:
 - if** stopping criteria is satisfied, then stop and output $\{x^1, \dots, x^N\}$ and $\{F(x^1), \dots, F(x^N)\}$.
- 10: $gen = gen + 1$.

2.2 Pareto Front Metrics

In this paper, several parametric metrics are used to test the performance of the four algorithms described in Sect. 2.1. The real importance for to do a metrics study reside in a wide acceptance of these metrics by the scientific community since it is necessary to reflect the quality of different algorithms as well as compare several approaches [13]. Previous related work [13] considered three aspects of a set of solutions: (1) Convergence i.e. the proximity to the Pareto optimal theoretical front; (2) Diversity as distribution and spread of solutions; and (3) Number of solutions.

2.2.1 Hypervolume (HV)

Hypervolume metric as known by other researchers as S-metric or Lebesgue measure and this metric is applied because gives as result a unique value in which it can measure both convergence and diversity [14].

Through this metric seeks to calculate the normalized volume of the objective space Ω covered by obtained Pareto set (P^*), it limited by a reference point r^* ;

$r^* = (r_1^*, \dots, r_m^*)$ in Ω . For solution $i \in P^*$, its calculate a hypercube c_i from solution i and reference point r .

2.2.2 Generational Distance (GD)

Generational Distance is a value that represents how far away the know Pareto Front is from the true Pareto Front in [15] is mentioned as an error measure. The authors

[14] say the smallest value indicates that all solutions are on the true Pareto Front. Then, to calculate GD its used Eq. 1

$$GD = \frac{\sqrt{\sum_{i=1}^n d_i^2}}{n} \quad (1)$$

where n is the number of vectors in the set of non-dominated solutions founded until the moment and d_i is the Euclidean distance between each of them and the closest element of the Pareto optimal set. Also in [16] when the GD value equals zero, $GD = 0$, it means that all elements generated are on the Pareto optimal set.

2.2.3 Inverted Generational Distance (IGD)

This metric uses the true Pareto Front as a reference and measures the distance of each one its elements from the true Pareto Front to a non-dominated solution obtained by an algorithm [14]. IGD measures the average Euclidian distance from the points uniformly distributed along the entire Pareto Front to its closest solution in he set of solutions obtained. So it avoids the situation where all solutions obtained are concentrated in a point which can lead to converge as long as diversity is not satisfied. This is why the metric IGD can measure the convergence and diversity of an algorithm simultaneously [17]. The smallest value of IGD is the algorithm that has the best performance [18]. To calculate IGD we use Eq. 2 proposed in [17]:

$$IGD(P, P^*) = \frac{\sum_{x^* \in P^*} dist(x^*, P)}{|P^*|} \quad (2)$$

where P^* is the set of points evenly distributed over true Pareto Front in the objective space and P is the obtained solutions set by an algorithm, $dist(x^*, P)$ is the Euclidian distance between a point $x^* \in P^*$ and its nearest neighbor in P and $|P^*|$ is the population size of P^* .

2.2.4 Spread (Δ)

Spread (Δ) metric measures the degree of dispersion achieved between the solutions obtained. In [3] seeks to obtain a solutions set that cover all the optima region of Pareto so the main interest with this metric is to achieve a set of solutions that covered all optimal Pareto.

In Deb [3] authors calculate the Euclidian distance d_i between consecutive solutions of the set of non-dominated solutions obtained and the average \bar{d} of these distances. Then, from the non-dominated solutions obtained set it is necessary to calculate the extreme solutions in the Pareto space by adjusting a parallel curve to a

true Pareto optimal front and use the next metric to calculate the amount of uniformity in the distribution. Equation 3 shows the calculus of this metric.

$$\Delta = \frac{d_f + d_l + \sum_{i=1}^{N-1} |d_i - \bar{d}|}{d_f + d_l + (N - 1)\bar{d}} \quad (3)$$

where d_f and d_l parameter are the Euclidian distances between the extreme solutions and the limit solutions of non-dominated obtained set. The value \bar{d} is the distances average d_i ; $i = \{1, \dots, N - 1\}$ this is assuming that there are solutions in the best non-dominated front. Optimal Pareto solutions (POS) that are smaller than Δ are considered the best POS that satisfies the condition of diversity [19].

2.2.5 Additive Unitary Epsilon Indicator ($I_{\varepsilon+}$)

This metric measures Pareto dominance. For two Pareto sets A and B, the epsilon indicator can be considered as a measure of the minimum distance that a set B must move so that a set A dominates it. If set A dominates set B, the indicated value will be negative. If B dominates A, then the value will be positive [20]. In [21] authors explains that two epsilon indicators: multiplicative and additive version both unary and binary. For the unary additive epsilon indicator $I_{\varepsilon+}^1$ first they define the unary multiplicative epsilon indicator $I_{\varepsilon}^1(A)$ as $I_{\varepsilon}^1(A) = I_{\varepsilon}(A, R)$, where R is any reference set of points. So, $I_{\varepsilon+}^1$ is defined analogously but is based on additive ε -dominance:

$$x^1 \leq_{\varepsilon+} x^2 \Leftrightarrow \forall i \in 1 \dots n : f_i(x^1) \leq \varepsilon + f_i(x^2)$$

Both unary epsilon indicators are to be minimized. In $I_{\varepsilon+}^1$ an indicator less equal than zero 0 implies that A weakly dominates the reference set R.

2.3 CEC 2009 Competition

The test instances or Benchmark Test are objectives functions in which the algorithms are tested to obtain data on their performance. In addition, these tests are proposed models that resemble real problems. This is why many tests have been proposed with restrictions and without restrictions within the multi-objective optimization to test the multi-objective algorithms. In this paper, we used CEC 2009 competition problems that these are a series of unconstrained problems which are well-known within the scientific community. CEC 2009 Special Session and Competition was proposed in [22]. For these problems are known in [23] as UF_{number} , for example: UF1 or UF01 for the unconstrained problem #1, UF2 for problem #2, ... seen [22], in our work we only use from UF1 until UF10. It is important that problem UF1, ..., UF7 are for two objectives and UF8, UF9 and UF9 are problems with three objectives.

2.4 *Statistical Test*

In this paper, different statistical tests are used to understand and visualize the results of the experiments with the algorithms multi-objective described in Sect. 2.1 and tested with the five Pareto Front metrics seen in Sect. 2.2. So that we need robust test to compare their performances in the CEC 2009 competition problems in Sect. 2.3 and validate our conclusions. Four tests were used to achieve this goal: Shapiro-Wilk Test, ANOVA one-way, Pairwise T-test with “Bonferroni” correction and Boxplot graphic. Shapiro-Wilk test is a test of normality in frequentist statistics published in 1965 by Samuel Sanford Shapiro and Martin Wilk [24, 25] which they give focus to find if a set of data have or follow a normal distribution. Analysis of Variance (ANOVA) is a collection of statistical models used to analyze the difference between the group mean and its associated procedures. This analysis was developed by the statistician and evolutionary biologist Ronald Fisher [26]. In this paper, we used a one-way F ANOVA test [27] which supports the existence of a difference between the means of the algorithms with statistical significance. Finally, we used the Pairwise T-test with the correction “Bonferroni” for the ANOVA one-way test through the multiple comparisons pairwise using this correction to α . The fact of testing many pairs of groups is known as multiple comparisons.

2.5 *jMetal*

jMetal [28, 29] is a framework was developed mainly by Juan J. Durillo and Antonio J. Nebro. Their main goal was the development of metaheuristics to resolver multi-objective optimization problems. This framework is implemented in Java language under the object-oriented paradigm. In [28] says there is not a single point of reference for multiobjective problems commonly accepted by the scientific community also that there is agreement on the metrics that should be used to evaluate the performance of the metaheuristics and finally the parameter may vary between the experiments that are carried out in the different studies. jMetal has implemented a wide variety of tools to work with multi-objective problems, for example, multi-objective algorithms, test instances, operators, quality indicators or quality metrics and more tools.

3 Methodology

In this section, we detail the proposed methodology. We executed experiments with the four multi-objective algorithms described in Sect. 2.1. We used the implementations available in jMetal repository. Then, we applied five Pareto Front metrics described in Sect. 2.2. Finally, we use several statistical tests to analyze the Pareto Front metrics as seen in Sect. 2.4.

The jMetal framework, described in Sect. 2.5, is a framework or software tool that has several metaheuristics implemented, we use the jMetal implementation of four multi-objective algorithms described in Sect. 2.1, NSGA-II (Sect. 2.1.1), MOPSO (Sect. 2.1.2), MOEA/D (Sect. 2.1.3) and MOEA/D-DRA (Sect. 2.1.4) these algorithms were put to test over the CEC 2009 multi-objective instances. Once a given experiment ends, the algorithm produces a set of solutions. This set is known as a Pareto front, in the final phase we calculate the metrics detailed in Sect. 2.2 for a given Pareto front.

From the statistical tests, we take as statistical significance a value of 95%, this is equivalent to 0.05 of confidence level, i.e. $\alpha = 0.05$. For each statistical test, we obtained a *P-value* which it is compared against the selected confidence level to find if exists significant evidence to support the test's null-hypothesis H_0 . Equation 4 shows the applied criteria.

$$\begin{cases} P_{value} < 0.05, H_0(\text{its reject}) \\ P_{value} \geq 0.05, H_0(\text{its accepted}) \end{cases} \quad (4)$$

In the Shapiro-Wilk test, the null-hypothesis H_0 considers that the data comes from a normal distribution or that the population is normally distributed. So, if the *P-value* is less than α H_0 is rejected and since there is evidence that supports the notion of the data does not belong to a normally distributed population. In another case, if the *P-value* is greater than α , H_0 is accepted since the data belongs to a normally distributed population.

For ANOVA one-way test, the null-hypothesis H_0 is that data comes from a single normal distribution. A *P-value* greater than α means H_0 is accepted i.e. data comes from a single normal distribution. If the *P-value* is less than α H_0 is rejected, meaning the data does not come from a single normal distribution i.e. there exists a significance statistical evidence about two or more well defined normal populations are producing the given samples.

For Pairwise T-test, the null-hypothesis H_0 considers that for a given pair of Algorithms A and B, both produce results that are statistically similar or produced by a single normal distribution. Therefore, if the *P-value* is greater than α then H_0 is accepted, so both algorithms, A and B, produce results statistically similar. If the *P-Value* is less than α means H_0 is rejected, so algorithm A and B produce results statistically different.

Once the three tests detailed before were carried out, the quartiles and interquartile range are plotted by means of a Boxplot. This visualization tool allows us to see the best and worst performance for each algorithm. This whole process is repeated for each proposed metric.

4 Experiments and Results

In this section we present the results produced by the selected four multi-objective algorithms through the application of the Pareto front metrics, namely: Hypervolume, GD, IGD, Spread and $I_{\epsilon+}$. Each configuration was put to test on the CEC 2009 Competition problems, more specific, instances: UF1, UF2, UF3, UF4, UF5, UF6, UF7, UF8, UF9 and UF10 following the methodology detailed in Sect. 3. For each configuration tested, we executed 35 repetitions per instance and pareto metric. Parameters used can be seen in Tables 1, 2, 3 and 4 for NSGA-II, MOPSO, MOEA/D and MOEA/D-DRA respectively.

We proceed to apply out methodology detailed in Sect. 3, as a first step we apply the Shapiro-wilk test to our Pareto front results computed through Pareto front metrics. Results are shown in Table 5.

In Table 5, we show the P -values obtained by the Shapiro-Wilk test for each problem. Our results show that in most cases our results come from a normal distribution. For example, the values obtained in the UF2 problem for the Hypervolume Pareto Front metric by the MOPSO Algorithm. P -value of the algorithm MOPSO is greater

Table 1 NSGA-II configuration

Population size:	– 500
Max evaluations:	– 5000
<i>Crossover</i>	
Probability:	– 0.9
Distribution index:	– 20.0
Crossover operator:	– <i>SBX Crossover</i>
<i>Mutation</i>	
Mutation operator:	– <i>Polynomial mutation</i>
Probability:	– $1.0/(\text{Number of problem variables})$
Distribution index:	– 20.0
<i>Selection</i>	
Selection operator:	– <i>Binary tournament2</i>

Table 2 MOPSO configuration

Max particle size:	– 500
Max iterations:	– 5000
ϵ -file size:	– 500
<i>Uniform mutation</i>	
Mutation probability:	– $1.0/(\text{Number of problem variables})$
Disturbance index:	– 0.5
<i>Non-uniform mutation</i>	
Mutation probability:	– $1.0/(\text{Number of problem variables})$
Disturbance index:	– 0.5
Max iterations:	– 500

Table 3 MOEA/D configuration

Population size:	- 500
Max evaluations:	- 5000
T:	- 20
Delta:	- 0.9
nr:	- 2.0
<i>Crossover</i>	
CR:	- 1.0
F:	- 0.5
Crossover operator:	- <i>Differential Evolution Crossover</i>
<i>Mutation</i>	
Mutation operator:	- <i>Polynomial mutation</i>
Probability:	- $1.0/(\text{Number of problem variables})$
Distribution index:	- 20.0

Table 4 MOEA/D-DRA configuration

Population size:	- 500
Max evaluations:	- 5000
Final size:	- 300
T:	- 20
Delta:	- 0.9
nr:	- 2.0
<i>Crossover</i>	
CR:	- 1.0
F:	- 0.5
Crossover operator:	- <i>Differential Evolution Crossover</i>
<i>Mutation</i>	
Mutation operator:	- <i>Polynomial mutation</i>
Probability:	- $1.0/(\text{Number of problem variables})$
Distribution index:	- 20.0

than the confidence level (0.05), this means the null-hypothesis H_0 is accepted i.e. data comes from a normal distribution.

Values of zero in the GD metric means this metric measures the average distance of all its optimal solutions from Pareto Front regarding the optimal solutions of true Pareto Front. Algorithms that produce the most similar solution are MOPSO and NSGA-II. In the case for the Additive Unitary Epsilon Indicator $I_{\epsilon+}$ and IGD metric we obtained zero as their values when we execute 35 execute for each algorithm, so when we did the Statistical Analysis, we obtained zero in their result but this metric. After this phase we continue applying ANOVA one-way test with each metric. Table 6 Shows our results for a given algorithm and metric.

In Table 6, results for the Statistical test ANOVA-one way are shown. We use instance UF8 as an example to explain this table. With the $Pr(> F)$ as *P-value* and compared against the confidence level of α , we see that this value is less than α .

Table 5 Results Shapiro-Wilk tests

HV	NSGA-II	MOPSO	MOEA/D	MOEA/D-DRA
UF1	<i>P-value</i> = 0.0547	<i>P-value</i> = 0.8343	<i>P-value</i> = 0.6326	<i>P-value</i> = 0.3218
UF2	<i>P-value</i> = 0.5746	<i>P-value</i> = 9.95E-06	<i>P-value</i> = 0.3451	<i>P-value</i> = 0.1736
UF3	<i>P-value</i> = 0.1646	<i>P-value</i> = 0.1238	<i>P-value</i> = 0.0025	<i>P-value</i> = 0.9820
UF4	<i>P-value</i> = 0.6536	<i>P-value</i> = 0.0007	<i>P-value</i> = 0.3923	<i>P-value</i> = 0.9288
UF5	<i>P-value</i> = 0.0272	<i>P-value</i> = 0.02494	<i>P-value</i> = 0.9566	<i>P-value</i> = 0.0407
UF6	<i>P-value</i> = 0.1028	<i>P-value</i> = 0.0036	<i>P-value</i> = 0.0007	<i>P-value</i> = 0.0076
UF7	<i>P-value</i> = 0.0952	<i>P-value</i> = 1.68E-17	<i>P-value</i> = 0.3039	<i>P-value</i> = 0.0683
UF8	<i>P-value</i> = 0.3081	<i>P-value</i> = 0.00577	<i>P-value</i> = 0.3063	<i>P-value</i> = 0.3573
UF9	<i>P-value</i> = 0.6107	<i>P-value</i> = 0.00463	<i>P-value</i> = 0.334	<i>P-value</i> = 0.3853
UF10	<i>P-value</i> = 0.3565	<i>P-value</i> = 0.3612	<i>P-value</i> = 0.9232	<i>P-value</i> = 0.5716
GD	NSGA-II	MOPSO	MOEA/D	MOEA/D-DRA
UF1	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.004	<i>P-value</i> = 5.56E-11
UF2	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.007	<i>P-value</i> = 2.57E-10
UF3	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.001	<i>P-value</i> = 6.47E-13
UF4	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.006	<i>P-value</i> = 4.11E-12
UF5	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.003	<i>P-value</i> = 2.98E-12
UF6	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.001	<i>P-value</i> = 2.31E-11
UF7	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.001	<i>P-value</i> = 1.37E-11
UF8	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.007	<i>P-value</i> = 1.46E-08
UF9	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0.003	<i>P-value</i> = 0.000171
IGD	NSGA-II	MOPSO	MOEA/D	MOEA/D-DRA
UF1	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0
UF2	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0
UF3	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0
UF4	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0
UF5	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0	<i>P-value</i> = 0

(continued)

Table 5 (continued)

HV	NSGA-II	MOPSO	MOEA/D	MOEA/D-DRA
UF6	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF7	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF8	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF9	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF10	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
Spread	NSGA-II	MOPSO	MOEA/D	MOEA/D-DRA
UF1	$P\text{-value} = 0.7126$	$P\text{-value} = 0.05873$	$P\text{-value} = 0.5669$	$P\text{-value} = 0.3724$
UF2	$P\text{-value} = 0.8299$	$P\text{-value} = 0.5768$	$P\text{-value} = 0.3482$	$P\text{-value} = 0.8177$
UF3	$P\text{-value} = 0.4781$	$P\text{-value} = 0.387$	$P\text{-value} = 0.4717$	$P\text{-value} = 0.8877$
UF4	$P\text{-value} = 0.01922$	$P\text{-value} = 0.8984$	$P\text{-value} = 0.3075$	$P\text{-value} = 0.7515$
UF5	$P\text{-value} = 0.9716$	$P\text{-value} = 0.00755$	$P\text{-value} = 0.3473$	$P\text{-value} = 0.001548$
UF6	$P\text{-value} = 0.1964$	$P\text{-value} = 0.05796$	$P\text{-value} = 0.5538$	$P\text{-value} = 0.375$
UF7	$P\text{-value} = 0.54$	$P\text{-value} = 0.03282$	$P\text{-value} = 0.4992$	$P\text{-value} = 0.2225$
UF8	$P\text{-value} = 0.8822$	$P\text{-value} = 0.8911$	$P\text{-value} = 0.4446$	$P\text{-value} = 0.04748$
UF9	$P\text{-value} = 0.5902$	$P\text{-value} = 0.02417$	$P\text{-value} = 0.6815$	$P\text{-value} = 0.6472$
UF10	$P\text{-value} = 0.0551$	$P\text{-value} = 0.5933$	$P\text{-value} = 0.1302$	$P\text{-value} = 0.6449$
$I_{\varepsilon+}$	NSGA-II	MOPSO	MOEA/D	MOEA/D-DRA
UF1	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF2	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF3	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF4	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF5	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF6	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF7	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF8	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF9	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$
UF10	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$	$P\text{-value} = 0$

Table 6 Results ANOVA one way

Hypervolume		Df	Sum Sq	Mean Sq	F value	Pr (>F)
UF1	<i>ind</i>	3	0.19722	0.06574	13.104	1.44E-07
	<i>Residuals</i>	136	0.6823	0.005017		
UF2	<i>ind</i>	3	0.13561	0.045202	17.128	1.71E-09
	<i>Residuals</i>	136	0.35892	0.002639		
UF3	<i>ind</i>	3	0.85684	0.285612	71.538	<2.20E-16
	<i>Residuals</i>	136	0.54297	0.003992		
UF4	<i>ind</i>	3	0.031773	0.0105909	19.717	1.13E-10
	<i>Residuals</i>	136	0.073052	0.0005371		
UF5	<i>ind</i>	3	0.52707	0.175688	9.1342	1.50E-05
	<i>Residuals</i>	136	2.61583	0.019234		
UF6	<i>ind</i>	3	2.5203	0.84009	53.566	<2.20E-16
	<i>Residuals</i>	136	2.1329	0.01568		
UF7	<i>ind</i>	3	1.0073	0.33576	36.165	<2.20E-16
	<i>Residuals</i>	136	1.2627	0.00928		
UF8	<i>ind</i>	3	0.58788	0.19596	44.742	<2.20E-16
	<i>Residuals</i>	136	0.59564	0.00438		
UF9	<i>ind</i>	3	0.52824	0.17608	50.598	<2.20E-16
	<i>Residuals</i>	136	0.47328	0.00348		
UF10	<i>ind</i>	3	0.59189	0.197298	46.397	<2.20E-16
	<i>Residuals</i>	136	0.57832	0.004252		
Generational distance		Df	Sum Sq	Mean Sq	F value	Pr(>F)
UF1	<i>ind</i>	3	0.00117452	0.00039151	174.99	<2.20E-16
	<i>Residuals</i>	136	0.00030428	0.00000224		
UF2	<i>ind</i>	3	0.02673	0.0089107	1.2702	0.2872
	<i>Residuals</i>	136	0.95408	0.0070153		
UF3	<i>ind</i>	3	0.00043045	1.43E-04	144.08	<2.20E-16
	<i>Residuals</i>	136	0.00013543	9.96E-07		
UF4	<i>ind</i>	3	1.32E-05	4.39E-06	893.02	<2.20E-16
	<i>Residuals</i>	136	6.68E-07	4.90E-09		
UF5	<i>ind</i>	3	0.0045786	0.0015262	113.01	<2.20E-16
	<i>Residuals</i>	136	0.0018368	0.00001351		
UF6	<i>ind</i>	3	0.0053466	0.0017822	47.154	<2.20E-16
	<i>Residuals</i>	136	0.0051402	0.0000378		
UF7	<i>ind</i>	3	0.00139347	0.00046449	184.38	<2.20E-16
	<i>Residuals</i>	136	0.00034261	0.00000252		

(continued)

Table 6 (continued)

Hypervolume		Df	Sum Sq	Mean Sq	F value	Pr (>F)
UF8	<i>ind</i>	3	0.0048546	0.0016182	83.196	<2.20E-16
	<i>Residuals</i>	136	0.0026453	0.00001945		
UF9	<i>ind</i>	3	0.0084895	0.00282984	66.007	<2.20E-16
	<i>Residuals</i>	136	0.0058306	0.00004287		
UF10	<i>ind</i>	3	0.0213163	0.0071054	143.22	<2.20E-16
	<i>Residuals</i>	136	0.0067471	0.0000496		
Spread		Df	Sum Sq	Mean Sq	F value	Pr(>F)
UF1	<i>ind</i>	3	33.007	11.0025	710.51	<2.20E-16
	<i>Residuals</i>	136	2.106	0.0155		
UF2	<i>ind</i>	3	45.14	15.0468	5721.3	<2.20E-16
	<i>Residuals</i>	136	0.358	0.0026		
UF3	<i>ind</i>	3	35.32	11.7732	1354.2	<2.20E-16
	<i>Residuals</i>	136	1.182	0.0087		
UF4	<i>ind</i>	3	32.322	10.774	6720.4	<2.20E-16
	<i>Residuals</i>	136	0.218	0.0016		
UF5	<i>ind</i>	3	33.99	11.1329	254.54	<2.20E-16
	<i>Residuals</i>	136	5.948	0.0437		
UF6	<i>ind</i>	3	30.1487	10.0496	247.76	<2.20E-16
	<i>Residuals</i>	136	5.5164	0.0406		
UF7	<i>ind</i>	3	37.635	12.545	1992.1	<2.20E-16
	<i>Residuals</i>	136	0.856	0.0063		
UF8	<i>ind</i>	3	1.77184	0.59061	379.08	<2.20E-16
	<i>Residuals</i>	136	0.21189	0.00156		
UF9	<i>ind</i>	3	1.59195	0.53065	218.47	<2.20E-16
	<i>Residuals</i>	136	0.33034	0.00243		
UF10	<i>ind</i>	3	2.11282	0.70427	281.16	<2.20E-16
	<i>Residuals</i>	136	0.34066	0.0025		

So, the null-hypothesis H_0 is rejected i.e. observed data do not come from a single normal distribution which means that the algorithms have a different performance with statistical significance. Then we used Pairwise T-test with Bonferroni correction to identify which exact algorithms offer statistically different results. (This last step with Bonferroni correction is applied to each algorithm an metric).

In the case of the UF8 problem, Results from Pairwise T-test are shown in Table 7 where it can be seen the P -Value produced when comparing two specific algorithms. In this case, MOEA/D-DRA and MOEA/D are the only algorithms with similar performance. We know this since the P -Value is greater than 0.05 so, the null-hypothesis

Table 7 Pairwise T test results for Hypervolume in UF8 problem

	MOEA/D	MOEA/D-DRA	MOPSO
MOEA/D-DRA	1	–	–
MOPSO	1.50E–15	<2e–16	–
NSGA-II	0.0077	8.00E–05	7.80E–08

H_0 is accepted. Multi-objective algorithms NSGA-II and MOPSO produce different results with statistical significance.

Figure 1 shows a Boxplot with the performance of the four algorithms obtained from Table 5 UTF 8. NSGA-II as MOPSO have a different performance in comparison among other algorithms. In the case of MOEA/D and MOEA/D-DRA are similar between them. We can also see that MOPSO algorithm has the best performance because it offers maximum results for Hypervolume, followed by NSGA-II as the second better algorithm. Finally, the worse algorithms in this case were MOEA/D and MOEA/D-DRA. These are similar but the average of MOEA/D is over of MOEA/D-DRA.

Table 8 shows algorithms that produced the best performance over each Pareto Front metric. The results were taken from the Statistical Analysis that we executed using the same methodology as detailed in previous sections, more specific from the Pairwise T tests where we compared each pair of algorithms and contrasted this information with its position in the Boxplot graphic. Importantly, in Table 8 some algorithms show symbol “*”, this means that this algorithm is similar only to the

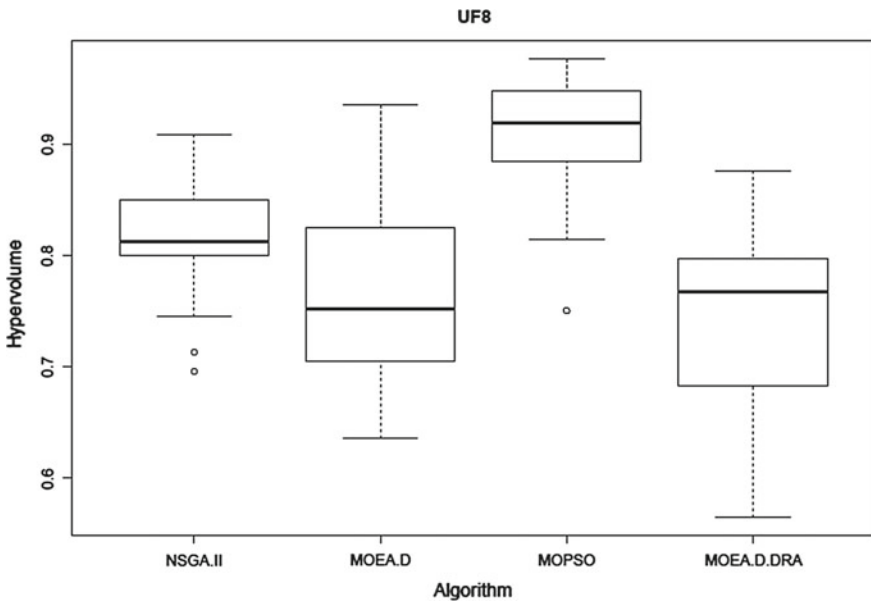


Fig. 1 Boxplot for hypervolume in UF8

Table 8 Algorithms with the best performance for each metric

	HV	GD	IGD	Spread	Epsilon indicator
UF1	<i>NSGA-II</i>	<i>NSGA-II</i> and <i>MOPSO</i> and <i>MOEA/D-DRA</i> *	4	<i>MOEA/D-DRA</i>	4
UF2	<i>NSGA-II</i> and <i>MOEA/D</i> and <i>MOEA/D-DRA</i>	4	4	<i>MOEA/D-DRA</i>	4
UF3	<i>NSGA-II</i> and <i>MOEA/D</i> and <i>MOEA/D-DRA</i>	<i>NSGA-II</i> and <i>MOPSO</i> and <i>MOEA/D-DRA</i> *	4	<i>MOEA/D-DRA</i>	4
UF4	<i>NSGA-II</i>	<i>NSGA-II</i> and <i>MOPSO</i> and <i>MOEA/D-DRA</i> *	4	<i>MOEA/D-DRA</i>	4
UF5	<i>NSGA-II</i> and <i>MOEA/D</i> and <i>MOEA/D-DRA</i>	<i>NSGA-II</i> and <i>MOPSO</i> and <i>MOEA/D-DRA</i> *	4	<i>MOEA/D-DRA</i>	4
UF6	<i>NSGA-II</i> and <i>MOEA/D</i> and <i>MOEA/D-DRA</i>	<i>NSGA-II</i> and <i>MOPSO</i> and <i>MOEA/D-DRA</i> *	4	<i>MOEA/D-DRA</i>	4
UF7	<i>NSGA-II</i>	<i>NSGA-II</i> and <i>MOPSO</i> and <i>MOEA/D-DRA</i> *	4	<i>MOEA/D-DRA</i>	4
UF8	<i>MOPSO</i>	<i>NSGA-II</i> and <i>MOPSO</i>	4	<i>MOEA/D</i> and <i>MOEA/D-DRA</i>	4
UF9	<i>MOPSO</i>	<i>NSGA-II</i> and <i>MOPSO</i>	4	<i>MOEA/D</i> and <i>MOEA/D-DRA</i>	4
UF10	<i>MOPSO</i>	<i>NSGA-II</i> and <i>MOPSO</i>	4	<i>MOEA/D</i> and <i>MOEA/D-DRA</i>	4

algorithm that is before it. For example, in the UF1 instance with GD metric, we have these algorithms: *NSGA-II*, *MOPSO* and *MOEA/D-DRA**; *MOEA/D-DRA* has the symbol “*”, so *MOEA/D-DRA* is similar to *MOPSO*, but it is not similar to *NSGA-II*. Also, in Table 8 can appear the number “4” that means the four algorithms are statistically similar among them.

For the Hypervolume metric, we gather statistically evidence that for three instances the best algorithm was *NSGA-II*. *MOPSO* was the best in other three instances. Although for the rest of the instances, we obtained that three algorithms have a similar performance.

With the Generational Distance (GD) metric, we obtained that for all instances at least two algorithms have similar performance. For six instances we obtained that *NSGA-II*, *MOEA/D* and *MOEA/D-DRA* are similar between themselves. In the UF2 instance, we observed that the four algorithms have similar performance. Finally, in the rest of problems, the best algorithms were *NSGA-II* and *MOPSO*.

For the Inverted Generational Distance (IGD) metric, we observed that all the four algorithms converge in the same Pareto Front, so their results are zero in this metric. We can use any of these algorithms because their performance is similar among themselves.

In the Spread column, the multi-objective algorithm MOEA/D-DRA predominates because it is the algorithm that produces better solutions. In UF8, UF9 and UF10 instances we observed that the multi-objective algorithms MOEA/D and MOEA/D-DRA have similar performance.

For the Additive Unitary Epsilon Indicator, we observed that all algorithms have similar performance since the difference between the solutions generated and the reference point is zero.

In summary; the Pareto Front metrics IGD, GD produced very similar observations. IGD and Additive Unitary Epsilon Indicator were not a good way to compare the performance of two given algorithms under our configuration. Our results showed also that the MOPSO and MOEA/D-DRA multi-objective algorithms are consistently the algorithms with the best performance under all metrics and instances.

5 Conclusions and Future Work

This paper presents a comparative analysis of four well-known multi-objective algorithms in the state of the art by means of their observed performance through five Pareto front metrics. This comparative followed a statistical process that authors suggest to replicate in similar cases. As future work, we propose to implement more algorithms of the state of the art because these algorithms present improvements to previous algorithms. Suggested algorithms to test can be NSGA-III or an improved version of the MOEA/D algorithm. In the case of the instance problems, authors recommend to work with the rest of the instances from CEC 2009 competitions as UF11, UF12, UF13, UF14, and UF15 and to execute tests over the ZDT instance problems; more specific with ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6 Instances.

Acknowledgements Authors express their gratitude to the Universidad de Guanajuato (UG) for its support provided to carry out the present research.

References

1. Wu, J., Azarm, S.: Metrics for quality assessment of a multiobjective design optimization solution set. *J. Mech. Des.* **123**(1), 18 (2001)
2. Giagkiozis, I., Purshouse, R.C., Fleming, P.J.: An overview of population-based algorithms for multi-objective optimisation. *Int. J. Syst. Sci. pbo'ijss' final Int. J. Syst. Sci.* **46**(00), 1–42 (2013)
3. Deb, K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **6**(2), 181–197 (2002)

4. Srinivas, N., Deb, K.: Multiobjective optimization using nondominated sorting in genetic algorithms. *Evol. Comput.* **2**(3), 221–248 (1994)
5. Dorado, D., Guasmayán, F., Bravo, M., Peluffo, D.: Estudio comparativo de NSGA-II y PSO como métodos de optimización multiobjetivo en problemas con frente optimo de pareto convexo. Nov 2016
6. Coello Coello, C.A., Lechuga, M.S.: MOPSO: a proposal for multiple objective particle swarm optimization. In *Proceedings of the 2002 Congress on Evolutionary Computation, CEC 2002*, Vol. 2, pp. 1051–1056 (2002)
7. Sierra, M.R., Coello Coello, C.A.: Improving PSO-based multi-objective optimization using crowding, mutation and ϵ -dominance, pp. 505–519. Springer, Berlin, Heidelberg (2005)
8. Wolpert, D.H., Macready, W.G.: No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1**(1), 67–82 (1997)
9. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 203–208 (2009)
10. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **13**(2), 284–302 (2009)
11. Zhang, Q., Li, H.: MOEA/D: a multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **11**(6), 712–731 (2007)
12. Khan, W., Zhang, Q.: MOEA/D-DRA with two crossover operators. In *2010 UK Workshop on Computational Intelligence, UKCI 2010*, pp. 1–6 (2010)
13. Riquelme, N., Von Lüken, C., Barán, B.: Performance metrics in multi-objective optimization. In: *Proceedings—2015 41st Latin American Computing Conference, CLEI 2015*, pp. 1–11 (2015)
14. Lwin, K., Qu, R., Kendall, G.: A learning-guided multi-objective evolutionary algorithm for constrained portfolio optimization. *Appl. Soft Comput. J.* **24**, 757–772 (2014)
15. Reguianski, T.L.: The air force institute of technology. *IRE Trans. Educ.* **E-5**(2), 117–118 (1962)
16. Coello Coello, C.A., Pulido, G.T., Lechuga, M.S.: Handling multiple objectives with particle swarm optimization. *IEEE Trans. Evol. Comput.* **8**(3), 256–279 (2004)
17. Li, X., Wei, L.X., Fan, R., Sun, H., Hu, Z.Y.: A hybrid multiobjective particle swarm optimization algorithm based on R2 indicator. *IEEE Access* **6**, 14710–14721 (2018)
18. Liu, H.L., Chen, L., Zhang, Q., Deb, K.: Adaptively allocating search effort in challenging many-objective optimization problems. *IEEE Trans. Evol. Comput.* **22**(3), 433–448 (2018)
19. Sato, H., Aguirre, H.E., Tanaka, K.: Local dominance using polar coordinates to enhance multi-objective evolutionary algorithms. In: *Proceedings 2004 Congress Evolutionary Computation*, vol. 1, pp. 188–195 (2004)
20. Dawson, P., Parks, G., Jaeggi, D., Molina-Cristobal, A., Clarkson, P.J.: The development of a multi-threaded multi-objective tabu search algorithm. In: Dawson, P., Parks, G., Jaeggi, D., Molina-Cristobal, A., John Clarkson, P (eds.) *Evolutionary Multi-Criterion Optimization*. Springer, Berlin, Heidelberg, pp. 242–256 (2007)
21. Zitzler, E., Knowles, J., Thiele, L.: Quality Assessment of Pareto Set Approximations, pp. 373–404. Springer, Berlin, Heidelberg (2008)
22. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W.: Multiobjective optimization test instances for the CEC 2009 special session and competition. In: *2009 IEEE Congress on Evolutionary Computation (CEC 2009)*, pp. 1–30 (2009)
23. Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In: *2009 IEEE Congress on Evolutionary Computation, CEC 2009*, pp. 203–208 (2009)
24. Soria-Alcaraz, J.A., Sotelo-Figueroa, M.A., Espinal, A.: Statistical comparative between selection rules for adaptive operator selection in vehicle routing and multi-knapsack problems. In: *Studies in Computational Intelligence*, Vol. 749, pp. 389–400 (2018)
25. Jia, K., He, Z.: DOA identification of communication emitters based on Shapiro-Wilk test and divisive hierarchical cluster analysis Kexin. In: *2010 6th International Conference on Wireless Communications Networking and Mobile Computing (WiCOM)*, pp. 1–4 (2010)

26. Danila, A., Ungureanu, D., Moraru, S.A., Voicescu, N.: An implementation of the variance analysis (ANOVA) for the power factor optimization at distribution level in smart grid. In: Proceedings—2017 International Conference on Optimization of Electrical and Electronic Equipment, OPTIM 2017 and 2017 Intl Aegean Conference on Electrical Machines and Power Electronics, ACEMP 2017, pp. 48–53 (2017)
27. Soria-Alcaraz, J.A., Espinal, A., Sotelo-Figueroa, M.A.: Evolvability metric estimation by a parallel perceptron for on-line selection hyper-heuristics. *IEEE Access* **5**, 7055–7063 (2017)
28. Durillo, J.J., Nebro, A.J., Luna, F., Dorronsoro, B., Alba, E.: *jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics* (2015)
29. Durillo, J.J., Nebro, A.J., Alba, E.: The *jMetal* framework for multi-objective optimization: design and architecture. In *IEEE Congress on Evolutionary Computation*, pp. 1–8 (2010)

Intelligent Agents

Towards an Agent-Based Model for the Analysis of Macroeconomic Signals



Alejandro Platas-López, Alejandro Guerra-Hernández,
Nicandro Cruz-Ramírez, Marcela Quiroz-Castellanos, Francisco Grimaldo,
Mario Paolucci and Federico Cecconi

Abstract This work introduces an agent-based model for the analysis of macroeconomic signals. The Bottom-up Adaptive Model (BAM) deploys a closed Walrasian economy where three types of agents (households, firms and banks) interact in three markets (goods, labor and credit) producing some signals of interest, e.g., unemployment rate, GDP, inflation, wealth distribution, etc. Agents are bounded rational, i.e., their behavior is defined in terms of simple rules finitely searching for the best salary, the best price, and the lowest interest rate in the corresponding markets, under incomplete information. The markets define fixed protocols of interaction adopted by the agents. The observed signals are emergent properties of the whole system. All this contrasts with the traditional macroeconomic approach based on the general equilibrium model, where perfect rationality and/or full information availability are assumed. The model is defined following the Overview, Design concepts, and Details

A. Platas-López (✉) · A. Guerra-Hernández · N. Cruz-Ramírez · M. Quiroz-Castellanos
Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho
5, 91000 Xalapa, Veracruz, Mexico
e-mail: alejandroplatas1@gmail.com

A. Guerra-Hernández
e-mail: aguerra@uv.mx

N. Cruz-Ramírez
e-mail: ncruz@uv.mx

M. Quiroz-Castellanos
e-mail: mquiroz@uv.mx

F. Grimaldo
Departament d'Informàtica, Universitat de València, Avinguda de la Universitat s/n, 46100
Burjassot, València, Spain
e-mail: francisco.grimaldo@uv.es

F. Grimaldo · M. Paolucci · F. Cecconi
Italian National Research Council, Institute of Cognitive Sciences and Technologies, Via Palestro
32, 00185 Rome, Italy
e-mail: mario.paolucci@istc.cnr.it

F. Cecconi
e-mail: federico.cecconi@istc.cnr.it

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_38

Protocol and implemented in NetLogo. BAM is promoted as a toolbox for studying the macroeconomic effects of the agent activities at the service of the elaboration of public policies.

Keywords Agent-based model · Macroeconomic · ODD protocol

1 Introduction

Both in the natural and social sciences, there are complex processes that (1) consist of many agents which interact with each other (2) exhibit emergent global properties, and (3) lack a centralized control governing such properties [1]. Analyzing these systems as a whole is an extremely complicated task, so models are used to describe them. A model is an abstract representation of reality, in which only the relevant characteristics of the system are considered for the analysis.

In economics, the continuous relationship between various agents such as households, companies, banks and the government, generates a large number of macroeconomic signals, such as production, unemployment, inflation, interest rates, among others. In macroeconomics, two approaches are distinguished to model this phenomena, the classical approach (top-down) based on the theory of general equilibrium and a new approach (bottom-up) based on agents [2].

The top-down models rest on the theory of general equilibrium, whose central statement establishes that from the interaction between supply and demand derives a general equilibrium on all markets. An important characteristic of these models is the market clearing condition (Walrasian auctioneer), which is given by a central authority that proposes a set of prices, determines an excess of demand at these prices and adjusts them to their equilibrium values. The roots of this approach go back to the nineteenth century, when many economists tried to formulate a full general equilibrium model, but it was conceived until 1874 by Leon Walras, a French economist [3]. The most recent versions of this model incorporate dynamism (the economic variables consider the expectations of the future), and randomness (as a source of uncertainty) and are called Dynamic Stochastic General Equilibrium (DSGE) models. The solution in this type of models is found when solving systems of equations, e.g., households optimize a utility function subject to a budget constraint, while companies maximize their profit subject to the restriction of technological resources [4].

One of the main limitations of these models is the assumption of equilibrium, since it is too simplistic for collecting the complexity of economic processes over time. Although external shocks can be used to get out of the equilibrium, by its nature, DSGE picks up small fluctuations around a stationary state, analyzing and predicting the signals of the economy in this way. So, these models behave well when there are no disturbances, but predict poorly when risk and uncertainty come into play.

Another disadvantage of this approach is that by the very nature of this approach, modeled through equations, agents are assumed homogeneous, i.e., they have the

same information and worse, they have complete information of the system with which they determine their optimal plans. Finally, the Walrasian trial and error mechanism has no counterpart in the real market economy, and goes against the spirit of complex systems, where there is no centralized control.

On the other hand, the bottom-up models conceive complex systems as composed of autonomous interactive agents. Agents base their behavior on simple rules and interact with other agents, which in turn influences their behavior. Two important features of this type of models are that (1) each agent has its own attributes and behavior, i.e., heterogeneity (2) the effects of the diversity among agents can be observed in the behavior of the system as a whole, emergence [5]. Despite their simplicity, these models are not devoid of rationality [2], economic agents guide their behavior to achieve a utility, i.e., instead of coding a specific goal, a measure is defined, allowing the agent to decide what is better for them, e.g., higher salary offered by firms, lower interest rate of banks, better leverage of firms. Although always within the cognitive limitations of the agents.

Bottom-up models do not make assumptions about the efficiency of markets or the existence of an equilibrium, so they can absorb the tensions or disturbances generated in periods of crisis through the emerging behavior resulting from the interaction between agents, in such a way that the panic of agents eventually spreads to the whole system. Finally, these models are non-linear, which implies that the generated effects do not have to be proportional to their causes. This allows to identify the causes in areas that in principle are not related. In some models, the effects can be of a magnitude much greater than the causes that provoke them while in others the effects dissipate in a conventional manner.

The main contribution of the paper is offering a complete and concise, basic Bottom-up Adaptive Model (BAM) based on the work of Delli Gatti et al. [6]. The model is described adopting the Overview, Design concepts, and Details (ODD) protocol [7, 8], for the sake of reproducibility. The resulting system is available at Github.¹ The paper is organized as follows: Sect. 2 introduces the BAM model conceptually, for then offering details accordingly to the ODD Protocol. Section 3 presents the implementation of the model in NetLogo. Section 4 presents results, as well as the empirical validation of the model by fulfilling some stylized facts used in economic theory. Finally, Sect. 5 presents our conclusions and future work.

2 The BAM Model

Despite the criticism for its excessive abstraction, the Walrasian economic model has persisted as a fundamental paradigm [9]. Indeed, because of its simplicity, it is a good starting point for exploring both perfect and imperfect economic models.

¹<https://github.com/alexplatasl/BAMmodel/>.

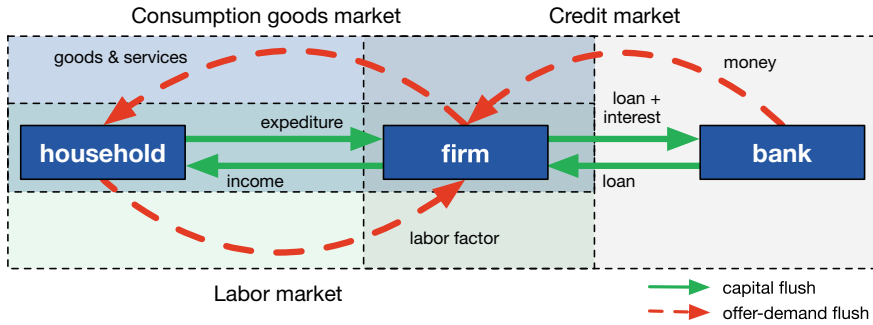


Fig. 1 The bottom-up adaptive macroeconomics model (BAM)

The Bottom-up Adaptive Model (BAM) [6] adopted in this paper is Walrasian in nature. As shown in Fig. 1, it is composed by the following types of agents:

- **Households** representing the point of consumption and labor force.
- **Firms** representing the transformation of work in goods and/or services.
- **Banks** providing liquidity to firms if necessary.

A large number of autonomous households, producers and banks operate adaptively in three totally decentralized and interconnected markets:

- A **labor market**, in which each household offers an inelastic unit of work per period, while firms demand depending on their production plans;
- A perishable consumer **goods market**, in which households spend all or part of their wealth and firms offer goods at different prices; and
- A **credit market** in which firms demand money if their resources are insufficient to cover their production expenses, and banks offer money at different interest rates.

Opportunities for exchange in these markets are discovered through a sequential process characterized by optimization, namely, maximizing wages, minimizing the price of goods consumed and minimizing the price of money (interest rate). Firms can modify prices and quantities adaptively given the signals of the inventory and the market price.

BAM was adopted because the agents that intervene in the model are those necessary to model disturbances that are similar to those observed in a real world economy; while generating macroeconomic signals of interest, e.g., inflation, unemployment, wealth, production among others are generated.

As mentioned in the introduction, for the sake of reproducibility, the details of the model will be described following the ODD protocol, which is organized in three parts:

1. **Overview.** A general description of the model, including its purpose and its basic components: agents, variables describing them and the environment, and

scales used in the model, e.g., time and space; as well as a processes overview and their scheduling.

2. **Design concepts.** A brief description of the basic principles underlying the model’s design, e.g., rationality, emergence, adaptation, learning, etc.
3. **Details.** Full definitions of the involved submodels.

2.1 Overview

Purpose. Exploring the use of the bottom-up approach for the study of macroeconomic signals, particularly the effect of the agent’s activities in such signals.

Entities, state variables, and scales.

- Agents: Firms, workers, and banks.
- Environment: Agents are situated in a grid environment which is meaningless with respect to the model. The environment is used exclusively as a visual aid for debugging.
- State variables: The attributes that characterize each agent are shown in Table 1.

Table 1 State variables by agent

Agent	Attribute	Type	Agent	Attribute	Type
Firm	Production-Y	Int	Worker	Employed?	Bool
	Desired-production-Yd	Int		My-potential-firms	AgSet
	Expected-demand-De	Int		My-firm	Ag
	Desired-labor-force-Ld	Int		Contract	Int
	My-employees	AgSet		Income	Float
	Current-numbers-employees-L0	Int		Savings	Float
	Number-of-vacancies-offered-V	Int		Wealth	Float
	Minimum-wage-W-hat	Float		Propensity-to-consume-c	Float
	Wage-offered-Wb	Float		My-stores	AgSet
	Net-worth-A	Float		My-large-store	Ag
Total-payroll-W	Float	Bank	Total-amount-of-credit-C	Float	
Loan-B	Float		Patrimonial-base-E	Float	
My-potential-banks	AgSet		Operational-interest-rate	Float	
My-bank	AgSet		Interest-rate-r	Float	
Inventory-S	Float		My-borrowing-firms	AgSet	
Individual-price-P	Float		Bankrupt?	Bool	
Revenue-R	Float				
Retained-profits-pi	Float				

- Scales: Time is discrete, e.g., each step represents a quarter. Quarters are adequate for long periods, months can be used for short ones.

Process overview and scheduling. The main loop of the simulation is as follows:

1. Firms calculate production based on expected demand.
2. A decentralized labor market opens.
3. A decentralized credit market opens.
4. Firms produce.
5. Market for goods open.
6. Firms will pay loan and dividends.
7. Firms and banks will survive or die.
8. Replacing of bankrupt firms/banks.

2.2 Design Concepts

Basic Principles. The model follows fundamental principles of neoclassical economics [10], since it gives great importance to money in economic processes and also the strategy for determining prices is given considering both supply and demand.

Emergence. The model generates adaptive behavior of the agents, without the imposition of an equation that governs their actions. Macroeconomic signals are also emergent properties of the system.

Adaptation. At each step, firms can adapt price or amount to supply (only one of the two strategies). Adaptation of each strategy depends on the condition of the firm (level of excessive supply/demand in the previous period) and/or the market environment (the difference between the individual price and the market price in the previous period).

Objectives. Agents do not explicitly have an objective, but implicitly they try to maximize a utility or attribute.

Learning. None for the moment, however, see the future work section for possible uses of learning in this model.

Prediction. Firms predict the quantities to be produced or the price of the good produced based on the excess of supply/demand in the previous period and the differential of its price and the average price in the market.

Sensing.

- Firms perceive their own produced quantity, good's price, labor force, net value, profits, offered wages; as well as the average market price and the interest rate of randomly chosen banks.
- Workers perceive the size of firms visited in the previous period, prices published by the firms in actual period and wages offered by the firms.

- Banks perceive net value of potential borrowers in order to calculate interest rate.

Interaction. Interactions among agents are determined by the markets:

- In the labor market, firms post their vacancies at a certain offered wage. Then, unemployed workers contact a given number of randomly chosen firms to get a job, starting from the one that offers the highest wage. Firms have to pay the wage bill in order to start production. A worker whose contract has just expired applies first to his/her last employer.
- Firm can access to a fully decentralized credit market if net worth are in short supply with respect to the wage bill. Borrowing firms contact a given number of randomly chosen banks to get a loan, starting from the one which charges the lowest interest rate. Each bank sorts the borrowers' applications for loans in descending order according to the financial soundness of firms, and satisfy them until all credit supply has been exhausted. The contractual interest rate is calculated applying a mark-up on an exogenously determined baseline interest rate. After the credit market is closed, if financial resources are not enough to pay for the wage bill of the population of workers, some workers remain unemployed or are fired.
- In goods market, firms post their offer price, and consumers contact a given number of randomly chosen firms to purchase goods, starting from the one which posts the lowest price.

Stochasticity. Elements that have random shocks are:

- Determination of wages when vacancies are offered (ξ).
- Determination of contractual interest rate offered by banks to firms (ϕ).
- The strategy to set prices (η).
- The strategy to determine the quantity to produce (ρ).

Collectives. Markets configure collectives of agents as described above. They include labor, goods, and credit markets. In addition, firms and consumers are categorized as rich and poor.

Observation. Along simulation are observed:

- Logarithm of real GDP.
- Unemployment rate.
- Annual inflation rate.
- Interest rate.

At end of simulation are computed:

- Philips curve (inflation/unemployment).
- Distribution of the size of firms.
- Distribution of wealth of households.
- Growth rate of real GDP.

2.3 Details

Initialization. The initialization parameters described in Delli Gatti [6] was adopted. For the values not provided in the text, they were obtained through experimentation. Table 2 shows the initial values of the model.

Input data. None, although data from real economies might be used for validation.

Submodels.

1. Production with constant returns to scale and technological multiplier: $Y_{it} = \alpha_{it}L_{it}$, s.t., $\alpha_{it} > 0$.
2. Desired production level Y_{it}^d is equal to the expected demand D_{it}^d .
3. Desired labor force (employees) $L_{it}^d = Y_{it}^d/\alpha_{it}$.
4. Current number of employees L_{it}^0 is the sum of employees with and without a valid contract.
5. Number of vacancies offered by firms $V_{it} = \max(L_{it}^d - L_{it}^0, 0)$.
6. If there are no vacancies ($V_{it} = 0$), wage offered is $w_{it}^b = \max(\hat{w}_t, w_{it-1})$, where \hat{w}_t is the minimum wage determined by law.
7. If $V_{it} > 0$, wage offered is $w_{it}^b = \max(\hat{w}_t, w_{it-1}(1 + \xi_{it}))$, where ξ_{it} is a random term evenly distributed between $(0, h_\xi)$.

Table 2 Parameters initialization

	Parameter	Value
I	Number of consumers	500
J	Number of producers	100
K	Number of banks	10
T	Number of steps	1000
C_P	Propensity to consume of poorest people	1
C_R	Propensity to consume of richest people	0.5
σ_P	R&D investment of poorest firms	0
σ_R	R&D investment of richest firms	0.1
h_ξ	Maximum growth rate of wages	0.05
H_η	Maximum growth rate of prices	0.1
H_ρ	Maximum growth rate of quantities	0.1
H_ϕ	Maximum amount of banks' costs	0.1
Z	Number of trials in the goods market	2
M	Number of trials in the labor market	4
H	Number of trials in the credit market	2
\hat{w}	Minimum wage (set by a mandatory law)	1
P_t	Aggregate price	1.5
δ	Fixed fraction to share dividends	0.15

8. At the beginning of each period, a firm has a net value A_{it} . If total payroll to be paid $W_{it} > A_{it}$, firm asks for loan $B_{it} = \max(W_{it} - A_{it}, 0)$.
9. For the loan search costs, it must be met that $H < K$.
10. In each period the k -th bank can distribute a total amount of credit C_k equivalent to a multiple of its patrimonial base $C_{kt} = E_{kt}/v$, where $0 < v < 1$ can be interpreted as the capital requirement coefficient. Therefore, the v reciprocal represents the maximum allowed leverage by the bank.
11. Bank offers credit C_k , with its respective interest rate r_{it}^k and contract for 1 period.
12. If $A_{it+1} > 0$ the payment scheme is $B_{it}(1 + r_{it}^k)$.
13. If $A_{it+1} \leq 0$, bank retrieves R_{it+1} .
14. Contractual interest rate offered by the bank k to the firm i is determined as a margin on a rate policy established by Central Monetary Authority \bar{r} , s.t., $R_{it}^k = \bar{r}(1 + \phi_{kt}\mu(\ell_{it}))$.
15. Margin is a function of the specificity of the bank as possible variations in its operating costs and captured by the uniform random variable ϕ_{kt} in the interval $(0, h_\phi)$.
16. Margin is also a function of the borrower's financial fragility, captured by the term $\mu(\ell_{it})$, $\mu' > 0$. Where $\ell_{it} = B_{it}/A_{it}$ is the leverage of borrower.
17. Demand for credit is divisible, i.e., if a single bank is not able to satisfy the requested credit, it can request in the remaining $H - 1$ randomly selected banks.
18. Each firm has an inventory of unsold goods S_{it} , where excess supply $S_{it} > 0$ or demand $S_{it} = 0$ is reflected.
19. Deviation of the individual price from the average market price during the previous period is represented as: $P_{it-1} - P_{t-1}$.
20. If deviation is positive $P_{it-1} > P_{t-1}$, firm recognizes that its price is high compared to its competitors, and is induced to decrease the price or quantity to prevent a migration massive in favor of its rivals; and vice versa.
21. In case of adjusting price downward, this is bounded below P_{it}^l to not be less than your average costs:

$$P_{it}^l = \frac{W_{it} + \sum_k r_{kit} B_{kit}}{Y_{it}}$$

22. Aggregate price P_t is common knowledge, while inventory S_{it} and individual price P_{it} are private.
23. Only the price or quantity to be produced can be modified. In the case of price, we have the following rule:

$$P_{it}^s = \begin{cases} \max[P_{it}^l, P_{it-1}(1 + \eta_{it})] & \text{if } S_{it-1} = 0 \text{ and } P_{it-1} < P \\ \max[P_{it}^l, P_{it-1}(1 - \eta_{it})] & \text{if } S_{it-1} > 0 \text{ and } P_{it-1} \geq P \end{cases}$$

where: η_{it} is a randomized term uniformly distributed in the range $(0, h_\eta)$ and P_{it}^l is the minimum price at which firm i can solve its minimal costs at time t (previously defined).

24. In the case of quantities, these are adjusted adaptively according to the following rule:

$$D_{it}^e = \begin{cases} Y_{it-1}(1 + \rho_{it}) & \text{if } S_{it-1} = 0 \text{ and } P_{it-1} \geq P \\ Y_{it-1}(1 - \rho_{it}) & \text{if } S_{it-1} > 0 \text{ and } P_{it-1} < P \end{cases}$$

where ρ_{it} is a random term uniform distributed and bounded between $(0, h_\rho)$.

25. Total income of households is the sum of the payroll paid to the workers in t and the dividends distributed to the shareholders in $t - 1$.
26. Wealth is defined as the sum of labor income plus the sum of all savings SA of the past.
27. Marginal propensity to consume c is a decreasing function of the worker's total wealth (higher the wealth lower the proportion spent on consumption) defined as:

$$c_{jt} = \frac{1}{1 + \left[\tanh \left(\frac{SA_{jt}}{SA_t} \right) \right]^\beta}$$

where SA_t is the average savings. SA_{jt} is the real saving of the j -th consumer.

28. The revenue R_{it} of a firm after the goods market closes is $R_{it} = P_{it}Y_{it}$.
29. At the end of t period, each firm computes benefits π_{it-1} .
30. If the benefits are positive, shareholders receive dividends $Div_{it-1} = \delta\pi_{it-1}$.
31. Residual, after discounting dividends, is added to net value from previous period A_{it-1} . Therefore, net worth of a profitable firm in t is:

$$A_{it} = A_{it-1} + \pi_{it-1} - Div_{it-1} \equiv A_{it-1} + (1 - \delta)\pi_{it-1}$$

32. If firm i accumulates a net value $A_{it} < 0$, it goes bankrupt.
33. Firms that go bankrupt are replaced with another one of size smaller than the average of incumbent firms.
34. Non-incumbent firms are those whose size is above and below 5%, the concept is used to calculate a more robust estimator of the average.
35. Bank's capital:

$$E_{kt} = E_{kt-1} + \sum_{i \in \Theta} r_{kit-1} B_{kit-1} - BD_{kt-1}$$

36. Θ is the bank's loan portfolio, BD_{kt-1} represents the portfolio of firms that go bankrupt.
37. Bankrupted banks are replaced with a copy of one of the surviving ones.

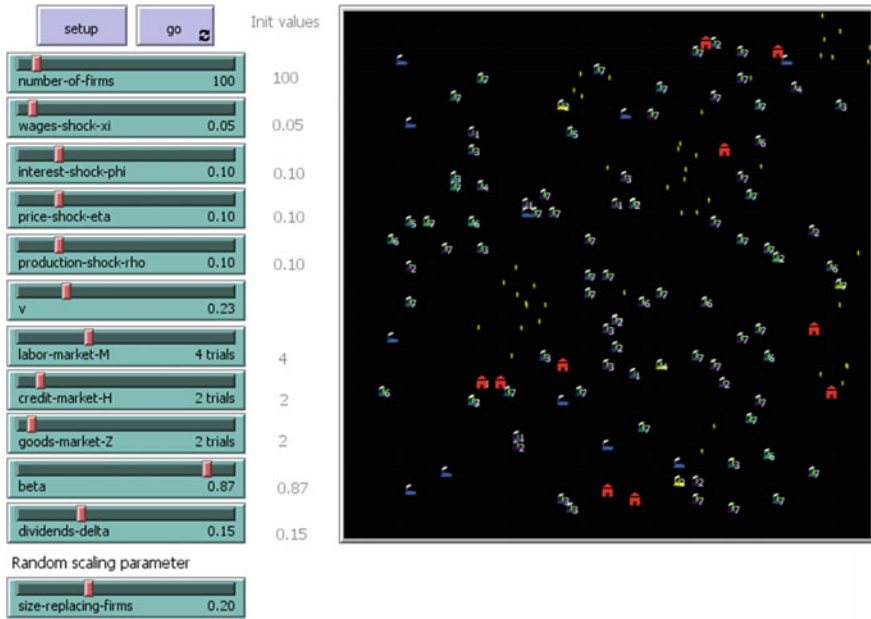


Fig. 2 The BAM model GUI: parameters and view of the world

3 Implementation

The BAM model was implemented in Netlogo [11]. Figure 2 shows the right side of the resulting GUI that allows the initialization of parameters and provides a view of the agents in a grid environment. As mentioned, the spacial issues in this view are meaningless, but the output is useful for debugging the system: Blue factories are the firms, red houses are the banks, green humans are employed workers while yellow ones are unemployed. Workers group around the firms where they work and shop. Factories display the number of employees.

4 Results

With the initial configuration of the parameters proposed by Delli Gatti et al. [6], the macroeconomic signals exemplified in Fig. 3 are produced. This output reflects a stable fictitious economy, with unemployment rate close to 10% and moderate inflation in the range of 1–6%. In the next section, some stylized facts that theoretically should show these signals will be tested.

At the micro level, validation consists of verifying the existence of stylized facts concerning statistical distributions of state variables at an individual level [6]. Wealth

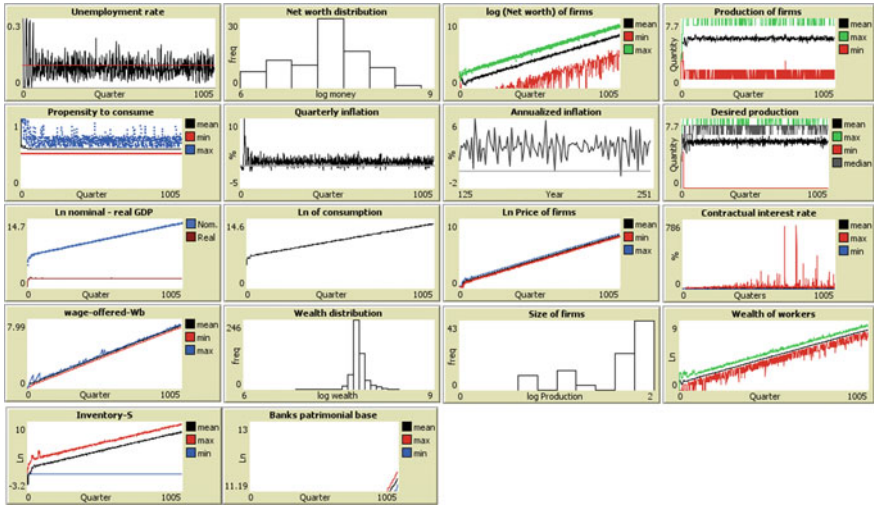


Fig. 3 The BAM model GUI: macroeconomic signals

and net worth in our case are characterized by a positive skew, which implies that there are few agents that become rich (Fig. 4).

To prove that the distributions of wealth of 100 independent runs have a positive skew (Fig. 5), level of skew was calculated with the method described by Joanes and Gill [12]:

$$b_1 = \frac{m_3}{s^3} = \left(\frac{n-1}{n} \right)^{3/2} \frac{m_3}{m_2^{3/2}} . \tag{1}$$

where,

$$m_r = \frac{1}{n} \sum (x_i - \bar{x})^r . \tag{2}$$

At the macro level it is assumed that a economy is characterized in the long run by balanced growth, so this assumption implies for example that growth rate of GDP is mean stationary [13], in other words, series do not have time-dependent structure. There are a number of non-stationary tests and the Augmented Dickey-Fuller may be one of the more widely used. It uses an autoregressive model and optimizes an information criterion across multiple different lag values [14].

Applying the test without and with trend for zero and 3 lags on last 500 quarter series of GDP growth of 100 independent runs, with $\alpha = 0.05$, it is possible to reject the null hypothesis of non-stationarity if the t-statistic value is less (more negative) than the critical values (-1.95 for test without trend and -3.42 for test with trend). As we shown in Fig. 6, for every independent run this stylized fact is fulfilled, GDP growth rate series are mean stationary.

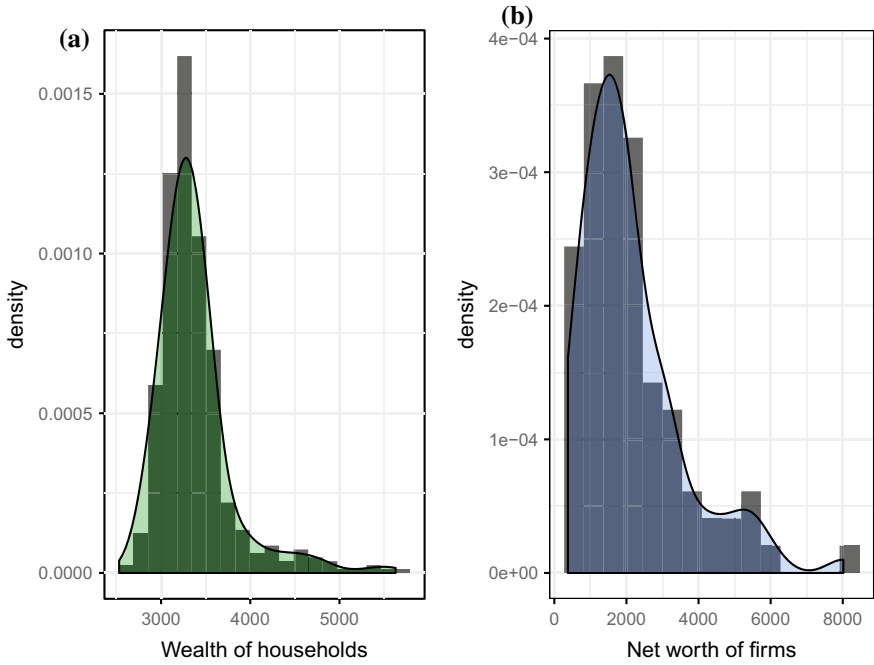


Fig. 4 Examples of our distribution of wealth (a) and net value (b) of a selected run

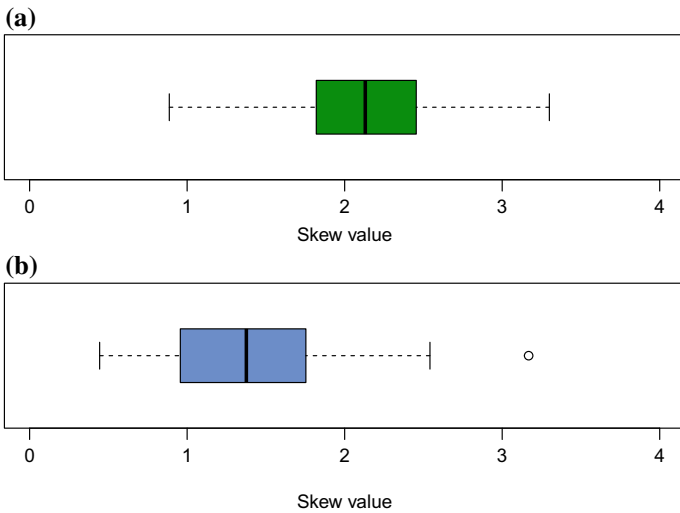


Fig. 5 Skewness values obtained over 100 independent runs of wealth distribution (a) and net worth (b). It is considered that values greater than 1 correspond to highly positively skewed distributions

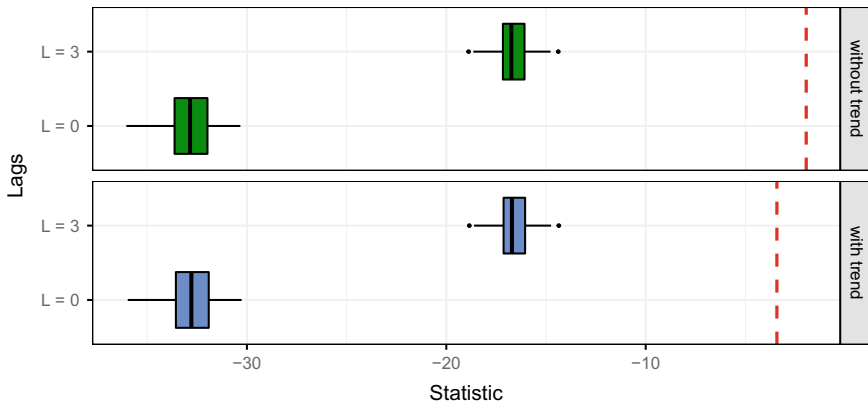


Fig. 6 Distribution of Dickey-Fuller t-statistics for logarithmic first differences of last 500 GDP quarters over 100 independent runs. Dashed lines are critical values

5 Conclusion and Future Work

The main contribution of this work is the complete definition of the BAM model and an open-source, full implementation of the model in NetLogo. The tests performed in this papers suggest that BAM is well suited for studying macroeconomic signals resulting from agent's activities and affected by external shocks, e.g., variation in the reference interest rate of the central bank.

Future work includes exploring the parameter space of BAM in order to get a better understanding of the behavior of the model, particularly for answering what-if questions, e.g., What happens to GDP if the reference interest rate change? Such exploration is also useful for validating other stylized facts.

BAM will be used to study the macroeconomic effects of extortion racket systems [15], e.g., with a certain probability, unemployed workers become extorters. They search for victims among their known companies that have not been already being extorted by another criminal. An extorted firm must take a decision about refusing to pay the extortion or paying; while extorters must decide to punish or not when the firms refused to pay. Such decisions depends on the probability of being punished, the probability of being captured by law, etc. What is the impact of extortion in the observed macroeconomic signals? Well, BAM can be used to compare such signals in the presence and absence of extortion.

Computational intelligence might be very useful for calibrating BAM for adjusting it to the behavior of a real particular economy. Data can be used to train models implementing the decisions of some of the agents in the model, e.g, the firms. Data can also be used to initialize the states variables of some agents, e.g., the workers. An study of modeling unemployment in Veracruz, Mexico based on Bayesian Networks [16], has followed this approach. Evolutionary computation might also be

explored as a tool for parameter calibration, e.g., finding the parameter values that minimize unemployment.

The current state of BAM is very encouraging for continuing with these lines of research. Open-sourcing it is also important for the validation of the model and to observe its applicability in other projects.

References

1. Boccarda, N.: *Modeling Complex Systems*. Springer, New York (2010)
2. De Grauwe, P.: Top-down versus bottom-up macroeconomics. *CESifo Econ. Stud.* **56**(4), 465–497 (2010)
3. Starr, R.M.: *General Equilibrium Theory: An introduction*. Cambridge University Press, New York (2011)
4. Cabezas-Gottschalk, E., et al.: Un modelo de equilibrio general dinámico y estocástico (DSGE) para la economía ecuatoriana. Banco Central del Ecuador, Quito (2016)
5. Macal, C.M., North, M.J.: Tutorial on agent-based modelling and simulation. *J. Simul.* **4**, 151–162 (2010)
6. Delli Gatti, D., et al.: *Macroeconomics from the Bottom-Up*. Springer, Milan (2011)
7. Grimm, V., et al.: A standard protocol for describing individual-based and agent-based models. *Ecol. Model.* **198**, 115–126 (2006)
8. Grimm, V., et al.: The ODD protocol: a review and first update. *Ecol. Model.* **221**, 2760–2768 (2010)
9. Tesfatsion, L.: Agent-based computational economics: a constructive approach to economic theory. In: Tesfatsion, L., Judd, K.L. (eds.) *Handbook of Computational Economics*, vol. 2, pp. 831–880. Elsevier, Amsterdam (2006)
10. Woodford, M.: Convergence in macroeconomics: elements of the new synthesis. *Am. Econ. J.: Macroecon.* **1**, 267–279 (2008)
11. Wilensky, U.: *NetLogo*. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL (1999)
12. Joanes, D.N., Gill, C.A.: Comparing measures of sample skewness and kurtosis. *Statistician* **47**, 183–189 (1998)
13. Evans, P.: *US stylized facts and their implications for growth theory*. Ohio State University, Columbus (2000)
14. Harris, R.I.D.: Testing for unit roots using the augmented Dickey-Fuller test: some issues relating to the size, power and the lag structure of the test. *Econ. Lett.* **38**, 381–386 (1992)
15. Troitzsch, K.G.: Distribution effects of extortion racket systems. In: Amblard, F., et al. (eds.) *Advances in Artificial Economics*, pp. 181–193. Springer (2015)
16. Díaz-Preciado, J.C., Guerra-Hernández, A., Cruz-Ramírez, N.: Un modelo de red bayesiana de la informalidad laboral en Veracruz orientado a una simulación social basada en agentes. *Res. Comput. Sci.* **113**, 157–170 (2016)

Fuzzy Worlds and the Quest for Modeling Complex-Adaptive Systems



Miguel Melgarejo

Abstract This chapter introduces the concept of fuzzy world as an ontological basis for modeling complex-adaptive systems. The concept is grounded on a phenomenological analysis of these systems over micro and macro scales. Discussion is developed from a recapitulation of some concepts of complexity science and complex systems modeling. Finally, the argument points out that fuzzy worlds find in fuzzy sets and systems theory a natural epistemological and methodological support.

1 Introduction

Complex systems surround our lives, these systems can be found not only across the natural world but also in technical worlds created by humans. Complex systems found in nature are so diverse that to develop a taxonomy for these systems has been a hard task. A very long list of complex systems can be given but surely a lot of pages should be written. There are some complex systems that are cited in literature for pedagogic purposes, for example ant colonies, flocking birds, oceans, cities, the atmosphere, the internet among others. We study about these examples superficially at some point of our basic education and we never hear about others in our lives. Although we are amazed by complex systems, most of the time how they behave remains a mystery for the majority of people.

Complexity science has emerged as a new field of study to tackle the problem of understanding the underlying mechanisms in complex systems, whereas complex systems engineering has been devoted to design and manage these systems. New categories have been identified as these two fields have been interacting, for example socio-technical systems refer to complex systems where both human and technology converge and socio-ecological systems describe the complex interaction of humans

M. Melgarejo (✉)
Universidad Distrital Francisco José de Caldas, Carrera 7a no 40-53 piso 5,
Bogotá, DC, Colombia
e-mail: mmelgarejo@udistrital.edu.co

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_39

567

and ecosystems. In fact the convergence of the three dimensions (social, technological and ecological) has called the attention of a portion of researchers and managers.

Modeling complex systems is an important part of the whole picture and the computational intelligence community has not been indifferent to this challenge. This chapter follows this interest presenting a discussion about developing complex models for complex-adaptive systems (i.e. complex systems where learning and adaptation is possible) from the ontological necessity of capturing complexity at both micro and macroscopic scales. A phenomenological perspective is introduced in which the building block for constructing complex simulation models relies on the idea of *fuzzy worlds*, a concept that takes a different path from the well-known agent-based framework.

The chapter is divided in three sections: a brief summary about complex and complex-adaptive systems, an argument in favor of building complex models that capture some attributes of complex-adaptive systems and finally the presentation of fuzzy worlds and their role in the quest to model these systems.

2 Complex Systems

2.1 *The Footprint of Complexity*

Nowadays there is not a unique definition for the attribute referred as complexity [26]. In terms of behavior, it is often used as a synonym for irregularity, uncertainty or unexpectedness. On the other hand, when emphasis is given to structure, complexity may refer to the abundance of variables and their relations. The absence of a formal postulate that defines complexity does not restrict the possibility of pointing out some elements that configure the footprint of a complex system [5, 18, 28]:

- *Nonlinearity*: complex systems exhibit the characteristics of non-linear dynamical systems such as sensitivity to initial conditions and non periodic but recurrent behaviors. Some forms of chaos can appear in the evolution of a complex system given by the multiplicity of non fixed interacting elements inside it.
- *Emergence*: unexpected behavior is a nice term that can be used in this context to make reference to an emergent property. In a complex system these behaviors depend on it as a whole and can not be explained by reducing them in terms of the individual parts that configure the system.
- *Self-organization*: a complex system is able to promote an internal organization in ordered and coherent states for a long range. Such states can sustain hierarchical structures that control disorder at local scales.
- *Unpredictability*: predicting the evolution of the dynamics of a complex system can be a hard task, not only because of the dynamics itself but also due to the incapacity of the observer. The measurement act may be a perturbation introduced in the system resulting in a modification of the trajectory the dynamics was following. In addition that modification can go back perturbing the observer, thus a sort of

communication is established between the system and who is observing it [28]. This deep interconnection implies that there would not be an ontological difference between both [16].

2.2 Levels of Analysis

The perception of complexity in a system is subject to the scale in which observations are made. Thus the analysis of complex systems should be performed in terms of a hierarchy of description levels. There is a serious connection between the level of description and the different epistemic perspectives that can guide the analysis process of the observer [27, 40]. A brief summary of such levels and related epistemic approaches is presented as follows:

- *Microscopic*: this is the level of fundamental interactions among the entities that constitute the complex system. Both Newtonian and quantum systems may be described from this level. That description is often characterized by an universality nuance in terms of laws that govern the behavior of such entities. These laws can be understood as minimal algorithms that determine the dynamics of the elements of the system.

The perspective that guides the analysis in this level of description is grounded on deterministic postulates. The dynamics of the complex systems can be studied in terms of universal principles like the Newton's laws or the Schrödinger's equation, which govern the behavior of the multiple entities inside the system. Deterministic chaos is often found in these cases due to the variability induced by the interaction among the entities.

Even though this level of analysis seems to be strictly destined to deal with systems governed by laws over their microscopic entities, it has been recognized the possibility of using it for approaching to systems where no complete knowledge of the rules is available, for example social systems [9]. However, the ignorance about rules would motivate to develop an approximation from other levels of analysis.

- *Macroscopic*: a new set of variables can be constructed from the multiplicity of entities found in the microscopic level. This set models the collective properties of such entities. Its dynamics is induced by clusters of microscopic processes. Therefore the number of macroscopic variables is much fewer than the entities that shape the system.

Equations that describe the behavior of macroscopic observations are presented as a balance between sources and sinks. This behavior often emerges as a nonlinear dynamics which is not reversible in time. On the other hand, dynamics of macroscopic variables can be understood as the smooth expression of all microscopic descriptions.

Expected variability in macroscopic levels is smaller than in microscopic ones. Thus the macroscopic description is understood as an average of microscopic

irregularities that retains wide scale features. In epistemic terms, the macroscopic representation is an empirical or ad hoc model that should be experimentally verified.

- *Mesoscopic*: an intermediate point between the previous levels can be sketched. Involved variables in this level are an extension of the macroscopic set. Possible descriptions of dynamics must consider the uncertainty associate to the variability of microscopic processes. Evolution in time should be understood in terms of a set of trajectories with finite fluctuations that can be modeled as a footprint around an average dynamics.

The nature of variables is assumed as random so that the system dynamics is studied in terms of how probability distributions evolve in time, for instance by using the Fokker-Planck equation [27]. The analysis of a complex systems in the mesoscopic level tries to model the uncertainty, which is classically performed by means of probability theory, however other ways are being investigated such as the possibility theory approach [20].

2.3 *Complex-Adaptive Systems*

From the spectrum of complex systems, those with learning and adaptation abilities are of special interest, particularly when these properties arise in the macroscopic level. These systems have been called as *Complex-adaptive systems* [18] since their internal configuration promotes the emergence of phenomena such as anticipation, collective behavior and evolution.

Human social systems configure an interesting class of complex-adaptive system whose main particularity is to set purposes not only in its interacting elements but also in the whole picture that identifies it [1]. The modeling of human social systems has followed different paths which cover mechanistic, organismic and more recently evolutionary approaches [10]. The main focus of these models are not the entities of the system (i.e. agents) but the rules that they generate and carry [9].

Approaching a social system implies taking advantage of an epistemic evolution to be able to capture the evolution of the dynamics of the system [35]. The multiple transformations that can occur from the adaptation of a social system make the search for its understanding only in terms of its parts too narrow. Thus to assign the category of “system” to a phenomenon demands an a-priori understanding of the context not only from the parts but also from the relations between them, which is the point where the purposes that identify the system can be observed.

Social systems demand an epistemology that recognizes the same definition of system may be dynamic, so that the epistemic background of social systems can be itself complex by integrating several forms of knowledge, for example one of them the approach from engineering [22]. Entities that shape the system can interpret it and disturb its original purposes, thus agents are at the same time parts and designers of the system. One relevant aspect of social agency is its immersion in technology [15], which makes human systems to be treated at same time as social and as technical

systems. This particularity produces in socio-technical systems, such as cities, the emergence of very unique properties that other similar complex systems found in nature do not exhibit [7].

Models for complex adaptive systems, in particular socio-technical systems, can be constructed from different epistemic perspectives. These go from the qualitative approaches of social sciences to the quantitative models of physics, crossing by the organismic models of biology [1], the agent-based models of computer science [13] or the rule-based models of the evolutionary economics [29] among others. An interesting aspect of these models has to do with the purpose behind them. Although prediction seems to be the unique purpose of modeling, today a multiplicity of purposes have been recognized which are connected to the intentions, perspectives or interests of the modeler [12].

3 Complex Models for Complex Adaptive Systems

3.1 Types of Complex Systems and Their Models

Complexity of a system is a wide concept that can be studied from different perspectives without the availability of a formal definition. The perspective that will be followed is described extensively in [3], which is focused on typifying a complex system according to its structure and the necessary knowledge to model and manage it. According to this framework complex systems can be typified as follows:

- *Type-I* complex systems are those whose dynamics is governed by simple rules but the system exhibits interesting behaviors such as irregular oscillations, chaotic attractors, self-organization and emergent properties. Models for this type of system are usually represented by regression equations, nonlinear differential equations, information flows etc. Some examples of this type are tectonic plates, sand piles and some cellular automata among others, whose dynamics can be explained from the Self-Organized Criticality (SOC) theory [4].
- *Type-II* complex systems are characterized by their diversity of rules, interacting processes and scale-dependent behavior. Models of these systems should capture both the multiplicity of rules as well as the scale transitions. Nowadays several examples are found: mental maps, fuzzy differential equations and hybrid intelligent tools. Spatial phenomena like territory transformation and land use in cities can be explained from this type of complex system.
- *Type-III* complex systems come in the scene when managing several perspectives towards particular common objectives of human agents. These agents interpret in different ways the dynamics of the system. Interpretations may produce consensus that is influenced by the emergent properties of the system in the long term. Several models that deal with this kind of complex behavior have been developed from decision theory.

- *Type-IV* complex systems exhibit macroscopic structures designed by societies to promote self-management. These systems are composed by diverse actors such as national governments, industries, local governments, universities among others. Part of the agents in the system often generate strategies focused in promoting the ultimate goal of controlling the system in several scales

When looking at the previous typification, complex adaptive systems may be included in type 3 or 4, however this observation does not discard that these systems can be studied from the other types of complexity. When approaching a complex adaptive system, models would be structured from the first two types of complexity so that they can be nurtured in the process from latter ones. Note that there is not particular emphasis regarding modeling tools in types III and IV. To develop these tools is currently an open problem.

3.2 *Modeling Complex-Adaptive Systems*

Constant adaptation in a socio-technical system compromises the controllability of its behavior given its unpredictable nature. Thus system dynamics theory can offer helpful insights about the modeling of these systems [34]. Models in this perspective are designed to understand the dynamics of the system rather to cope with its accurate prediction. Diverse epistemic perspectives should be taken into account when conceiving these models, transcending the purely scientific perspective [14].

Modeling complex adaptive systems demands today an understanding exercise that goes beyond the reductionist approach of traditional science. The whole is more than the sum of its parts is a well known premise in this quest and not necessarily the simplest explanations, tested under same conditions, are correct, challenging the Occam's razor [24]. The ultimate goal of modeling a system of this kind is to capture some of its properties in an artificial representation. That movement from reality to the representation depends on the intention of the modeler regarding the model [12, 30].

Modern science has encountered in simulation a valuable tool for dealing with complexity [12]. Simulation models of complex adaptive systems must approach complexity by being also complex. The idea that a reductionist simulation model exhibits the richness of a complex dynamics is indefensible today [33]. Therefore adaptations in a complex system should be transferred to the model, which is only possible as long as it is able of self-organization.

The process of modeling a complex adaptive systems requires a reduction of complexity without omitting essential components. In the end the model is still complex since the real system is so [38]. A minimal simulation model is necessary that preserves key elements of the system. Including additional components would not necessarily give useful knowledge, but surely increases the computational cost of the simulation. Therefore simplest complex models can be used to guide the initial discussions about the system, whereas more detailed models are preserved for a

posterior deeper analysis. To capture the entire footprint of complexity of a system in a simulation model with reasonable costs in time, effort and resources is today an open problem [6, 33, 37].

Transferring complexity from the system to the model is critical and invites to think about several questions: Is just one modeler able to transfer a good portion of the complexity from the system to the complex model? Should the transference of complexity be performed by a complex adaptive system composed by modelers and intelligent modeling tools? Which characteristics should this system have? Is it sufficient one complex adaptive system to model another one? Supposing the complexity of a system can be quantified, would not be necessary a more complex system to perform the modeling task? Some of these questions are partially considered in [3, 39].

Designing a complex system is a process guided by learning and evolution. It embodies a metaphor in which modeling agents interact collaboratively in the integration of micro-worlds (i.e. elements of the system) like children constructing a toy from constructions blocks [8]. Thus a complex system can be interpreted as a macro-world shaped by the ensemble of micro-worlds, which are more than being interconnected, they are entangled.

The process of configuring a simulation complex model can be delineated from the previous metaphor. Simulation micro-worlds are configured in order to structure simulation macro-worlds that shape the model of the complex-adaptive system. This is similar to the model building approach of systems dynamics [34], however micro-worlds a key differential element.

In models of complex-adaptive systems, micro-worlds can be configured as networks of agents or rules, however these networks can also be the result of self-organizing agents. Assign a purpose to self-organized networks is somehow fuzzy, but these may represent small associations or territories. Macro-worlds would emerge as the interconnection of smaller worlds reflecting a wider organization supported on interacting associations. As a result, it would be expected that emergent macro-worlds operate in several scales.

The simulation building blocks are based on the presence of cognitive agents [36], which represent human beings. Given their autonomy, plasticity and openness to be, establishing an guiding theory about human behavior is not just around the corner [19]. The way in which humans relate to technology they build is conditioned by their autonomy [2]. In the end, each agent can freely interpret the objects that are part of its world. Therefore, visualizing the consequences of this relationship would exceed any analytical effort.

Although capturing the whole complexity of a human being in a cognitive agent would be unsuccessful, this does not imply that to approach this goal can not be developed from some perspective. Complex system engineering as well as multi-agent systems methodologies can find inspiration regarding this task in different representations of the world, for example in [32] a combination of attachment theory and ant colony algorithms is proposed to study the properties of some human social systems. However, the purely scientific character given to the problem of representation is up to date a difficulty in human systems simulation [11, 36].

4 Fuzzy Worlds

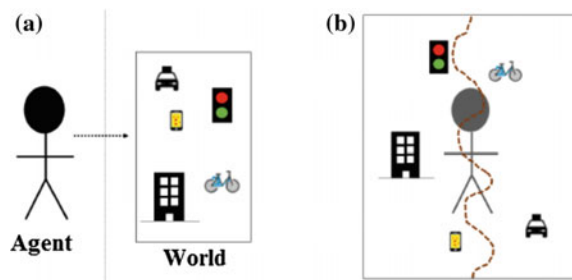
One of the fundamental aspects of agent design focused in human behavior is the problem of interaction. Communication is a key element in the interaction between humans and their organizations [19]. It requires a common ground where humans share interpretations of their world in terms of rules. In the design of a complex model using the building block metaphor, agents in micro-worlds may satisfy the condition of common ground. In a group of agents that share the same environment, each agent can interpret the context in a particular way thus its world acquires a different dimension regarding the other agents. Therefore the agent and the world are one.

The human agent may be represented as an evolving process [31] that relates rules and the surrounding field, instead of an independent element that interacts with its environment [11]. Thus there is no clear boundary between the agent and its world. Figure 1 presents a comparison between the classical vision of an agent (Cartesian) and an agent which is immerse in the world, called here as the Heideggerian perspective on an agent. In the classical perspective the boundary that separates the agent and its world is well defined, on the other hand the boundary is fuzzy in the Heideggerian perspective [23].

Immerse agents in a micro-world play a variety of roles in different moments, however these roles may be modified by new rules or relations between existent rules. Role changes would develop in a temporal scale, therefore the structure of the world that was consistent with existent roles gives way progressively to a new one. Adaptations in agents promote adaptations in the world. The agent is able of transforming its world, however the world strikes back by also transforming the agent. Thus talking about one necessarily refers to the other (i.e. fuzzy world).

If a modeling approach for complex adaptive systems following this perspective is attractive, emphasis would be given to micro-worlds rather than agents. This imposes a challenge to incorporate the self-interpretation of the human agent as a part of the model. According to [15] self-interpretation is given by the direct interaction with the world without any kind of mediation. Hence the traditional approach of artificial intelligence, where the agent has an internal representation of the knowledge of its desires and intentions expressed as a logic the agent uses to deliberately infer, must

Fig. 1 **a** Classical agent (Cartesian), there is a clear boundary between the agent and the world. **b** Immerse agent in the world (Heidggerian), the boundary between the agent and the world is fuzzy



be reformulated [36]. The simulation micro-world should be able of executing a self-modification according to perturbations induced by itself or other micro-worlds which is entangled with.

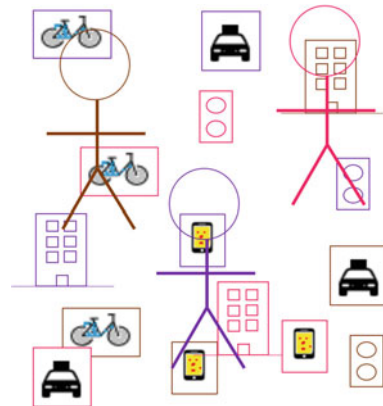
Should the conception of building blocks based on fuzzy worlds be developed as a purely scientific task or as a convergence field of different kinds of knowledge? To talk about a complex-adaptive system by itself is already imposing a particular way of looking at reality, since relations between agents are understood as flows of mater, energy or information. Giving this connotation implies to uncover the human dimension as something susceptible of being modeled and computationally represented. A purely scientific perspective to define a fuzzy micro-world would require guiding theories of a “computational human nature” which would be in the opposite direction of the opening to the be.

Although the scientific approach to fuzzy worlds would find a contradiction, the quest for modeling complex adaptive systems from this perspective should be faced in the aim to grasp something about these systems. The uncertainty regarding the comprehension of these systems can not be controlled, however to cope with is a possibility. The paradigm of fuzzy sets or its several ramifications (i.e. interval fuzzy sets, type-2 fuzzy sets, intuitionistic fuzzy sets, etc.) [20, 25] can be used to model the inherent uncertainty of a micro-world. Modeling a fuzzy world in this sense is a design exercise that should find inspiration in both experience and language.

Engineering can think about the modeling of a complex-adaptive systems as a design process that creatively and rigorously connects building blocks [21, 22]. Reality is susceptible to be captured not by fragmentation (i.e. analysis) but by aggregation (i.e. synthesis). This process is performed having in mind the necessity of a minimum representation of the complexity. Hence aggregation should be understood as the confection of an entanglement of fuzzy worlds, where scale should not be a problem since micro and macro worlds are mutually related.

A graphical representation of entangled fuzzy micro-worlds is depicted in Fig. 2. The vision of interaction (between agents) is replaced here by the vision of entan-

Fig. 2 Entangled fuzzy micro-worlds. Same pieces of reality can be interpreted in different ways. Each color (cyan,purple and brown) represents a particular way of interpretation



glement (of worlds). Entangled fuzzy micro-worlds are enriched by the diversity of interpretations about the human and technical phenomena that converge in the aggregation. Note the notion of complexity is extended here since the premise that the whole is more than the sum of its parts is debatable in the sense that the parts are not simple. Complex macro-worlds can emerge from the entanglement of fuzzy micro-worlds which is also complex. Hence this perspective of modeling is revealing that complexity would be scale-free attribute in complex-adaptive systems.

The building block as the entanglement of fuzzy micro-worlds embodies a paradox since it is an element which is at same time a complex phenomenon. It can be the size of a micro-world or several micro-worlds. The building block contains agents that interpret their own worlds, but also worlds that modify the agents. The same argument can be extrapolated to the construction of a macro-world from micro-worlds: the macro-world is shaped by micro-worlds but these are modified in the construction process. Figure 3 depicts the entangling process of macro-worlds W_A and W_B that are composed of several fuzzy micro-worlds (micro-worlds that share common attributes are represented by the same color). Notice the final entanglement produces a new fuzzy micro-world. This is possible since macro phenomena (i.e. structures, organizations, etc.) can introduce new rules or interpretations in the context of an existent micro-world.

The mental exercise around this entangling process evokes the phenomenology of the worldliness of the world [11, 17], paraphrasing: If a common structure for the sub-worlds of a world is discovered then the structure of the world has been found. If there is one common structure of the sub-worlds, this same structure should be



Fig. 3 The entangling process of macro-worlds W_A and W_B . Micro-worlds that share common characteristics are depicted with a particular color (orange,gray and black). In the end, the entanglement of macro-worlds produces a new micro-world (blue)

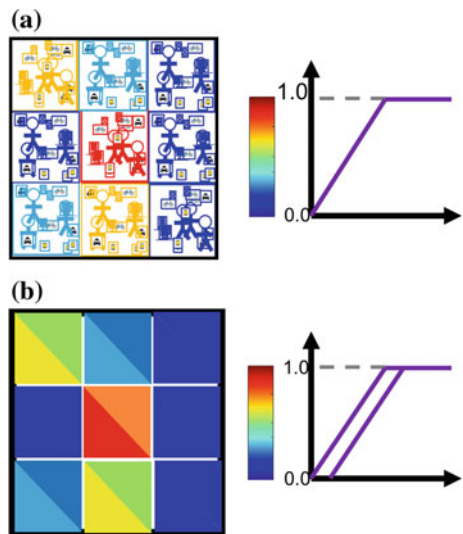
discovered in the world that contains these sub-worlds. This consideration allows to think about the structure of the world as self-similar no matter the scale.

One of the purposes of phenomenological analysis is to discover the structure of the world. Interpretation is in the heart of this analysis which is grounded on natural languages. If the paradigm of computing with words [24] is used to interpret the dynamics of sub-worlds in a complex-adaptive system, the model can be understood as a linguistic phenomenology of the system. In this context fuzzy worlds can be modeled by means of the perceptions and their relations given by one or several observers. Thus fuzzy sets and systems would be the natural representation mechanisms for fuzzy worlds.

A graphical view of two linguistic phenomenologies is depicted in Fig. 4. A macro world is considered where the word “high” is being interpreted from the analysis of a linguistic variable in the fuzzy worlds. Phenomenology represented in Fig. 4a makes use of a type-1 fuzzy set [20] to model the word. The set has been used to produce a heat map over the macro-world given some numerical observations surveyed from the worlds. The observer interprets the current state of the system is a kind of isotropic dissipation from the center to the corners. Another phenomenology is presented in Fig. 4b where the same word “high” is represented as an interval type-2 fuzzy set [25]. In this case linguistic phenomenology represents the interpretation of two observers about the system. Both agree that an isotropic dissipation is happening in the current state of the system, however one of them perceives a faster dissipation dynamics.

The fuzzy world concept can be considered as an ontological basis for modeling complex-adaptive systems grounded on phenomenological analysis. It finds in fuzzy set theory and fuzzy logic a natural epistemological and methodological tool to support the process of modeling. Although the fuzzy systems perspective has been used in the past to deal with complex systems, the ontological perspective introduced

Fig. 4 Linguistic phenomenology of a macro-world: **a** Captured by a type-1 fuzzy perception, **b** captured by an interval type-2 fuzzy perception



by fuzzy worlds points out a different path regarding the classical approach of agent-based modeling. Formalization of fuzzy worlds and its application to complex system engineering should be a matter of near-future discussions.

References

1. Ackoff, R.L., Gharajedaghi, J.: Reflections on systems and their models. *Syst. Res.* **13**(1), 13–23 (1996)
2. Arendt, H.: *The Human Condition*. University Press, Chicago (1989)
3. Arnold, T.R.: Procedural knowledge for integrated modelling: towards the modelling playground. *Environ. Model. Softw.* **39**, 135–148 (2013)
4. Bak, P., Chen, K.: Self-organized criticality. *Sci. Am.* **264**(1), 46–53 (1991)
5. Bar-Yam, Y.: *Dynamics of Complex Systems*. Addison Wesley Longman, Reading, Massachusetts (1997)
6. Berger, T., Birner, R., Díaz, J., McCarthy, N., Wittmer, H.: Capturing the complexity of water uses and water users within a multi-agent framework. In: Craswell, E., Bonnell, M., Bossio, D., Demuth, S., Van De Giesen, N. (eds.) *Integrated Assessment of Water Resources and Global Change: A North-South Analysis*, pp. 129–148. Springer, Dordrecht (2007)
7. Bettencourt, L.: The origins of scaling in cities. *Science* **340**(6139), 1438–1441 (2013)
8. Cowan, F.S., Allen, J.K., Mistree, F.: Functional modelling in engineering design: a perspectival approach featuring living systems theory. *Syst. Res. Behav. Sci.* **23**(3), 365–381 (2006)
9. Dopfer, K.: The economic agent as rule maker and rule user: Homo Sapiens Oeconomicus. *J. Evol. Econ.* **14**(2), 177–195 (2004)
10. Dopfer, K.: Evolutionary economics : a theoretical framework. In: *The Evolutionary Foundations of Economics*, p. 5 (2005)
11. Dreyfus, H.: *Being-in-the-World: A Commentary on Heidegger's Being and Time*, Division I. Bradford Book, London, UK (1990)
12. Epstein, J.: Why model? *J. Artif. Soc. Soc. Simul.* **11**(4), 6 (2008)
13. Epstein, J.M.: *Generative Social Science* (2006)
14. Gadamer, H.G.: Notes on planning for the future. *Daedalus* **95**(2), 572–589 (1966)
15. Heidegger, M.: The question concerning technology. In: *The Question Concerning Technology and other essays*, chap. 1, pp. 4–35. Garland publishing (1977)
16. Heidegger, M.: *Being and Time* (1953), 2nd edn. SUNY Press (2010)
17. Heidegger, M., Grene, M.: The age of the world view. *Boundary 2* **4**(2), 341–355 (1976)
18. Holland, J.H.: Complex adaptive systems. *Daedalus* **121**(1), 17–30 (1992)
19. Jelinek, M., Romme, A.G.L., Boland, R.J.: Introduction to the special issue: organization studies as a science for design: creating collaborative artifacts and research. *Organ. Stud.* **29**(3), 317–329 (2008)
20. Klir, G., Yuan, B.: *Fuzzy Sets and Fuzzy Logic*. Prentice Hall, New Jersey (1995)
21. Kroes, P.: Engineering design. In: *Technical Artefacts: Creations of Mind and Matter*, pp. 127–161. Springer, Heidelberg (2012)
22. Kroes, P., Franssen, M., Van De Poel, I., Ottens, M.: Treating socio-technical systems as engineering systems : some conceptual problems. *Syst. Res. Behav. Sci.* **814**, 803–815 (2006)
23. Melgarejo, M., Obregon, N.: Diseño de modelos complejos para la simulación de sistemas socio-técnicos. *Educación y humanismo* **19**(33) (2017)
24. Mendel, J.: Computing with words: Zadeh, Turing, Popper and Occam. *IEEE Comput. Intell. Mag.* **2**(4), 10–17 (2007)
25. Mendel, J.M.: *Uncertain Rule-Based Fuzzy Systems*. Springer, Heidelberg (2017)
26. Mitchell, M.: *Complexity: A Guided Tour*. Oxford University Press (2009)
27. Nicolis, G., Nicolis, C.: *Foundations of Complex Systems Nonlinear Dynamics, Statistical Physics, Information and Prediction*. World Scientific Publishing Co., London, UK (2007)

28. Nicolis, G., Nicolis, C.: Foundations of complex systems. *Eur. Rev.* **17**, 237 (2009)
29. Olaya, C., Gómez-quintero, J., Salas, D.: Ontology in action : urban mobility as evolving knowledge. In: 24th Annual Conference of the European Association for Evolutionary Political Economy. Cracow, Poland (2012)
30. Pahl-Wostl, C.: The implications of complexity for integrated resources management. *Environ. Model. Softw.* **22**(5), 561–569 (2007)
31. Rescher, N.: Process philosophy. In: Zalta, E. (ed.) *The Stanford Encyclopedia of Philosophy* (2008)
32. Riveros Varela, C.A., Beltran Velandia, F., Melgarejo Rey, M.A., Gonzalez Romero, N., Obregon Neira, N.: Foraging multi-agent system simulation based on attachment theory. In: Sanayei, A., Rössler, O.E., Zelinka, I. (eds.) *ISCS 2014: Interdisciplinary Symposium on Complex Systems*, pp. 359–364. Springer International Publishing, Cham (2015)
33. Rzevski, G.: Modelling large complex systems using multi-agent technology. In: 13th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, pp. 434–437 (2012)
34. Schwaninger, M., Ambroz, K., Olaya, C.: The complexity challenge: a case for model-based management . In: *Proceedings of the 2007 International Conference of the System Dynamics Society*, pp. 1–29 (2007)
35. Sice, P., French, I.: A holistic frame-of-reference for modelling social systems. *Kybernetes* **35**(6), 851–864 (2006)
36. Sun, R.: Cognitive science meets multi-agent systems: a prolegomenon. *Philos. Psychol.* **14**(1), 5–28 (2001)
37. Torrens, P.M., Nara, A.: Modeling gentrification dynamics: a hybrid approach. *Comput. Environ. Urban Syst.* **31**(3), 337–361 (2007)
38. Van Delden, H., Seppelt, R., White, R., Jakeman, A.J.: A methodology for the design and development of integrated models for policy support. *Environ. Model. Softw.* **26**(3), 266–279 (2011)
39. Voinov, A., Bousquet, F.: Modelling with stakeholders. *Environ. Model. Softw.* **25**(11), 1268–1281 (2010)
40. Wolfram, S.: *A New Kind of Science*. Wolfram Media Inc (2002)

Procedural Generation of Levels for the Angry Birds Videogame Using Evolutionary Computation



Jaime Salinas-Hernández and Mario Garcia-Valdez

Abstract This paper consisted in the generation of an evolutionary computation-based system capable of generate and evolve the structures of which one level of the Angry Birds game is composed, these structures are evaluated according to the stability of the structure as well as for the complexity of said structure. For this a level generation system was designed based on the data obtained from the game, said data consist in the number and type of pieces that appear and the applied gravity of the game, the individuals of the group are evaluated by a simulation of the generated level and then checking how the structures are affected by the gravity of the game. The evolutionary computation system as the main objective of generating structures based on the existent pieces of the game and evolving said pieces by combining them in a process that simulates the rules of the Open-Ended Evolution algorithm in which the evolution of this compounds is not inclined towards a numeric objective rather than to extend the diversity from which the pieces may be selected for a level.

Keywords Genetic algorithm · Procedural generated content · Open-ended evolution · Evolutionary computation

1 Introduction

Procedural content generation is an interesting subject on the entertainment area, primarily in the video game industry because of the easiness and quickness provided to generate content for different video game projects, in this paper a evolutionary computation-based system is proposed to generate complex content for videogames based on an evolutionary algorithm and open-ended- evolution.

For this paper a procedural generation content system is proposed for the game angry birds, angry birds is a game developed by Rovio Entertainment in which the

J. Salinas-Hernández · M. Garcia-Valdez (✉)
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: mariosky@gmail.com

J. Salinas-Hernández
e-mail: salinas.jaime1217@gmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_40

objective is to destroy structures placed on the map as well as eliminating green pig-like entities in order to obtain the highest score possible, the game mechanics is to use a slingshot to shoot a certain number of birds with different abilities to destroy the structures and the objectives [1].

While it is possible to create an evolutionary agent to play the levels and obtain the best score the main objective of this paper is to generate the levels that can be played by the agents or physical players, the levels themselves need to be complex in the sense that interesting structures can be created but it has to be possible to complete in some way [2, 3].

In Sect. 2, a background of the subjects used on this paper will be presented, this subjects, while not all being used on by different authors on their works provide a great help on the development of the proposed system, Sect. 3, contains a quick overview of some aspects used by different authors in order to tackle this problem, their design and basic functionality, in Sect. 4 the description of the current state of the generated system is explained, from the basis of the development to the different methods that have been used to progress on the different sections of the system, in Sect. 5 we explain the future work of the paper, where we are in the development phase and how the future implementations are to further improve the paper as well as the conclusions that we reached.

2 Background

In this Section we provide a quick overview of the literature about the areas that provide the basis to the development of the current paper, this section is divided in three parts, the first one gives us an explanation of what procedural content generation entails, the second one provides an explanation on genetic algorithms and how can they be used to solve optimization problems, and finally, the third one provides us with an overview of Open-Ended Evolution and what it means to have an Open-Ended Evolutionary algorithm.

2.1 *Procedural Content Generation*

Procedural content generation (PCG) denotes the way to automatically generate content by algorithms instead of generating the same content by manual means such as with different personnel.

In the videogame industry many developers use a PCG style software that produces raw images and designs used by other personnel to improve it and use it in order to reduce the development cycles, and the productions cost [4]. Nowadays there are game developer of AAA games that use software to auto generate some elements of the environment such as vegetation of terrain layout specially for open-world games,

this in turn cuts a great amount of development time for said games [5, 6]. PCG as six focus areas listed below

PCG as six focus areas listed as follows:

- **Level Design:** This aspect involves only the design of specific sections of a game, primarily those in which the player is able to interact.
- **Audio:** This aspect is focused on designing soundtracks that are able to react to a player specific action.
- **Visuals:** This area is focused on creating or improving visual representations of objects on games such as giving raw designs of players faces of improving the images of other graphics.
- **Narrative:** This aspect is focused on the generation of readable and coherent stories or plotlines of games.
- **Gameplay:** This aspect uses agents to test and evaluate the mechanic on which a game is based by trying to play as a human would in order to improve said mechanics.
- **Game Design:** This aspect is focused on generating the rulesets and mechanics of a new game [7–9].

2.2 Genetic Algorithm

Is a method developed by John Holland on 1975 [10] the genetic algorithm (GA) is an optimization and search technique based on the principles of genetic and natural selection, it allows a population composed of many individuals, which by themselves represent a possible solution to the problem, to evolve under specific selection rules in order to obtain the maximum fitness.

In other words a genetic algorithm is a method to solve optimization problems by creating a population that will represent a possible solution and repeatedly doing genetic operations as selection, crossover and mutation on each member of the population in order to generate the members of the population for the next generation that would be closer to the best solution of the problem than their parents, the cycles in this method where the operations are made are called generations and by moving over this the population “evolves” o in other words, they get closer to the best possible solution to a specific problem.

The main life cycle of a genetic algorithm shown of Fig. 1 is described as follows:

1. Initialize a random n population of values suitable for the solution.
2. Evaluate the fitness of each individual (solution) in the population.
3. Check if the stopping criteria has been met.
4. If not.
 - a. Introduce new individuals to the population by crossover operations.
 - b. Mutate a random value of the new individuals.
 - c. Select new cross over individuals according to their fitness for the next generation.

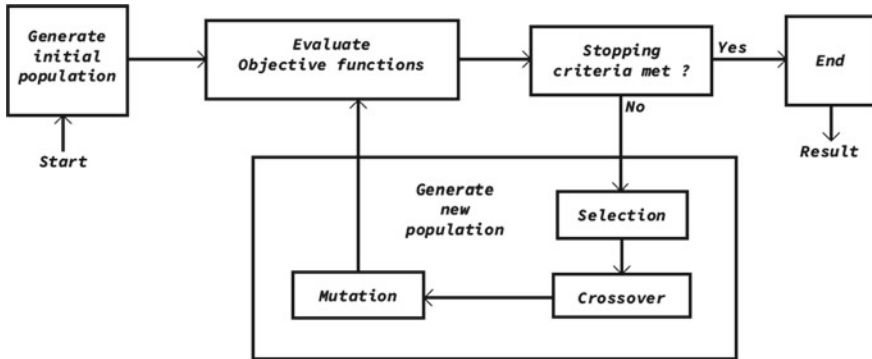


Fig. 1 Life cycle of a genetic algorithm

5. Loop from step 2 until an end condition is satisfied.
6. Select the best individual of the population as the solution for the problem.

2.3 Open-Ended Evolution

Open-ended evolution (OEE) is a variant of genetic evolution algorithm in which the objective differs from the regular genetic algorithm one, in this case the final objective is not to reach a final state of the evolution but to continue evolving and creating new types of entities more complex on each new generation, these same entities have the possibility to evolve not only inside the frame of their type but create entirely new groups [11].

Another explanation provided by Taylor et al. [12, 13] defines OEE has a system capable of creating novelty beyond a point where nothing changes on the current group of interest rather than just converging to a stable or quasi-stable state.

For this paper OEE is defined simply has a process of genetic evolution that encapsulates a group of similar processes and that is able to increase their number by the more complex the entities become.

3 Overview of Existing Work

In this Section we present some of the different approaches to the procedural generation of content for the angry birds video game.

Kaidan et al. [14] propose a model in which the generation of content is controlled by the skill of a player while actively playing the game, the moment the player finishes a certain level the algorithm obtains the score and generates the next level to be played. The generation of the levels is based on a formula which take into account

the average velocities to evaluate the levels however the generation of the levels did not take into account the number of birds and pigs placed on the level and as such the levels were able to be cleared with ease by the participants.

Renz et al. [15] created an article in which the angry birds level generation contest mechanics of the IEEE Conference on Computational Intelligence and Games are based on, it explains some of the solving methods proposed by previous competitors. Here it is explained that there are two tracks of the angry birds competition, the first one is focused on training agents capable of playing the game as well as a human could, taking into consideration the angle of the shots and the movement or destruction of the pieces created with that shot, the second track of the competition is based on the levels themselves in which a system can automatically create levels that are interesting to a human player, playable as well as difficult able to be completed with a given amount of birds.

Jiang et al. [16] proposed the use of predefined letter style patterns and combined to create a small pool of words used to be combined in order to create levels with text on them, the participants would play the game and after that they would be asked if they liked the layout of the levels, the way the requirements for the competition were fulfilled were by creating the letter structures as stable as possible, use the provided layered structures, generating a said number of usable birds and pigs by using an heuristic calculation.

4 Current Approach

In this Section we describe the approach that was used for the development of this paper, first presenting a quick overview of the integration of a genetic algorithm with the combination of pieces, the way the open-ended segment of the system handled the creation of new compounds for the pool of pieces and then an explanation on the way the individuals are evaluated.

The game contains a total of 11 pieces as shown on Fig. 2 that cannot be modified, it is not possible to add more or modify the existing ones, with this in mind the purpose is to create bundles of this same pieces in different locations and angles in order to create structures that can be used as building blocks for more complex levels, in order to do this a sequence of events needs to be done as follows:

1. Generate the first generation of the population
2. Run a Simulation
3. Evaluate each member of the population
4. If a member of the population has remaining pieces obtain the location and angle of the pieces, create a bundle and add it to the pool
5. Select the parents for the next generation
6. Do the crossover and mutation operations
7. Repeat the cycle.



Fig. 2 List of pieces in the angry birds game

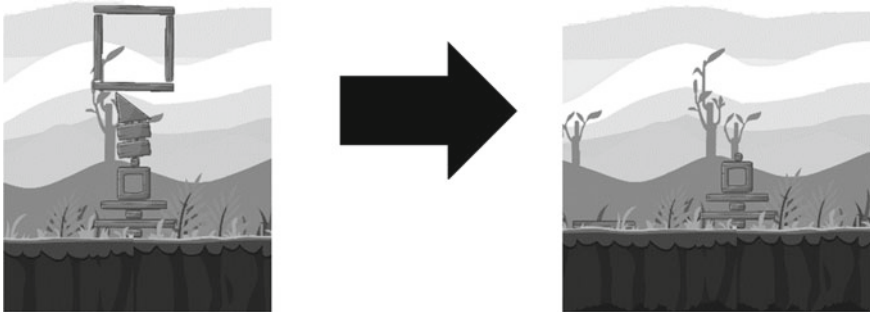


Fig. 3 Structure at beginning and end of a simulation

Since the algorithm has to generate more complex structures the remaining pieces of a member of the population have to be used has a starting point, this new additions to the piece pool are added to the population by the mutation operation, the objective in the first phase of the system is to generate structures that are capable of maintaining their own balance as shown on Fig. 3, since the objective of open-ended evolution for this system is to add new bundles to the pool of options the algorithm will first run a simulation with a group of pieces and after it the ones that were not destroyed by falling because of gravity will be saved and added to the pool of options.

In order to add a bundle of pieces as a new component first it has to be measured, an imaginary bounding box is defined by calculating the lengths of all the pieces, finding the borders of said pieces and find the points that could define a box around all of them, calculate the height and width then calculate the center point of said box and calculate the offsets from the center of the group to the center of each respective piece as shown on Fig. 4, this information is needed in order to add it as a new element so it can be used to place the pieces in their respective new positions when the files needed for the simulation are being created.

In case one or more of this pieces is rotated or needs to be rotated we defined a function that first obtains the data of a each piece in the group, finds the 4 edges of it and then if the piece needs a rotation different than 0 degrees a function to rotate the figure is called, the function returns a list of points that represent the borders of the rotated piece, after this the other pieces are processed the same way and with the points we calculate the entire bounding box as well as the center of the group that will be added.

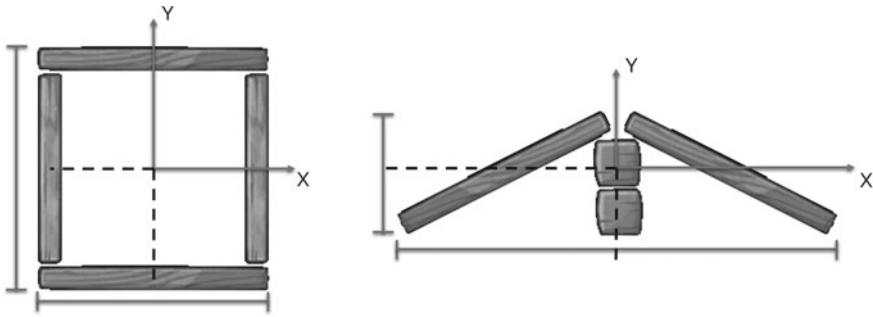


Fig. 4 bounding box calculation

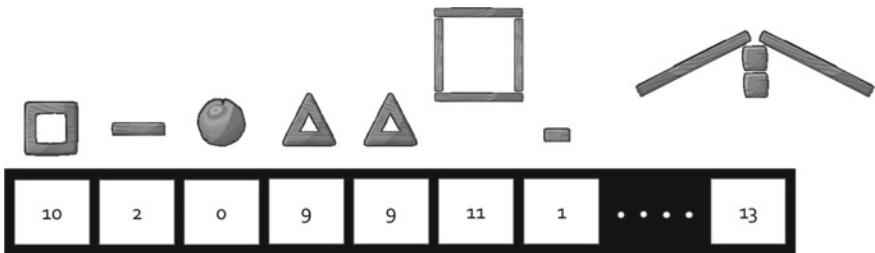


Fig. 5 Chromosome chain of an individual

In order to create a level generation algorithm there has to be a way to represent the level structure in a way that the genetic algorithm is able to interpret it and at the same time be able to be interpreted by the game itself [17], so in this case the way to represent the elements of a member of the population is by using the id of each piece and the new added ones has a pointer to the respective data that needs to be obtained, this way and individual can be represented as shown on Fig. 5 where each chromosome represents a certain item in the pool of pieces, the numbers for this elements are assigned in a first come first served basis, the numbers are infinitely incremented and this same pieces can be repeated on each individual of the population.

Once the pieces have been added and the individuals are ready to be evaluated a simulation on the game environment is scheduled by order in the population, all the individuals have a maximum of ten seconds to maintain balance or reach absolute stability, if absolute stability is reached it means one of two things, first the generated structure is completely stable or second, the tower fell to either side and the pieces were destroyed or stopped moving by gravitational pull, when either of this happens by more than two seconds the simulation ends regardless of time remaining and the next individual is simulated.

The evaluation process for this paper is still in the development stages, according to another work by Yannakakis et al. [18] there are various ways to evaluate the generated content, one example of this is an adaptive generation to optimize the

player experience [19], however in order to check that the evolution in this current system is working has it should a criteria has been temporarily made as follows, each individual in the population has 100 points so before the simulation starts each of them are viable candidates to be an elite member of the population regardless relative height or complexity, then the remaining of the process is separated in three phases has follows.

First, after each of the individual simulations are over the evaluation process takes place, in this phase the resulting level is checked for pieces, each one of them is counted and then the resulting number is compared to that of the initial quantity of pieces that were originally put in before the simulation in order to calculate the first element of the evaluation process, each piece represents a certain percentage of 100% so the missing pieces are subtracted from the total.

Second, after removing percentage according to the missing pieces an individual remaining pieces are checked for their positions in comparison to the original ones before the simulation, using the formula 1 each piece within the individual is evaluated according to the center point of the piece, if the piece moved more than a certain threshold half the percentage for one piece is removed from the remaining percentage, after this using the formula 2 the position of the piece is further checked this time by the difference in angle from the original one, in case the difference of angles is more than a certain threshold half a piece value is further removed from the total.

$$error_{xy} = \begin{cases} 0.1 & \text{if } 0.08 > d, \\ \frac{100}{length_pieces} * 0.5 & \text{if } 0.08 < d. \end{cases} \quad (1)$$

$$\text{where : } d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

$$error_r = \begin{cases} 0 & \text{if } -5 \leq r \leq 5, \\ \frac{100}{length_pieces} * 0.5 & \text{if } r < -5 \text{ or } r > 5. \end{cases} \quad (2)$$

$$\text{where : } r = |rotation_0| - |rotation_f|$$

Finally, the total height of the remaining group of pieces is calculated by combining the bonding boxes of all pieces, then the individuals are ordered by the obtained percentage and then further ordered by the total height, this way the most balanced ones will have the best probability to be selected together for a crossover operation to create the next generation.

Using the proposed approach a modification to the generation of individuals was developed, this modification allowed the generation to create structures that are able to maintain balance except for some cases where some element of the individual where irregular pieces, this pieces like triangles and circles when placed in positions between the base of the tower to before the top caused the tower to fall depending if the subsequent pieces in the individual where able to outbalance the structure.

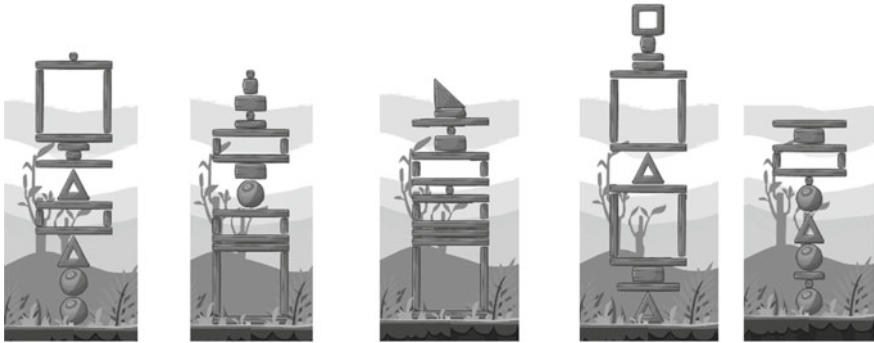


Fig. 6 Result samples of current system

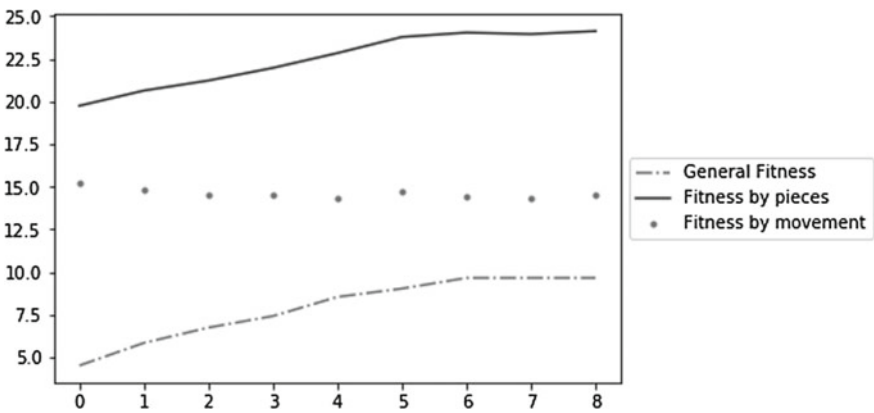


Fig. 7 Results with current approach

With this in consideration the results as shown in Fig. 6 and subsequently Fig. 7 where able to be obtained, however since them objective of the generated system is to construct more complex structures the obtained results will be used as a base for the further development of the system with the implementation of the concepts presented in the future work section.

5 Conclusions and Future Work

The use of artificial intelligence in videogames is a common thing almost since before the first generation of videogames on 1972 as simply entities that would follow a set group of actions but as the generation kept advancing the need for more intelligent entities appeared, as of this generation some videogames use AI as ways to improve the interaction between man and game, but also using it in order to facilitate the way

videogames are created, so for this system the focus of using evolutionary algorithms as great interest, as a starting point a rather simple game can be used as a workbench of the work but it also helps us define how much more can be done when all the required objects are obtained, improve on the same system and find ways for it to be used on different other games.

Since there is a need to create complex structures than can cover a great section of the play map, a method to evaluate the distribution is needed, for this instance we propose using the rule of thirds [20] which is a guideline used while creating an image in which the purpose is to create aesthetical pleasant images where the focus point or the most important area is at the center of the image as shown on Fig. 8, using this rule different distribution groups or masks are created has shown on Fig. 9 in order to use them before the simulation of an individual takes place in order to distribute the pieces in a way that can cover more area and create more balanced levels instead of using a tower like distribution as shown on Fig. 10.

Using this same distributions as a base, the masks that will be used on the individuals to balance the content in the level will be defined the example below where a value of 1 represents an area of the level that needs to have pieces and 0 represents sections were no piece will be added.

```

type_castle = [[0,0,0,1,0,0,0],
               [1,0,1,1,1,0,1],
               [1,1,1,1,1,1,1]]
type_house  = [[0,0,1,1,0,0,0],
               [0,1,1,1,1,0,0],
               [0,1,1,1,1,0,0]]

```

Using this same distribution a new evaluation will be added in which according to how much of the area is filled a score will be given and this score will be used as a new sorting value at the moment of selecting individuals for the crossover operation, since as shown on Fig. 8 there is a possibility that a huge mask is given to an individual it can be possible that the amount of pieces will not be able to cover the full area given, so in order to prevent this kind of problems we also propose to give each mask a value that refers to the minimum amount of pieces and height required to fill the most of the mask then check the amount of pieces and height before simulation for a single individual and then randomize which mask of the ones that can be used will be assigned to that individual.

Another proposed method for the evolution of the individuals is to use a mutation operation that is capable of adding or removing pieces from an individual, this way the diversity of the population will be greater on future generations, also by using the same mutation idea a change on the mask assigned to a particular individual can be modified as well, in the case a mask is either too small or too big for an individual because of the orientation of the pieces a precaution has been created, in case the mask is too small for the number of pieces after filling the mask areas the code will take the remaining pieces and begin placing them as if the mask was repeated, and in the case the number of pieces is too small the system will simply return the pieces that were able to be fitted in and continue with the process as usual.



Fig. 8 Rule of thirds example

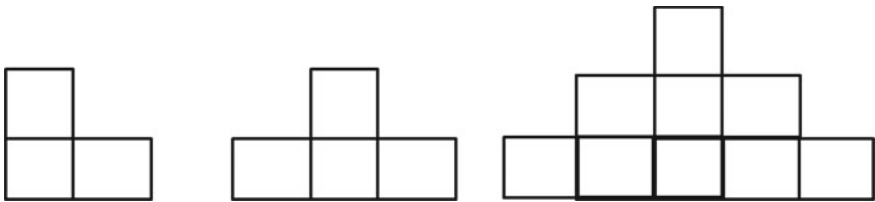


Fig. 9 Masks examples with the rule of thirds

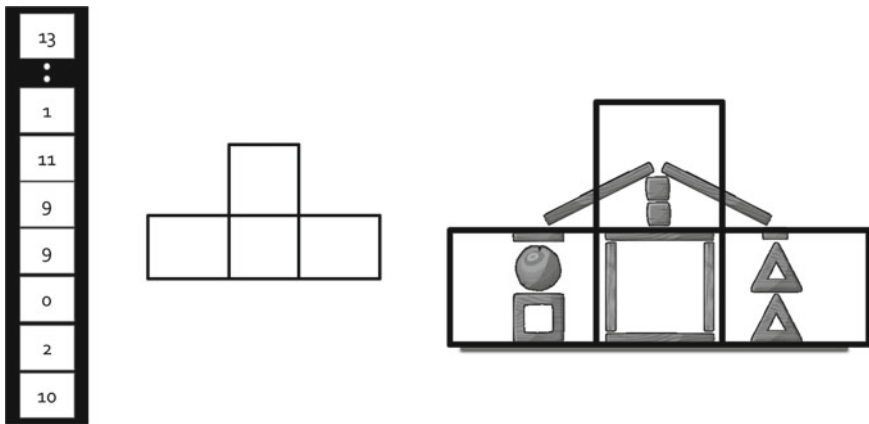


Fig. 10 Rule of thirds applied to an individual

References

1. Rovio Entertainment Corporation. *Angry Birds* (2009)
2. Stephenson, M.J.B., Renz, J., Ge, X., Ferreira, L.N., Togelius, J., Zhang, P.: The 2017 AIBIRDS Level Generation Competition. *IEEE Trans. Games*, 1–1 (2018)
3. Stephenson, M., Renz, J., Ferreira, L., Togelius, J.: Competitions—IEEE Conference on Computational Intelligence and Games, 14–17 August 2018, Maastricht (The Netherlands)
4. Togelius, J., Shaker, N., Nelson, M.J.: Introduction. In: Shaker, N., Togelius, J., Nelson, M.J. (eds.) *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, pp. 1–15. Springer, Heidelberg (2016)
5. Togelius, J., Yannakakis, G.N., Stanley, K.O., Browne, C.: Search-based procedural content generation: a taxonomy and survey. *IEEE Trans. Comput. Intell. AI Games* **3**, 172–186 (2011)
6. Smith, G., Othenin-Girard, A., Whitehead, J., Wardrip-Fruin, N.: *PCG-Based Game Design: Creating Endless Web*, tech. rep. (2012)
7. Yannakakis, G.N., Togelius, J.: *Artificial Intelligence and Games (First Public Draft)*, p. 359 (2017)
8. Smith, G.: Understanding procedural content generation. In: *Proceedings of the 32nd Annual ACM Conference on Human Factors in Computing Systems - CHI '14*, pp. 917–926. ACM Press, New York (2014)
9. Shaker, N., Togelius, J., Nelson, M.J.: *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*. Springer, Heidelberg (2016)
10. Holland, J.H.: Adaptation in natural and artificial systems. In: *Sgart Newsletter* (1975)
11. Standish, R.K.: Open-ended artificial evolution. *Int. J. Comput. Intell. Appl.* **03**, 167–175 (2003)
12. Taylor, T., Bedau, M., Channon, A., Ackley, D., Banzhaf, W., Beslon, G., Dolson, E., Froese, T., Hickinbotham, S., Ikegami, T., McMullin, B., Packard, N., Rasmussen, S., Virgo, N., Agmon, E., Clark, E., McGregor, S., Ofria, C., Ropella, G., Spector, L., Stanley, K.O., Stanton, A., Timperley, C., Vostinar, A., Wiser, M.: Open-ended evolution: perspectives from the OEE workshop in York. *Artif. Life* **22**, 408–423 (2016)
13. Taylor, T.: Evolutionary innovations and where to find them: routes to open-ended evolution in natural and artificial systems. In: *Artificial Life*, vol. 25 (2018)
14. Kaidan, M., Chu, C.Y., Harada, T., Thawonmas, R.: Procedural generation of angry birds levels that adapt to the player's skills using genetic algorithm. In: *2015 IEEE 4th Global Conference on Consumer Electronics (GCCE)*, pp. 535–536. IEEE, Oct 2015
15. Renz, J., Ge, X., Verma, R., Zhang, P.: Angry birds as a challenge for artificial intelligence. In: *Thirtieth AAAI Conference on Artificial Intelligence*, Mar 2016
16. Jiang, Y., Harada, T., Thawonmas, R.: Procedural generation of angry birds fun levels using pattern-struct and preset-model. In: *IEEE Conference on Computational Intelligence and Games* (2017)
17. Togelius, J., Shaker, N., Nelson, M.J.: Representations for search-based methods. In: Shaker, N., Togelius, J., Nelson, M.J. (eds.) *Procedural Content Generation in Games: A Textbook and an Overview of Current Research*, pp. 159–179. Springer, Heidelberg (2016)
18. Yannakakis, G.N., Togelius, J.: Experience-driven procedural content generation. *IEEE Trans. Affect. Comput.* **2**, 147–161 (2011)
19. Shaker, N., Yannakakis, G., Togelius, J.: Towards automatic personalized content generation for platform games. In: *Sixth Artificial Intelligence and Interactive Digital Entertainment Conference*, Oct 2010
20. Rowse, D.: Rule of thirds. In: *Digital Photography School* (2012)

A Multi-agent Environment Acting as a Personal Tourist Guide



Asya Stoyanova-Doycheva, Todorka Glushkova, Vanya Ivanova,
Lyubka Doukovska and Stanimir Stoyanov

Abstract Despite Bulgaria's unique geographical location and its possession of one of the oldest cultures in Europe and the world, our cultural heritage is relatively unknown outside the country. At present, modern information and communication technologies are not used in their full potential for its promotion and advertising. This article presents an intelligent tourist guide which can support tourists during their visits to cultural and historical sites. Furthermore, the guide takes into account various factors, such as the tourists' preferences, location, time available, and the presence and location of cultural and historical objects in the area, in order to propose virtual or real cultural and historical routes. Tourists can choose one of the two available options to implement a route—a virtual tour or a real walk. A typical scenario demonstrates the usage of the tourist guide.

Keywords Tourist guide · Intelligent agents · Ontologies · Ambient-oriented modeling · IoT applications

A. Stoyanova-Doycheva · T. Glushkova · V. Ivanova · S. Stoyanov
Paisii Hilendarski University, 4000 Plovdiv, Bulgaria
e-mail: astoyanova@uni-plovdiv.net

T. Glushkova
e-mail: glushkova@uni-plovdiv.bg

V. Ivanova
e-mail: vantod@uni-plovdiv.bg

S. Stoyanov
e-mail: stani@uni-plovdiv.net

L. Doukovska (✉) · S. Stoyanov
Institute of Information and Communication Technologies, Bulgarian Academy of Sciences,
1113 Sofia, Bulgaria
e-mail: doukovska@iit.bas.bg

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_41

1 Introduction

Due to its unique geographical location, the Balkan region used to be a crossroads for many nationalities for thousands of years. It was a cradle for one of the oldest cultures in Europe and the world. This enormous cultural heritage is little known outside Bulgaria and modern information and communication technologies have not been used in their full potential for its promotion and advertising. The Balkan region and Bulgaria in particular are an interesting tourist destination. For the digital presentation of our cultural and historical heritage, an ancestor of the tourist guide described here, known as the BECC (Bulgarian Electronic Cataloguing Cultural) environment, was developed more than ten years ago [1]. In BECC, the cultural and historical objects are presented according to the CCO standard [2].

The project is currently being updated in connection with the development of an e-learning environment called ViPS (Virtual Physical Space) [3]. ViPS is being implemented as a reference architecture that can be adapted to various CPSS-like applications such as a smart seaside city [4] or a smart tourist guide [5]. Cyber-Physical Spaces (CPS) and Internet of Things (IoT) are closely related concepts. Despite certain differences, the integration of the virtual and the physical worlds is common to both of them. By placing the person (the user) in the center of such spaces, they become Cyber-Physical-Social Spaces (CPSS). In terms of software architecture, CPSS include a variety of components designed to provide effective support to different user groups, taking into account changes in the physical environment.

Recently, using the results of previous developments, we have been working on an intelligent guide that takes into account various factors, such as tourists' preferences, location, time available, and the presence and location of cultural and historical objects in the area, in order to offer virtual or real cultural and historical routes. Additionally, the tourist guide will be able to direct and advise the tourist when following the route. The personal Tourist Guide (TG) is implemented as a multi-agent environment.

This publication provides an overview of the TG and demonstrates its use in a scenario. Section 2 presents a brief overview of selected related works. The functionality of the guide is explained in Sect. 3. The general architecture of the tourist guide is introduced in Sect. 4. Section 5 renders the kernel of the guide's server, known as CHH OntoNet. The next section looks at a typical scenario for using the tourist guide. Finally, the last section of the article summarizes briefly the current state of implementation and presents some ideas for the future expansion of the TG.

2 The Related Works

The CPSS and IoT approaches reveal new, unprecedented possibilities to support tourists. It is becoming increasingly common for a tourist to be able to activate a wide range of services from their own mobile phone. The mobile tourism domain

knows a new generation of mobile tourist applications that guide tourists in their travels, to provide all the necessary information about local tourist attractions of a region, hotels, restaurants, and online reservations.

The paper [6] presents a system known as the Tourist Assistant (TAIS) to support tourists in a region. The system analyzes the user's actions to reveal the user's preferences and it recommends cultural heritage objects based on the information received and the current situation in the region. It also provides tourists with transportation options to reach the cultural heritage. The information about a cultural object combines texts, images, and videos extracted from accessible Internet sources. The TAIS offers a range of intelligent services that support semantic data exchange using appropriate ontologies. The physical space, in which the tourist is located and modeled as a cultural space. The objects of the cultural heritage provided to tourists are listed by means of a recommendation service implementing a collaborative filtering technique. Mobile city guides operating on their mobile devices can effectively assist tourists in an unknown location. Currently, specialized literature presents a variety of such systems providing customized tour recommendations to help tourists make feasible plans and visit sites of interest within the time available. However, existing tour generators consider available attractions only as sites that do not have physical locations.

A different approach is presented in [7]. The approach draws attention to the fact that tourists usually walk through pedestrian zones, shopping areas or urban regions of architectural, cultural and scenic value rather than just visiting restricted areas or taking the fastest route to city attractions. As a result, a context-aware mobile city guide for Athens providing personalized tour planning services for tourists is implemented and described in this contribution. The city guide includes walking routes and thus supports a more experienced survey of tourist destinations. As expected, this wider acceptance of tourist attractions greatly increases the complexity of modeling the related optimization problem.

A recommender system that provides personalized information about locations of potential interest to tourists is presented in [8]. The system generates suggestions, consisting of tourist sites, according to the current position and historical data describing the tourist movements. For the selection of tourist sites, the system uses a set of points of interest identified a priori. We evaluate our system on two datasets: real and synthetic, both storing trajectories, which describe previous movements of tourists. The proposed solution is highly feasible and the results show that the solution is both efficient and viable. Regardless of the use of mobile devices in online mode, the information is not always relevant to the user and does not meet their needs at any time and any place. For this reason, a set of mobile technologies and methods is discussed in [9], which aim to achieve an approach to m-tourism providing the user with tourist recommendations based on its context and preferences to meet their needs.

We are now aware that tourists generally spend a lot of time planning their trips because they have to make the most of every moment. In this sense, new technologies offer unlimited route planning capabilities that can be adapted to unforeseen circumstances occurring during the journey. The paper [10] describes a location independent

plan consultant implementing the Hill Climbing search algorithm. As examples, the application has been tested for the capitals—Rome and Ankara. In paper [11] an application is presented that aims to help tourists better understand the cities they visit, even in the absence of local information or a specialized tour guide.

Many articles are dedicated to special groups of tourists or special types of tourism; for example, in combination with health care and spa procedures or wine and food tastings. A proposal to present tourist and cultural information to children through an augmented-reality approach is discussed in [12]. To focus the attention of young tourists, a virtual tourist guide is used that appears as a comic book superhero. In order to achieve a realistic representation of the avatar, a technique for fast simulating talking heads is explored. In addition, an Android prototype is presented, which shows the effectiveness of the approach to enhancing children's learning. A recommendation system for collaborative ayurvedic treatment and tourism is described in [13]. In the proposed model, keywords, feedback, reviews, and profiles are effectively used to determine user preferences. Personalized recommendations for preferred ayurvedic treatment locations are generated combined with tourist sites, travel, food and accommodation. The implementation of the presented model is supported by data mining and collaborative filtering techniques.

In some cases, dedicated devices, robots or techniques such as virtual reality or machine learning can help tourist guides. Deciding what to see in a large museum or recognizing folklore elements in an exposition can be overwhelming. In [14], a Smartwatch-based system is presented designed to facilitate museum gallery exploration. The design and implementation of the system are also discussed, followed by an evaluation conducted with twelve visitors to a Natural History Museum. In [15], the authors introduce the implementation of a tour-guide robot using Kinect technology to facilitate the process of tourist guides. The robot is able to replace a human guide; it will follow tourists wherever they go, avoiding obstacles in its way, and provide information about the place once the tourists have given it the command to do so. The robot detects different objects, thus providing information about them.

3 General Review of the Tourist Guide

The Tourist Guide (TG) provides intelligent assistance taking into account the personal preferences, the actual location and available time for tourists. System users have the opportunity to ask the system questions about a specific cultural or historical object and obtain information about it in the form of text, video, and pictures. The system can determine the location of the user using the GPS on his/her mobile device and enquire if he/she wants to look at the cultural and historical sites in the vicinity. It can do a survey to determine what type of objects or sites he or she is interested in and suggest generating a cultural and historical route to follow. An overview of the TG functions is presented in Fig. 1.

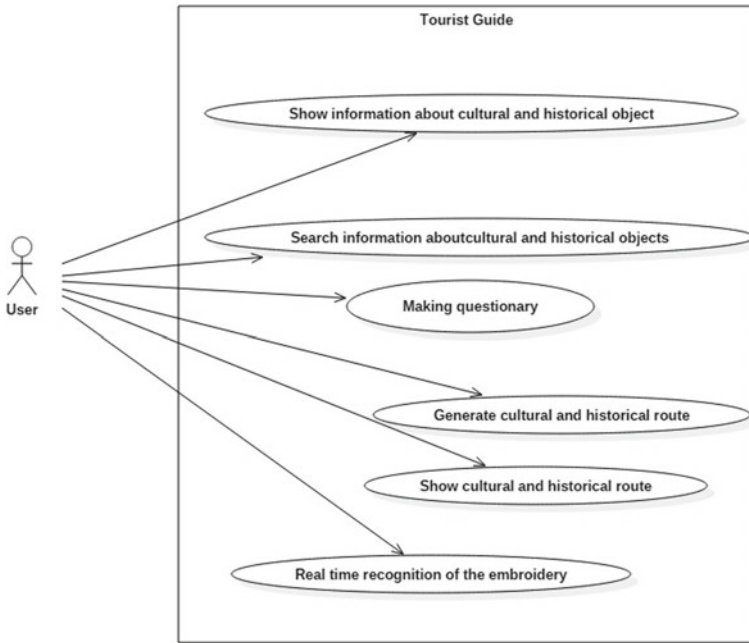


Fig. 1 A use case diagram of the tourist guide

A hierarchy of various ontologies, developed following the CCO standard, was implemented to present the cultural and historical objects. Each object has two types of presentation on the guide’s server:

- A cultural and historical object (CHH objects)—depending on the nature of the presentation, it includes different features in accordance with the CCO standard.
- Ambient—for characterization of the location and condition as a physical feature in a real location (area) of a separate CHH object or a group of CHH objects, designated as an exposition.

In addition to the functions of the environment, a module for automatic recognition of different folklore elements, included in the objects, is being developed. In the current version, embroidery recognition capabilities will be integrated. For this purpose, an algorithm for machine learning with neural networks is used, where based on a real-time photo of the embroidery, the system identifies what type it is and supplies information about it.

The tourist can choose one of the two options to implement the route:

- A virtual tour—the TG provides information about selected cultural objects in the form of text, photos, and videos, depending on the information stored on the system server.

A real walk—the TG assists the tourist during his/her visit and viewing of the objects.

4 The Tourist Guide Architecture

The TG is implemented as a multi-agent system consisting of intelligent and reflex agents. The agents are located in the two parts of the guide (Fig. 2.):

- A back-end component: it consists of different modules, distributed in two layers—a knowledge base and operational assistants performing the tasks of gathering information for the client’s needs and generating it in an appropriate cultural and historical route;
- A front-end component: it consists of an intelligent assistant that takes care of presenting the route and object information to the client’s mobile device using the information generated by the operational assistants in the back-end layer.

In this part of the article we will briefly present the individual components included in the TG architecture.

Questioner Generation Assistant (QGA). This Operations Assistant is responsible for user surveys. Its main task is to create a user account that contains the main objects about which he/she wishes to obtain information. The survey is visualized by the application interface on the user’s mobile device (TG). The assistant poses a sequence of questions to identify the user’s areas of cultural interest. For example, the user can indicate that he/she is interested in cultural and historical objects or sites and folklore in the region of Plovdiv. The assistant proceeds with specific questions, for instance if the user takes interest in national costumes he/she may see in the

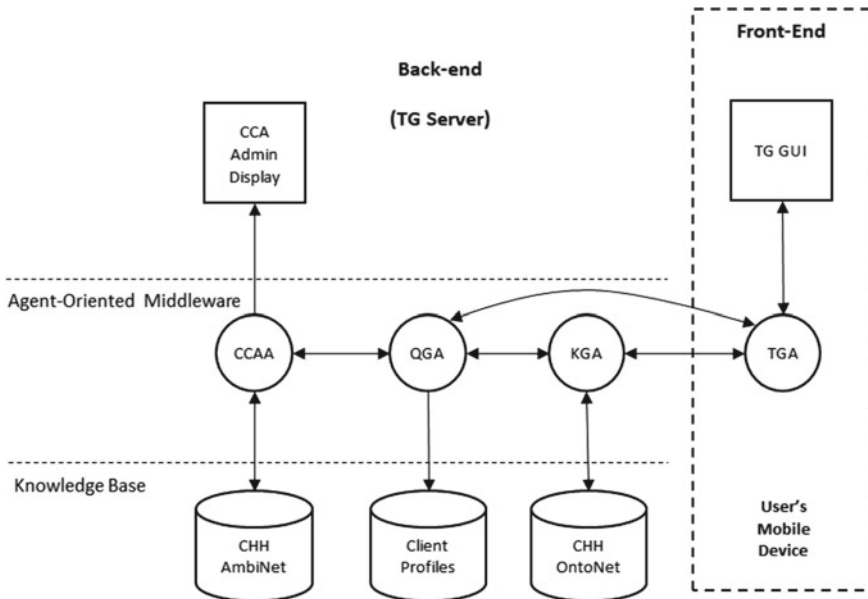


Fig. 2 General architecture of the tourist guide


```
Agent container Main-Container@192.168.0.104 is ready.
-----
Език/Language: bg | en
en
Hello user! I`m Console agent! What do you want to know today? Let me ask you some qu
estions first.. :)
Where do you want to go?
Plovdiv
What object do you want to see?
costume
log4j:WARN No appenders could be found for logger (Jena).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
In Plovdivthere are 2 types of bulgarian national costume. Which one do you want to s
ee?
male costume
female costume
male costume
In Plovdivthere are 2 types of male costume. Which one do you want to see?
white colored man costume
black colored costume
white colored man costume
In Plovdiv I can show you info about or you preffer to go there and see it?
show
see
show
Good choice, dear. But I`ll give you more information next time... Right now the KGA
is not available.
```

Fig. 3 An extract from a survey conducted by the QGA

area. According to the received answer, the agent uses the ontology knowledge base, which presents the cultural objects of Bulgaria, to ask a subsequent question, for example what kind of costumes he/she is attracted by. At the end of the QGA survey, the agent generates a list of objects that engage the user and which it registers as interests in his/her profile.

This information is passed to another assistant called the Knowledge Generator Assistant. An extract from a QGA survey is presented in Fig. 3.

KGA (Knowledge Generation Assistant). Using the tourist profile, the assistant selects the elements of the primary route. The primary route elements are expositions or separate CHH objects.

CCAA (Calculus of a Context-aware Ambients Assistant). It generates a final route by completing the primary route with additional information such as the location and status of the expositions (or individual objects), the working time, etc. The assistant uses the ambient presentation of the CHH objects included in the primary route. In fact, the final route is a set of possible sequences for viewing the objects. The Calculus of Context-aware Ambients Assistant (CCA-A) is an intelligent assistant which has to convert the original route, created by the KGA, into an appropriate format for the CCA system, taking into account the dynamically incoming information from the outside world. The Calculus of Context-aware Ambients is a system based on the ambient notation. This is an identity used to describe an object or a component such as a process, a device, a location, etc. It enables mobile ambients to react in response to environmental changes. Consequently, mobility and context-awareness

obtain particular importance and significance. In our system ambients are the cultural and historical objects that are located in the CHH AmbiNet knowledge Base. The CCA generates variations of routes depending on the client's wishes, cultural object specifics, working time of the museums, and the user's location. It sends the generated route models to the QGA, which, in turn, lists the routes in order, from where the user can select a route that suits him/her, and choose whether it is virtual or real.

TGA (Tourist Guide Assistant). It operates in the Front-end component and performs the following basic functions:

- It serves as a tourist's GUI—the tourist can only communicate with the guide through this assistant. This agent is responsible for the proper visualization of the information obtained from the operating agents on the client's mobile device. It visualizes the questions that the QGA generates and returns the received answers back to it. It is responsible for visualizing the information about the various cultural and historical objects and for visualizing the route generated by the CCA.
- Establishing the tourist location—this is achieved by using the GPS capabilities of the client's mobile device to determine his or her position.
- Life Cycle Management—it prepares a “schedule” for visiting the cultural objects and follows its observance.

Real-time embroidery recognition. An important part of the Bulgarian cultural and historical heritage are the national costumes and dresses decorated with a lot of embroidery and needlework. In every region in Bulgaria embroidery has specific characteristics, which are determined by the combination of colors, stitches, symbols, ornaments, sewn beads, coins, tassels, and sequins. These features can determine the place where the embroidery was made. By using machine learning methods, in particular neural network algorithms, the tourist guide can be taught to recognize embroidery, conclude whether it is Bulgarian or not, and classify it in relation to its area of production [16]. This classification could help it to extract information from related ontologies and to enrich and expand the user's knowledge.

The user can take a photo of embroidery and the TG can try to identify the embroidery in real time and convey information to the tourist. The task of the tourist guide can be divided into three separate parts: recognition of an image as embroidery, identification of the embroidery as Bulgarian, and its classification in relation to the area of production. All of these three tasks can be completed by using methods based on neural networks.

A solution to the first task—recognizing the image as embroidery, is Hopfield's associative memory [17], whose neural network was used for this purpose. The embroidery itself is represented as a vector of ornaments. The ornaments are the different characteristics of the embroidery—its color, shape, motif, and stitch. By joining these elements a matrix of vectors is obtained. The set goal is to teach the network to recognize two of the basic stitches used in embroidery—slanted and cross. Recognizing stitches and from there embroidery motifs too can serve as input for the KGA to find the relevant national costume in the knowledge base with ontologies and present it to the user.

5 CHH OntoNet

The knowledge base of the Tourist Guide has been developed as a semantic network describing knowledge about the cultural and historical heritage of Bulgaria (CHH OntoNet). Our main task was to cover Bulgarian folklore—folk costumes, dialects in Bulgaria, and old houses that identify the unique cultural and historical heritage of Bulgaria. Currently, it includes 10 ontologies based on the CCO standard, 400 classes, and 3000 axioms, presented in 2 languages: Bulgarian and English. Protégé [18] was used to implement the ontologies. Figure 4 shows a schematic diagram of them.

Ten ontologies have been created, most of which correspond to the CCO standard dictionaries; these ontologies are Cities, Folklore Regions, Subjects, Agents, Materials, and Objects. Each of them describes a concept and its characteristics necessary for the presentation of the cultural and historical objects. Two ontologies have been developed that interpret the idea of the CCO standard objects: Landmarks and Costumes. The objects described in these two ontologies support the elements required by the standard, but additional features have also been added to fully represent the concepts. The first ontology (Landmarks) contains knowledge about different landmarks in Bulgaria. The Costumes ontology describes traditional Bulgarian costumes and presents knowledge about the most common types of costumes, their characteristics including mandatory or additional items (clothing) based on the region, in which they are used, and others. For each costume, different clothes are defined together with their distinctive features.

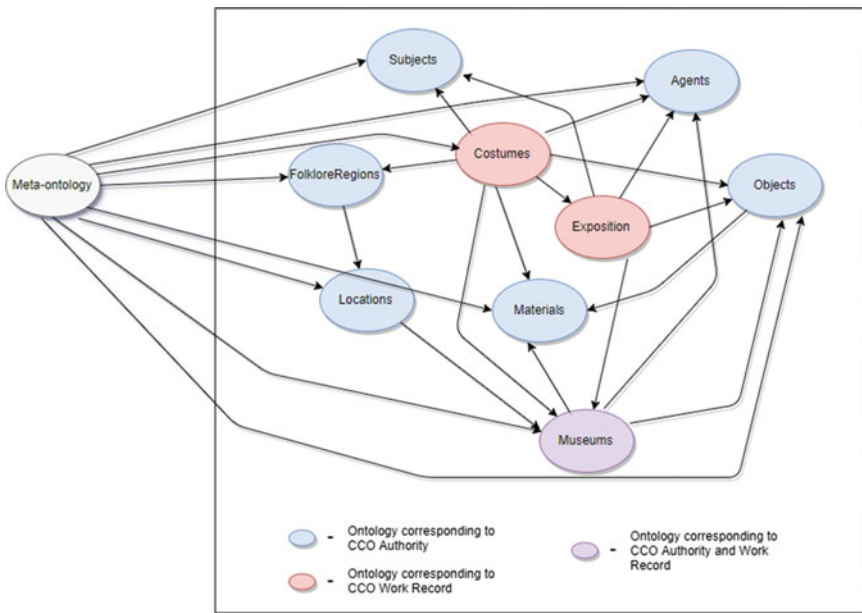


Fig. 4 Hierarchical structure of the ontologies in CHH OntoNet

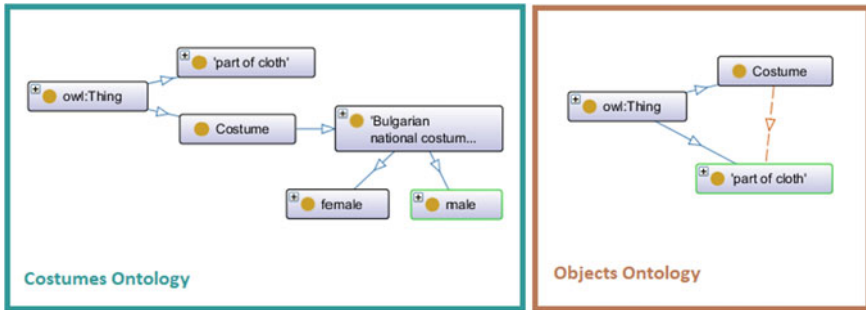


Fig. 5 Using terms from the objects ontology in the costumes ontology

Bulgarian costumes are extremely colorful, with varied details depending on the region and the time they were worn, the occasion, the season, etc. This diversity, as well as adherence to the CCO standard, necessitates differentiating concepts into separate ontologies. On the other hand, a method is needed for these concepts to be treated as a whole by intelligent agents. The approach chosen to manage the classes, individuals and properties is by using Internationalized Resource Identifier (IRI). Each defined class, individuality or property has a unique IRI. Accordingly, an ontology-defined class may be used in another one; during the creation of the concept in the second ontology the unique IRI is set, which is obtained while defining the concept in the first ontology.

For example, the Costumes ontology uses terms such as Clothes and Costume, which are described in the Objects ontology (Fig. 5).

The Costumes ontology is an example of the realization and presentation of the various cultural and historical objects in the CHH OntoNet. The Costume is a basic class. It is defined in the Objects ontology, where the costume features are described as a garment type. Since on the one hand traditional Bulgarian costumes are a kind of costumes, but on the other hand they have unique features too, a Bulgarian National Costume subclass of the Costume class was created in the Costumes ontology. Here the exact elements of each costume are described, including the mandatory elements that highlight the types of Bulgarian traditional costumes, as well as additional ones. The costumes are divided into two main types: male and female, with additional subclasses of Bulgarian National Costume, Male and Female respectively. There follows the definition of the varieties of male and female costumes.

A lot of characteristics or attributes have been created in the ontologies to set the axioms needed to describe the concepts. Among the properties in the Costumes ontology are `hasAdditionalClothesElement`, `hasRequiredClothesElement`, `isPlacedIn`, `isCreatedBy`, `isCreatedOn`, and others. Some of them represent the items that the objects must own according to the CCO standard (`isCreatedBy`, `isCreatedOn`); others have been added to obtain a detailed and accurate description of the costume, for example the `hasRequiredClothesElement` attribute. It is used to indicate

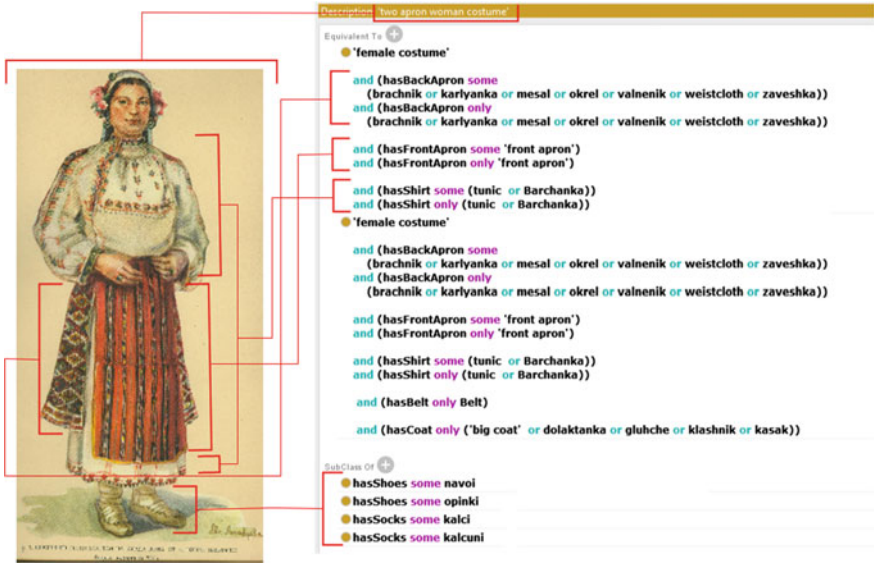


Fig. 6 A two-aproned woman costume

the mandatory elements of specific costumes. Similarly, `hasAdditionalClothesElement` is used to create axioms, which define other elements that a particular costume may include, but are not necessarily present.

The structure of the ontology is demonstrated with the `TwoApronWomanCostume` class (Fig. 6). The class presents a woman's two-aproned costume. Typically, every Bulgarian traditional costume is determined by the types of clothes involved in its composition.

The distribution of knowledge in particular ontologies is very important. It is easy and convenient to compare knowledge to the requirements of a standard. In addition, the partition of the domain of the cultural and historical heritage of Bulgaria into separate sub-domains allows effective, distributed maintenance and editing of the ontologies and the knowledge in them. Specific ontologies can be upgraded and changed without influencing the others. Moreover, the addition of knowledge and new ontologies related to new objects is simple and it does not require making changes to the structure of the other ones.

The objects of the cultural and historical heritage such as traditional Bulgarian costumes are usually placed in different expositions. Meanwhile, these expositions are located in specialized museums, which is a precondition for the development of additional ontologies containing knowledge about expositions and museums.

In addition to all the discussed ontologies related to the domain of the Bulgarian cultural heritage there is another one called Meta Ontology. Its purpose is to be used by the QGA for generating the first questions about the location and type of objects. Since there are several ontologies describing different subdomains, it is necessary to make a simple mapping that shows what kind of knowledge to search for in which

ontology. For this reason, the Meta Ontology has only two classes—QuestionWord and Answer, one object property, called isRelatedWith, and several individuals that are used by the agent.

6 CHH AmbiNet

In ambient representation objects are specified as abstract nodes of a graph (AmbiNet). An ambient is an identity characterized by a boundary, location and mobility, and it is possible to create hierarchies or networks of ambients [19]. For modeling purposes, an ambient will be represented as a structure with the following elements: an identifier (name), a corresponding set of local agents, and a set of adjacent sub-ambients. In this way there is an opportunity for recursively building more complex structures of ambients.

Ambient Intelligence (AmI) makes the daily information environment of users “more sensitive” to their problems and peculiarities by adding sensors for sensing and actuators for influencing the users’ environment. As determined in [20], the main features of AmI technologies are sensitivity, adaptability, transparency, universality, and intelligence. The basic idea behind AmI is that applications using real-time aggregated information and background data accumulated over time can make decisions for the benefit of users. Typical examples are personal assistants that, depending on the situation and current context, are able to provide proactive help. Sensible requirements to them can be the ability to recognize the tourist, to learn or know his/her preferences and likes, and to help organize his/her tourist trip.

The π -calculus, which represents a kind of process calculus [21], is a fundamental formalism, suitable for ambient-oriented modeling. To model the scenarios in the CHH AmbiNet, we will use the Calculus of Context-aware Ambients (CCA) [22], which enables mobile ambients to respond to environmental changes. Consequently, the mobility and awareness of the context gain particular importance and significance. To represent the properties of the CCA processes, a logical language is used, in which the context expression plays a major role. Context expressions are used to ensure fulfillment of a certain opportunity only in the presence of specific environmental conditions, i.e. in a specific context.

In CCA notation, there are three possible relationships (parent, child, sibling) between two ambients. Each ambient can interact with other ambients by sending and receiving messages through the handshaking process. The notation “:” is a symbol for any sibling; “ \uparrow ” and “ \downarrow ” are symbols for parent and child; “ $\langle \rangle$ ” means sending, and “ $()$ ”—receiving a message. An ambient can be mobile; with CCA, it can move in and out: entering another ambient and becoming its child or coming out of its parent and becoming its sibling. In CCA, four syntax categories can be distinguished: processes P, capabilities M, locations α and context expressions k.

In CHH AmbiNet we will present the CCA-modeling of the following service: Generating a route to visit expositions with Bulgarian embroidery, determined after

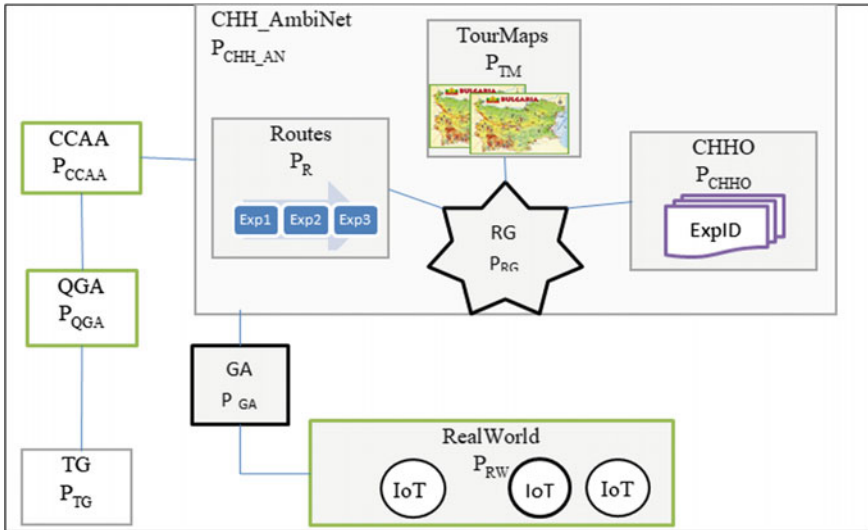


Fig. 7 Ambient structure

a survey with the tourist. The following figure presents the main ambients and the interaction between them (Fig. 7).

Description and Service Scenario:

- After a dialogue with the tourist’s personal assistant (TG), the QGA determines his/her willingness to review expositions containing Bulgarian embroidery and sends a request to the CHH_AmbiNet space to implement the service.
- The CHH_AmbiNet space processes the request data to verify that there is not a ready route in the collection with already generated Routes. If none exists, it summons the RG route generator requesting that a new route should be generated.
- The RG ambient extracts a list of embroidery expositions from the CHHO and sends this list to the GA guards, which in turn verify by communicating with the real world PW whether these expositions are open for real or virtual world visiting.
- As soon as the RG receives the list of accessible expositions, it requires from the TourMaps ambient a suitable tourist map and, by a modified method A*, searches for appropriate routes, which it records in the Routes collection.
- The generated routes, together with the tourist map, are sent to the tourist’s TG to implement a real or virtual tour.

CCA modeling of the service

The ambients involved in the modeling process are:

- TG—Tourist personal assistant (Tourist Guide)
- QGA—Questioner Generation Assistant
- CCAA—Calculus of Context-aware Ambients Assistant

- CHH_AN—ambient parent (Culture Historical Heritage AmbiNet)
- CHHO—(Culture Historical Heritage Object) an ambient containing as sub-ambients tourist objects with their physical characteristics such as location, expositions, in which they are currently included, working time, etc.
- ExpID—an ambient of a specific exposition, in which the tourist object is located
- TourMaps—an ambient containing tourist maps as sub-ambients
- Routes—a collection of already generated tourist routes
- RG—a Route Generator
- GA—guards providing the connection to the physical world
- PhW—a Physical World ambient.

The QGA receives a request from the TG of a particular tourist containing his/her location and a request to survey expositions with Bulgarian embroidery. Upon receiving the request, the QGA sends it to the CCAA to generate an appropriate route.

$$P_{QGA} \triangleq \left(\begin{array}{l} !TG :: (TG_i, location, WantSeeEmbroidery). \\ CCAA :: < TG_i, location, Embroidery > .0| \\ !CCAA :: (TG_i, ListRoutes).TG :: < ListRoutes > .0 \end{array} \right)$$

The CCAA receives a request from the QGA requesting a route to visit expositions with embroidery in the area, which it passes to the CHH_AN for execution. As a result, it returns a list of the generated routes (ListRoutes).

$$P_{CCAA} \triangleq \left(\begin{array}{l} !QGA :: (TG_i, location, Embroidery). \\ CHH_AN :: < TG_i, location, Embroidery > .0| \\ !CHH_AN :: (TG_i, ListRoutes). \\ QGA :: < TG_i, ListRoutes > .0 \end{array} \right)$$

As soon as the CHH_AN receives a request from the CCAA, it in turn sends a request to the Routes to search for a completed route. If none exists, it turns to the RG for its generation. In the process of generating an appropriate route, it addresses the GA to receive the list of only the expositions open for visiting (ListOpenedExpositions). Once it receives the generated routes it returns them to the CCAA.

$$P_{CHH_AN} \triangleq \left(\begin{array}{l} !CCAA :: (TG_i, location, Embroidery). \\ Routes \downarrow < TG_i, location, Embroidery, ?has_Route > .0| \\ Routes \downarrow (TG_i, location, Embroidery, No_Route). \\ RG \downarrow < TG_i, location, Embroidery, needGenerateRoute > .0| \\ RG \downarrow (TG_i, location, ListExpositions). \\ GA :: < TG_i, location, ListExpositions > .0| \\ GA :: (TG_i, location, ListOpenedExpositions). \\ RG \downarrow < TG_i, location, ListOpenedExpositions > .0| \\ Routes \downarrow (TG_i, ListRoutes).CCAA :: < TG_i, ListRoutes > .0 \end{array} \right)$$

Routes—it receives a request from its ambient parent CHH_AN to search for a completed route that meets the tourist's requirements. If there is no appropriate route, it awaits the generation of new routes by the RG and forwards them for execution.

$$P_R \triangleq \left(\begin{array}{l} CHH_AN \uparrow (TG_i, location, Embroidery, ?has_Route). \\ CHH_AN \uparrow \langle TG_i, location, Embroidery, No_Route \rangle .0| \\ RG :: (TG_i, ListRoutes). \\ CHH_AN \uparrow \langle TG_i, ListRoutes \rangle .0 \end{array} \right)$$

RG—it receives a request from its ambient parent CHH_AN to search for a route so it turns to the TourMaps with a request to provide an appropriate tourist map, and to the CHHO to generate a list of expositions (ListExpositions). Once it has received the list of expositions, it sends them to the CHH_AN with a demand to check in the real world whether each of these expositions can be visited and viewed physically or virtually. When it receives the list of expositions open for visiting (ListOpenedExpositions), it positions them on the tourist map and, using a suitable algorithm, generates appropriate routes (ListRoutes) that it transmits to the Routes for storage and use.

$$P_{RG} \triangleq \left(\begin{array}{l} CHH_AN \uparrow (TG_i, location, Embroidery, needGenerateRoute). \\ TourMaps :: \langle TG_i, location, getTourMap \rangle . \\ CHHO :: \langle TG_i, location, getListExpositions \rangle .0| \\ TourMaps :: (TG_i, location, TourMap). \\ CHHO :: (TG_i, location, ListExpositions). \\ CHH_AN \uparrow \langle TG_i, location, ListExpositions \rangle .0| \\ CHH_AN \uparrow (TG_i, location, ListOpenedExpositions). \\ Routes :: \langle TG_i, ListRoutes \rangle .0 \end{array} \right)$$

CHHO—as soon as it receives a request from the RG, it checks in which expositions are located the objects requested by the tourist, and then delivers a list of the expositions and information about their location.

$$P_{CHHO} \triangleq \left(\begin{array}{l} RG :: (TG_i, location, getListExpositions). \\ ExpID \downarrow \langle TG_i, location, getLocation_ExpID \rangle .0| \\ ExpID \downarrow (location_ExpID). \\ RG : \langle TG_i, location, ListExpositions \rangle .0 \end{array} \right)$$

TourMaps—upon receiving a request from the RG, it sends an appropriate tourist map of the area.

$$P_{TM} \triangleq \left(\begin{array}{l} RG :: (TG_i, location, getTourMap). \\ RG :: \langle TG_i, location, TourMap \rangle .0 \end{array} \right)$$

GA—it receives a request from the CHH_AN to check whether the respective expositions are available for visiting. Then it sends the request to the ambient supporting the real world information PhW and as a result receives a reply for each exposition. After that it sends a list of all the expositions open for visiting (ListOpenedExpositions) to the CHH_AN.

$$P_{GA} \triangleq \left(\begin{array}{l} !CHH_AN :: (TG_i, location, ListExpositions). \\ PhW :: < ExpID, location_ExpID > .0 | \\ PhW :: (reply, IoT_ID). \\ CHH_AN :: < TG_i, location, ListOpenedExpositions > .0 \end{array} \right)$$

PhW—it receives a request from the GA to verify whether a specific exposition from the list is accessible for visiting and returns a reply containing a confirmation or information about the reason why the real-world exposition cannot be entered upon.

$$P_{PhW} \triangleq \left(\begin{array}{l} GA :: (ExpID, location_ExpID). \\ GA :: < reply, IoT_ID > .0 \end{array} \right)$$

CCA Admin Display

The realization of tourist services depends on changes in the world around us. This determines the necessity of continuous monitoring of the processes of the described ambients in the CHH AmbiNet. The CCA Admin Display module is designed to track the scenario implementation processes. For this purpose, simulators of the presented tourist services have been developed. We will look at a simulator tracking the search and find the tourist routes process described in the presented CCA model. Since the syntax of the CCA contains a collection of specific symbols, it is necessary to use the appropriate programming language—ccaPL, in order to create a simulator of the scenario under consideration. The language is a computer-readable version of the CCA syntax [23]. The interpreter of the ccaPL environment is developed on Java. Founded on the basic version, a special simulator was developed to track the processes of the personalized context-sensitive tourist guide.

Each row of output of the program execution is read as follows: the notation “A === (X) ==> B” means that ambient “A” sends an “X” message to ambient “B”. The notations “Child to parent”, “Parent to child,” and “Sibling to sibling” provide information about the relationship between the sender A and the recipient B in terms of the hierarchy of ambients.

The output data from the environment is difficult to read without prior knowledge of the CCA formalization. For ease of use, an animator has been developed in the ccaPL environment. The objective of the animator is to present graphically the ambients and their processes in a ccaPL program. Figure 8 shows the output screen for tracking the basic scenario for the service realization, modelled in the previous section.

The system administrator can track the implementation of a sample scenario in simulation mode both through the console and the animator in the ccaPL environment.

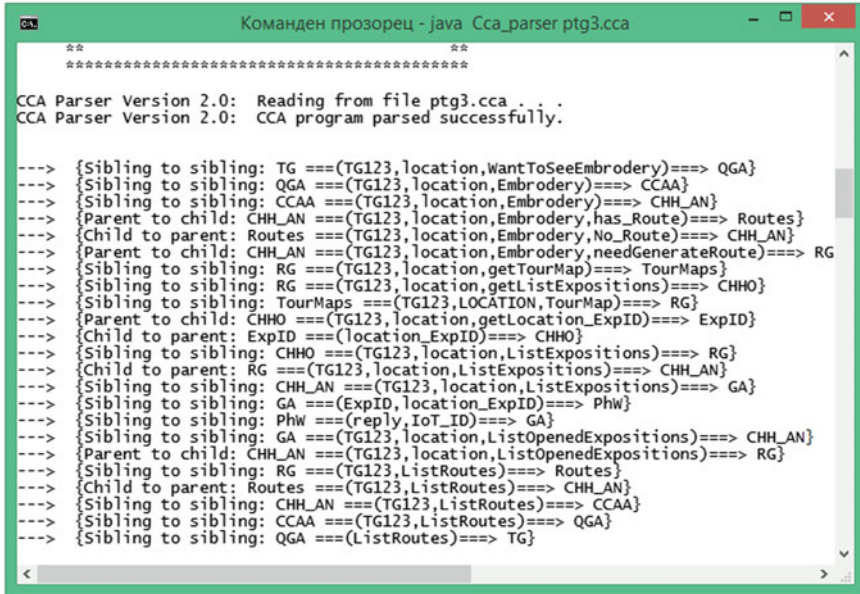


Fig. 8 An administrator console for tracking the processes in the scenario

For example, let us suppose that after a dialogue with the tourist, the QGA has determined his/her desire to visit expositions with national embroidery and let the tourist’s mobile device specify his/her location. The QGA transmits the location and the wish of the tourist to the CCAA. The CCAA transmits this information to the CCA_AN, where, after a dialogue between the ambients described, it is established that expositions are currently open for visiting. Several possible visiting routes are generated—ListRoutes. This list is transferred to the tourist and after he/she selects a route, the TG starts the next services related to visiting the sites. In the Fig. 9 is presented the animated simulator of the example given.

7 Conclusion

The basic components of the Tourist Guide are assistants and ontologies. The assistants are implemented as intelligent rational agents with BDI architecture [24], they are located on mobile user devices and on the Tourist Guide server. The server agents were developed using the JADE development environment [25] and those for mobile devices—in the Jadex environment [26]. The ontologies were implemented in the Protégé environment.

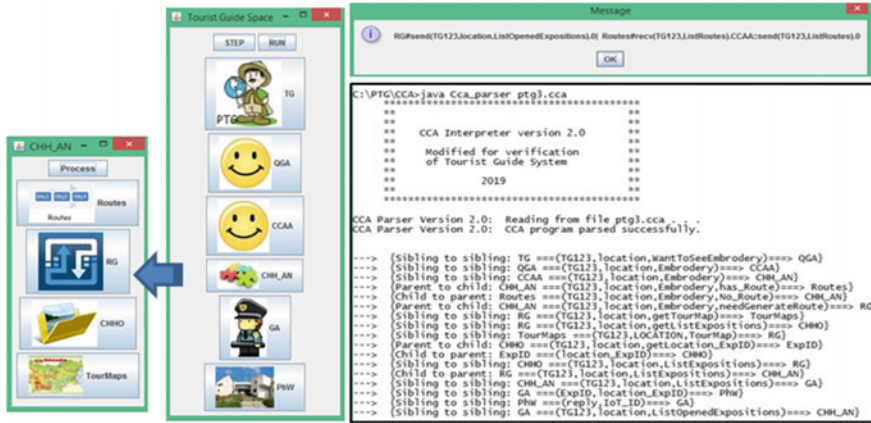


Fig. 9 An animated ccaPL simulator

Currently, new expansions of the Tourist Guide are being prepared. The Tourist Guide can offer various auxiliary tourist services such as information about appropriate restaurants, hotels, shops, and medical assistance. A module for the recognition of folklore elements is being developed by machine learning methods. Another idea under consideration is to integrate the Tourist Guide into a virtual reality environment.

Acknowledgements The authors wish to acknowledge the partial support of the National Program “Young Scientists and Postdoctoral Students” of the Ministry of Education and Science in Bulgaria, 2018–2019 and of the MES by the Grant No. D01-221/03.12.2018 for NCDSC—part of the Bulgarian National Roadmap on RIs.

References

1. Trendafilova, A.M.: Development of an environment for building a common catalogue for representation of the culture-historical heritage of Bulgaria. In: Cybernetics and Information Technologies, vol. 7, Issue 1, Bulgarian Academy of Sciences, pp. 95–105 (2007)
2. http://cco.vrafoundation.org/index.php/aboutindex/who_is_using_cco/
3. Stoyanov, S., Stoyanova-Doycheva, A., Glushkova, T., Doychev, E.: Virtual physical space—an architecture supporting internet of things applications. In: XX-th International Symposium on Electrical Apparatus and Technologies SIELA 2018, 3–6 June 2018, Bourgas, Bulgaria (2018)
4. Stoyanov, S., Orozova, D., Popchev, I.: Internet of things water monitoring for a smart seaside city. In: XX-th International Symposium on Electrical Apparatus and Technologies SIELA 2018, 3–6 June 2018, Bourgas, Bulgaria (2018)
5. Glushkova, T., Miteva, M., Stoyanova-Doycheva, A., Ivanova, V., Stoyanov, S.: Implementation of a personal internet of thing tourist guide. Am. J. Comput. Commun. Control 5(2) (2018). ISSN: 2375-3943, 39-51
6. Smirnov, V., Kashevnik, A.M., Ponomarev, A.: Context-based infomobility system for cultural heritage recommendation: Tourist Assistant—TAIS. J. Pers. Ubiquitous Comput. 21(2), 297–311 (2017) (Springer, London, UK)

7. Gavalas, D., Kasapakis, V., Konstantopoulos, C., Pantziou, G., Vathis, N.: Scenic route planning for tourists. *J. Pers. Ubiquitous Comput.* **21**(1), 137–155 (2017) (Springer, London, UK)
8. Baraglia, R., Frattari, C., Muntean, C.I., Nardini, F.M., Silvestri, F.: RecTour: a recommender system for tourists. In: *IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, pp. 92–96 (2012)
9. Hoadjli, A., Kazar, O., Rezeg, K.: A layered design approach for mobile tourism. In: *8th International Conference on Information Technology (ICIT)*, 17–18 May 2017, Amman, Jordan (2017). <https://doi.org/10.1109/icitech.2017.8079986>
10. Özcan, S.C., Kaya, H.: An analysis of travelling salesman problem utilizing hill climbing algorithm for a smart city touristic search on OpenStreetMap (OSM). In: *2nd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 19–21 October 2018, Ankara, Turkey (2018). <https://doi.org/10.1109/ismsit.2018.8567045>
11. de Farias, I., Leitão, N., Teixeira, M.M.: Urbis: a touristic virtual guide. In: *2017 12th Iberian Conference on Information Systems and Technologies (CISTI)*, 21–24 June 2017, Lisbon, Portugal (2017). <https://doi.org/10.23919/cisti.2017.7975918>
12. Sorrentino, F., Spano, L.D., Scateni, R.: SuperAvatar Children and mobile tourist guides become friends using superpowered avatars. In: *2015 International Conference on Interactive Mobile Communication Technologies and Learning (IMCL)*, 19–20 November 2015, Thessaloniki, Greece (2015). <https://doi.org/10.1109/imctl.2015.7359591>
13. Dileep, M.R., Prathyash, V., Jithin, J.: AyurTourism: A Web Based Recommendation System for Medical Tourism, *ICDCN'19*, 4–7 January 2019, Bangalore, India, pp. 492–495 (2019)
14. Banerjee, A., Robert, R., Horn, M.S.: FieldGuide: Smartwatches in a Multidisplay Museum Environment, *CHI 2018*, 21–26 April 2018, Montréal, QC, Canada (2018)
15. Al-Wazzan, A., Al-Farhan, R., Al-Ali, F., El-Abd, M.: Tour-guide robot. In: *International Conference on Industrial Informatics and Computer Systems (CIICS)*, 13–15 March 2016, Sharjah, United Arab Emirates (2016). <https://doi.org/10.1109/iccisii.2016.7462397>
16. Toskova, A., Toskov, B., Doukovska, L., Daskalov, B., Radeva, I.: Neural networks in the intelligent educational space. In: *Advances in Neural Networks and Applications—ANNA'18*, St. Konstantin and Elena Resort, Bulgaria, pp. 1–6, *IEEE Xplore Digital Library*, VDE (2018). Print ISBN: 978-3-8007-4756-6
17. Pajares, G.: A Hopfield neural network for image change detection. *IEEE Trans. Neural Netw.* **17**(5), 1250–1264 (2006). <https://doi.org/10.1109/tnn.2006.875978>
18. Protégé. <https://protege.stanford.edu/>
19. Cardelli, L., Gordon, A.D.: Mobile ambients. *Theor. Comput. Sci.* **240**, 177–213 (2000). (Elsevier Science)
20. Cook, D.J., et al.: Ambient intelligence: technologies, applications, and opportunities. *Pervasive Mobile Comput.* **5**(4), 277–298 (2009)
21. Milner, R., Parrow, J., Walker, D.: A calculus of mobile processes, Parts 1–2. *Inf. Comput.* **100**(1), 1–77 (1992) (Elsevier Science)
22. Siewe, F., Cau, A., Zedan, H.: CCA: a calculus of context-aware ambients. In: *International Conference on Advanced Information Networking and Applications Workshops—WAINA'09* (2009). <https://doi.org/10.1109/waina.2009.23>
23. Al-Sammarraie, M.H.: Policy-based Approach for Context-Aware Systems, Ph.D. Thesis, Software Technology Research Laboratory, De Montfort University, Leicester, United Kingdom (2011)
24. Wooldridge, M.: *An Introduction to Multiagent Systems*, Wiley, Hoboken (2009)
25. Bellifemine, F., Caire, G., Greenwood, D.: *Developing Multi-agent Systems with JADE*. Wiley, Hoboken (2007)
26. JADEx Agents. <http://jadex-agents.informatik.uni-hamburg.de/>

Pattern Recognition

Comparing Evolutionary Artificial Neural Networks from Second and Third Generations for Solving Supervised Classification Problems



G. López-Vázquez, A. Espinal, Manuel Ornelas-Rodríguez,
J. A. Soria-Alcaraz, A. Rojas-Domínguez, Héctor Puga, J. Martín Carpio
and H. Rostro-González

Abstract Constituting nature-inspired computational systems, Artificial Neural Networks (ANNs) are generally classified into several generations depending on the features and capabilities of their neuron models. As generations develop, newer models of ANNs portrait more plausible properties than their predecessors, accounting for closer resemblance to biological neurons or for augmentations in their problem-solving abilities. Evolutionary Artificial Neural Networks (EANNs) is a paradigm to design ANNs involving Evolutionary Algorithms (EAs) to determine inherent aspects of the networks such as topology or parameterization, while prescinding—totally or partially—from expert proficiency. In this paper a comparison of the performance of evolutionary-designed ANNs from the second and third generations is made. An EA-based technique known as Grammatical Evolution (GE) is used to automatically design ANNs for solving supervised classification problems. Partially-connected three-layered feedforward topologies and synaptic connections for both types of considered ANNs are determined by the evolutionary process of GE; an explicit training task is not necessary. The proposed framework was tested on several well-known benchmark datasets, providing relevant and consistent results; accuracies exhibited by third-generation ANNs matched or bested those from second-generation ANNs.

G. López-Vázquez · M. Ornelas-Rodríguez (✉) · A. Rojas-Domínguez · H. Puga · J. M. Carpio
Division of Postgraduate Studies and Research, National Technology
of México/León Institute of Technology, León, Guanajuato, Mexico
e-mail: manuel.ornelas@itleon.edu.mx

A. Espinal · J. A. Soria-Alcaraz
Department of Organizational Studies,
DCEA-University of Guanajuato, Guanajuato, Mexico

H. Rostro-González
Department of Electronics, DICIS-University of Guanajuato,
Salamanca, Guanajuato, Mexico

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_42

Furthermore, produced networks achieved a considerable reduction in the amount of existing synapses, as in comparison with equivalent fully-connected topologies, and a lower usage of traits from the input vector.

Keywords Artificial neural networks · Grammatical evolution · Evolutionary artificial neural networks

1 Introduction

Artificial Neural Networks (ANNs) are powerful algorithms inspired in biology that represent computational models based on the dynamics of biological brains. ANNs have been applied in a wide variety of scopes to solve many kinds of problems, e.g. pattern recognition [20, 34], robotic locomotion [14, 33], function approximation [6, 29], etc.

Inherently, the implementation of an ANN to solve a specific problem implies the definition of its design; the designer typically performs this process in an empirical manner, by following a practical standard or based on experimental insight, since the complexity of the problem and the vastness of the design criteria suppose the absence of a general methodology to intuitively produce an ultimate solution. As the ANNs design is intrinsically related to performance, exploring the issue of achieving an appropriate design process has been a research endeavor; combinatorial issues [1] and learning issues [15, 19] have been previously addressed. Progress has been made in easing or enhancing the learnability of the ANNs, for example, by limiting the ANN's architecture [16]; partially-connected ANNs have shown competitive performance when compared to their fully-connected equivalents, while lowering their complexity [7].

Consequently, a combination of ANNs and Evolutionary Algorithms (EAs) has constituted the so-called Evolutionary Artificial Neural Networks (EANNs), in an effort to endue the ANNs with an optimized design that reckon all required criteria (e.g. optimizing weights, topology, learning rule, etc.). In this manner, EANNs may allow to resolve the inconveniences of ANN design, while prescinding, partially or completely, from human experts [5, 23].

More recently, third-generation ANNs (i.e., Spiking Neural Networks, SNNs) are increasing in utilization in consequence of the computational capability of the spiking neurons [11, 19]. Unsurprisingly, the trending in utilizing EAs to determine design features has also reached this generation; in [8, 9], SNNs topologies are defined by a EA known as Grammatical Evolution (GE) [28].

This work proposes an EA-based methodology to design three-layered feed-forward ANNs, in order to assess the differences between the second and third generations while solving supervised classification problems. The methodology considers partial connectivity between input and hidden layers, which may contribute to reduce the topological complexity of the generated EANNs, and prescinds from an explicit training process due to an inherent search process which automatically aims to find

the best parameters. The rest of the paper is organized as follows: Sect. 2 exposes a brief description of the main concepts employed in this work, Sect. 3 depicts the proposed methodology, Sect. 4 contains the experimental configuration of the scheme and the results obtained, and Sect. 5 includes the conclusions and some insight into future work.

2 Background

2.1 Artificial Neural Networks

The Artificial Neural Networks (ANNs) are computing systems pictured as a set of interconnected units (so-called *neurons*) that aim to resemble the biological brain. The network is able to convey information through the connections (also known as *synapses*) as each unit process its incoming signals to propagate them along its linked units. Typically, synapses between units are conditioned by a *weight* that modulates the strength of the signal, and that can be adjusted by means of a training process. ANNs are used to comply with the notion of learning [16], e.g. to perform operations such as classification or grouping, among other complex tasks, based on training data. There can be distinguished three generations of ANNs according to their computing units [18]; the first attempts on the field are those of threshold units such as the McCulloch-Pitts units [21] or the Perceptron [26], although further research has led to the development of the second, and more recently, to the third generation of ANNs. **Second-generation ANNs.** The second generation of ANNs considers computing units that apply continuous activation functions (whether linear or non-linear). ANNs of this generation can be trained with gradient descent-based algorithms such as the Backpropagation learning rule [27].

Third-generation ANNs. The third generation of ANNs is constituted by the Spiking Neural Networks (*SNNs*), as they introduce the firing time component in their computation process [18]. Therefore, this generation is based on spiking neurons [10] such as the Integrate and Fire model [17] or the Hodgkin-Huxley neuron [13]. Networks from this generation are able to handle and solve problems of digital, analogical and spatio-temporal types.

2.2 Evolutionary Algorithms

The Evolutionary Algorithms (EAs) are population-based optimization algorithms that use mechanisms inspired in biological evolution. EAs entail a population of individuals, each representing a candidate solution for a given problem. Through the iterative application of specific operators, the population is evolved over time to

comply with a measure of quality. Crossover, Selection and Mutation are some of the operators applied on the population over the generations.

2.3 Evolutionary Artificial Neural Networks

Evolutionary Artificial Neural Networks (EANNs) are a combination of ANNs and EAs, that emerged as an attempt to resolve some of the most prominent issues inherent to the ANNs; through Evolutionary Algorithms, EANNs are optimized to automatically determine their parameters, encoding schemes, fitness functions, evolution processes, training, architectures, learning rules, etc. [32].

3 Methodology

The methodology portrayed in this section (Fig. 1) is proposed to automatically design partially-connected feedforward Artificial Neural Networks of second and third generations to solve supervised classification problems by means of Grammatical Evolution. In it, a Context-Free Grammar in Backus-Naur Form (*BNF Grammar*) is used to govern the creation of words, then a *Mapping Process* transforms these words into functional ANNs designs. Such networks are assessed by a *Fitness Function* and a *Target* definition, and finally a *Search Engine* (an Evolutionary Algorithm) guides the iterative process of optimization required by the Grammatical Evolution.

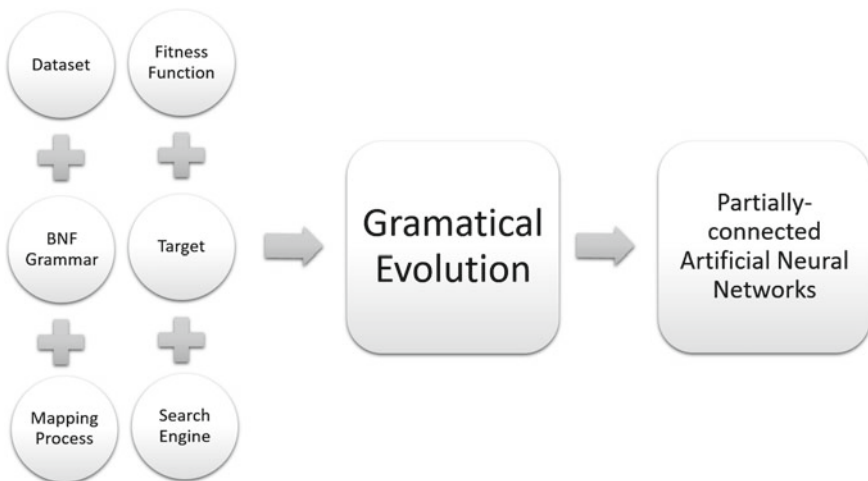


Fig. 1 General diagram of the proposed Framework

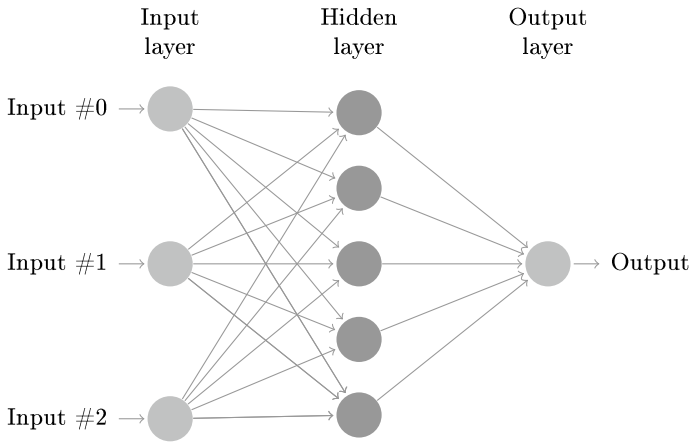


Fig. 2 Depiction of a multilayer perceptron

3.1 *Second-Generation: Multilayer Perceptron*

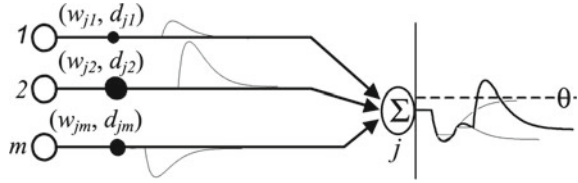
The Multilayer Perceptron (MLP) (Fig. 2) is a type of second-generation fully-connected feedforward Artificial Neural Network with at least three layers: input layer, one (or more) hidden layer(s), and an output layer. A non-linear activation function is employed in each unit in the hidden and output layers when forwarding the information, and backpropagation [27] is typically used for training. The MLP is employed in this work as the choice of second generation model for comparison purposes.

3.2 *Third-Generation: Spike Response Model*

As a third-generation ANN model, the Spike Response Model (SRM) is conceived to more closely resemble the dynamics of the electric potential in biological neurons. The incoming impulses from the dendritic connections (i.e., the presynaptic connections) of a specific unit are build up through time as a summation until a certain threshold value (θ) is reached. At this specific instant in time, the focused unit emits an impulse to forward it through its axon connections (i.e. postsynaptic connections) to other units (see Fig. 3). Each incoming impulse is weighted and delayed by its specific synaptic parameters.

To calculate the electric potential v of unit j at time t , a linear summation of the presynaptic potentials $p_i(t)$ from its connecting presynapses Γ_j is considered, weighed by the w_{ji} parameter, as in Eq. (1).

Fig. 3 Electric potential of unit j : linear summation of excitatory/inhibitory presynaptic potentials. Adapted from [2]



$$v_j(t) = \sum_{i \in \Gamma_j} w_{ji} p_i(t) \tag{1}$$

The potential $p_i(t)$ is then calculated as in Eq. (2), where t_i is the impulse time of unit i , d_{ji} is the synaptic delay parameter, whereas τ is the time constant parameter that marks the decay rate of the potential.

$$p_i(t) = \begin{cases} \frac{t-t_i-d_{ji}}{\tau} \exp\left[-\frac{t-t_i-d_{ji}}{\tau}\right] & \text{if } (t - t_i - d_{ji}) > 0 \\ 0 & \text{else} \end{cases} \tag{2}$$

This work considers the SRM as basis to develop third-generation ANNs.

Temporal Encoding. For input data often represent scalar information, an adequacy must be performed beforehand to enable the data to be introduced into the spiking network. Thus, data x is encoded to stand for spikes $y(x)$ over a period of time $[t_l, t_u]$, by employing the one-dimension encoding [3] in Eq. (3)

$$y(x) = \left[\frac{(t_u - t_l)}{r} * x \right] + \left[\frac{(t_l * x_M) - (t_u * x_m)}{r} \right] \tag{3}$$

where x_M and x_m are the maximum and minimum values that x takes, respectively; r is the range between x_M and x_m .

3.3 Grammatical Evolution

As an Evolutionary Algorithm, the Grammatical Evolution (GE) [24] acts as an optimization process centered on a population of individuals, each representing a possible solution for the problem. By employing a problem-specific context-free grammar and a mapping process, the GE relates the genotype, i.e. the genetic information of an individual, to its phenotype (i.e., the actual configuration of a functional ANN). A search engine iterates this process evaluating every individual (or more adequately, ANN) with a fitness function, aiming to evolve the population towards the optimal solution.

BNF Grammar. Given that the ANN unit models deployed in this work are intrinsically different, two different grammars must be employed to accommodate the needs accordingly. The *Backus-Naur Form* (BNF) grammar proposed in [25] (Grammar 1) is used to develop second-generation ANNs, while Grammar 2 is employed to define third-generation ANNs. Both grammars aim to define the topology of the network focusing on one hidden unit at a time, defining the parameters for its presynapses and posynapses.

```

(network) = (hiddenNeurons)&(outputLayer)
(hiddenNeurons) = (hiddenNeuron) | (hiddenNeuron)_(hiddenNeurons)
(outputLayer) = i0,(weight)_i0,(weight)_ ...
(hiddenNeuron) = (neuronInputs)/i0(weight)#(neuronOutputs)
(neuronInputs) = (neuronInput) | (neuronInput)/(neuronInputs)
(neuronOutputs) = (neuronOutput) | (neuronOutput)/(neuronOutputs)
(neuronInput) = (inputNeuron),(weight)
(neuronOutput) = (outputNeuron),(weight)
(inputNeuron) = i1 | i2 | i3 | i4 ...
(outputNeuron) = o1 | o2 | o3 ...
(weight) = (sign)(digitList).(digitList)
(sign) = + | -
(digitList) = (digit) | (digit)(digitList)
(digit) = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Grammar 1 BNF grammar for designing second-generation ANNs, as in [25]

```

(architecture) = (hiddenNeurons)
(hiddenNeurons) = (hiddenNeuron) | (hiddenNeuron)(hiddenNeurons)
(hiddenNeuron) = (preSynapses)(posSynapses){
(preSynapses) = (preSynapse) | (preSynapse)(preSynapses)
(preSynapse) = (inputNeurons)@(synapse)(
(posSynapse) = (outputNeurons)@(synapse)
(inputNeurons) = 0 | 1 | 2 | 3 ...
(outputNeurons) = o
(synapse) = (weight),(delay)
(weight) = (sign)(digit)(digit)(digit).(digit)(nonZeroDigit)
(delay) = (delayBound)(digit).(digit)(nonZeroDigit)
(sign) = + | -
(delayBound) = 0 | 1
(nonZeroDigit) = 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
(digit) = 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

```

Grammar 2 Proposed BNF grammar for designing partially-connected third-generation ANNs

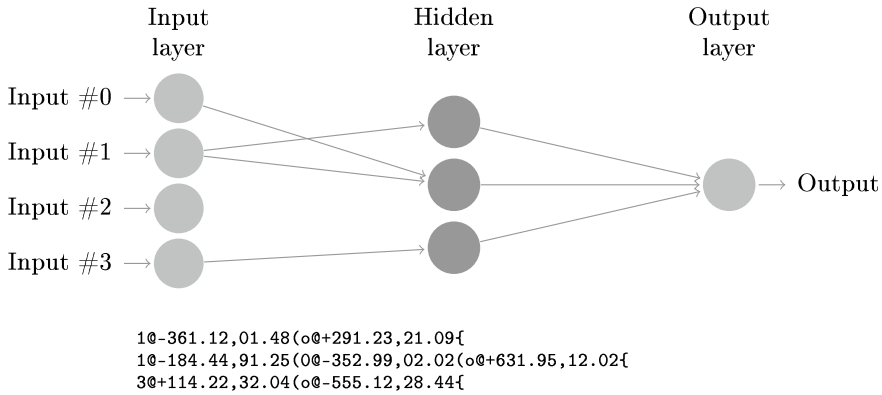


Fig. 4 Sample network and corresponding word generated by Grammar 2

Regarding words produced by Grammar 2, the various hidden units are marked with the opening curled-bracket symbol ($\{$), whereas the different synapses are delimited by the opening parenthesis ($($). The weight and delay parameters are preceded by the *at* symbol ($@$). A sample network and its corresponding word generated by Grammar 2 is illustrated in Fig. 4.

Mapping Process. With the aid of the BNF Grammar, the mapping process translates the genotypic traits from an individual (typically a succession of numbers) into a codified word that define the characteristics of a network. This work utilizes the depth-first mapping process, which employs a tree-like structure to progressively derive its leftmost nodes until all the possible derivations are depleted. Basically, it determines the next element of the word by selecting between the options in the grammar with the result of a modulus operation made on the traits in the genotype and the amount of options available in the actual node.

Target. The target is set to represent a particular state of the network while computing the input patterns, as it must codify the class to which a sample belongs. It is useful in the training and recall processes, and so a variation should be evident between generations. In this work, second-generation ANNs employ the one-hot encoding, in which the output units (equal in number to the amount of classes in the specific dataset) set a combination in which the highest output value marks the class. For third-generation ANNs, a particular firing time is assigned to each class in the dataset employed, resulting in a desired time-to-first-spike in the output unit for every sample belonging to a specific class.

Fitness Function. The squared error is considered in this work as the fitness function to assess the quality of the solutions. Nevertheless, the function definition must be accommodated to comply with the specific neural model. The squared error definition regarding the considered second-generation model is portrayed in Eq. (4).

$$E_2 = \sum_p^P \sum_o^O (h_o^a(p) - h_o^d(p))^2 \quad (4)$$

where P is the number of training samples, O is the number of units in the output layer, while $h_o^d(p)$ is the value of the output unit o with regard to the one-hot encoding definition for the class of the training sample p . $h_o^a(p)$ is the actual value of the unit o . Referring to the considered third-generation model, its squared error function definition is as in Eq. (5).

$$E_3 = \sum_p^P \sum_o^O (t_o^a(p) - t_o^d(p))^2 \quad (5)$$

where $t_o^a(p)$ is the actual firing time and $t_o^d(p)$ is the desired firing time of neuron o .

Search Engine. The Genetic Algorithm (GA) was chosen to be the Search Engine in this work, to conform with the methodology in [25]. The well-known GA [31] guides the optimization process required by the Grammatical Evolution.

4 Experiments and Results

The following datasets were taken for experimentation purposes: Balance Scale, Breast Cancer Wisconsin (Breast Cancer), Glass Identification (Glass), Ionosphere, Iris Plant and Wine (see Table 1). Every dataset from the list is available at the UCI Machine Learning Repository [4].

An even distribution of the datasets was deemed to ultimately acquire two subsets of approximately the same size serving as the Design and Test sets. Samples belonging to specific classes were evenly distributed between the subsets.

The Design set was used to run the GE process to determine the best network topology and parameters possible. Then, the Test set was used to assess the performance of the designed network.

Table 1 Datasets taken for experimentation

Dataset	Instances	Classes	Features
Balance scale	625	3	4
Breast cancer	683	2	9
Glass	214	6	9
Ionosphere	351	2	33
Iris plant	150	3	4
Wine	178	3	13

The following configurations were considered:

- Configuration **A**; it employs the parameters defined in [25], focusing on developing second-generation partially-connected ANNs.
- Configuration **B**; it aims to be an homology of configuration *A* but used to produce third-generation partially-connected ANNs.

Thirty three independent experiments for each configuration were performed to comply with the Central Limit Theorem [12] and so provide statistical certainty. Specific parameters set for experimentation are provided next:

SRM. *Membrane potential time constant* $\tau = 9$, *Target*—depending on the number of classes in the data set - [12 (ms), 15 (ms), 18 (ms), . . .], *Simulation time*—[10 (ms), target of the last class plus two], *Threshold* $\theta = 1$ millivolts (mV), *Weight range* $\in [-999.99, 999.99]$, *Delay range* $\in [0.01, 19.99]$ (ms).

Temporal Encoding. The one-dimension encoding scheme observes a temporal range from 0.01 to 9 (ms).

GA. Binary Search Space, *Codon size* = 8, *Individual Dimension* = 4000 (500 codons), *Population size* = 100, *Function calls* = 1,000,000, *Selection*—K-Tournament ($K = 5$), *Elitism percentage* = 10%, *Crossover*—One-Point, *Mutation - Bit Negator* (5%).

Obtained results are summarized in Tables 2 and 3. The performance of the configurations on specific datasets can be found in Table 2; *Design Accuracy* shows the average rate (and standard deviation) of proper classification of the fittest network obtained with the Design set, while *Test Accuracy* indicates the performance of such network on the Test set; the best values are indicated in boldface. Higher values are better, being 1.0 the best possible solution. Table 3 compiles important traits of the generated networks: the average amount of Input Features actually employed, and its proportion regarding the size of the Input vector, the average amount of hidden units, and the average amount of synapses.

4.1 Statistical Tests

A further analysis of the obtained data was performed to endow statistical reliance to the results. Assuming normality of the data (i.e. asserting that Normal Distributions can model the obtained data), a Shapiro-Wilk [30] test was applied. This test stated an important insight since it proved the normality of data, confirming the assumptions previously made. Then, a significant difference between the behaviour of the configurations was expected to be denoted, and so a *Tukey HSD* [22] was applied assuming that data belonged to the same Normal Distribution. Table 4 shows the results obtained by the *Tukey HSD* test on configurations *A* and *B*; the adjusted-p-value is enough to reject the assumption of the test, with a significance value of 0.05, ultimately evidencing a statistically-trustworthy difference between the configurations. As a complement of the box plots in Figs. 5 and 6, this analysis led to resolution that configuration *B* provides better results than configuration *A* when compared along.

Table 2 Design and test accuracies for configurations *A* and *B* on all considered datasets

Dataset	Configuration	Design accuracy	Test accuracy
Balance scale	<i>A</i>	0.7486 ± 0.0525	0.7219 ± 0.0629
	<i>B</i>	0.7331 ± 0.0718	0.6944 ± 0.0752
Breast cancer	<i>A</i>	0.9494 ± 0.0141	0.9418 ± 0.0238
	<i>B</i>	0.9474 ± 0.0121	0.9405 ± 0.0151
Glass	<i>A</i>	0.2549 ± 0.1345	0.2404 ± 0.1433
	<i>B</i>	0.4288 ± 0.0673	0.4002 ± 0.0590
Ionosphere	<i>A</i>	0.8549 ± 0.0374	0.8374 ± 0.0295
	<i>B</i>	0.8543 ± 0.0537	0.8137 ± 0.0708
Iris Plant	<i>A</i>	0.8857 ± 0.1111	0.8663 ± 0.1269
	<i>B</i>	0.9733 ± 0.0157	0.9382 ± 0.0217
Wine	<i>A</i>	0.6881 ± 0.1549	0.6415 ± 0.1551
	<i>B</i>	0.7375 ± 0.1126	0.6816 ± 0.1098

The best results are indicated in boldface

Table 3 Network traits for configurations *A* and *B* on every dataset

Configuration	Dataset	Average # features employed	Features employed (%)	Average# hidden units	Average # synapses
<i>A</i>	Balance scale	2.97 ± 0.67	0.74	1.58 ± 0.85	9.58 ± 2.89
	Breast cancer	2.52 ± 0.99	0.28	2.03 ± 1.09	8.79 ± 2.04
	Glass	2.21 ± 1.32	0.25	1.48 ± 0.93	11.73 ± 3.77
	Ionosphere	2.24 ± 1.05	0.07	2.21 ± 1.30	7.61 ± 2.81
	Iris plant	1.30 ± 0.52	0.33	1.70 ± 1.00	8.18 ± 2.35
	Wine	2.48 ± 1.23	0.19	1.97 ± 1.75	9.73 ± 4.48
<i>B</i>	Balance Scale	3.52 ± 0.56	0.88	4.09 ± 1.58	12.36 ± 4.32
	Breast cancer	6.12 ± 1.47	0.68	4.03 ± 1.85	14.24 ± 5.46
	Glass	5.70 ± 1.62	0.63	3.03 ± 1.82	11.88 ± 6.30
	Ionosphere	5.52 ± 2.24	0.17	3.55 ± 2.05	12.15 ± 6.68
	Iris plant	3.39 ± 0.89	0.85	4.73 ± 2.60	13.27 ± 6.93
	Wine	6.15 ± 1.88	0.47	3.82 ± 1.49	13.03 ± 4.93

Table 4 Tukey HSD test with Bonferroni correction for *A* and *B* configurations

Tukey HSD	Diff	Lower	Upper	Adjusted p-value
<i>A</i> versus <i>B</i>	0.014	0.0030	0.026	0.0078

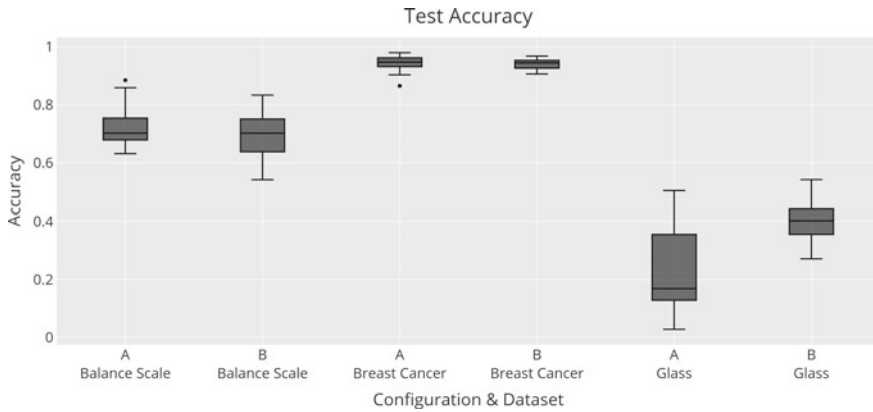


Fig. 5 Box plots of the test accuracy of both considered configurations on the balance scale, breast cancer and glass datasets

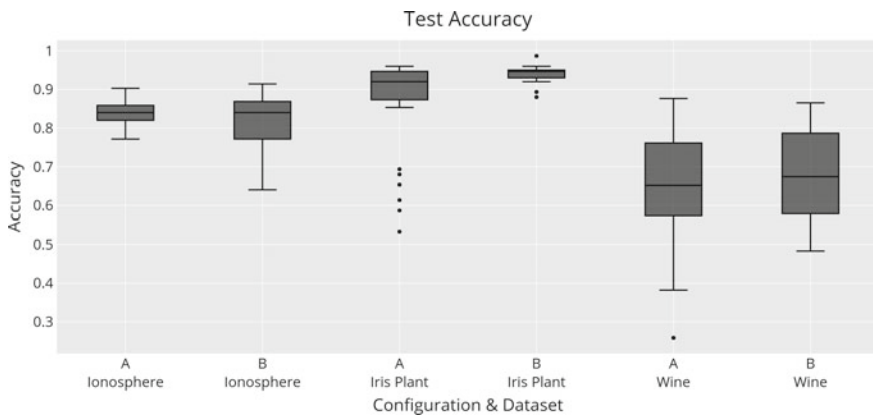


Fig. 6 Box plots of the test accuracy of both considered configurations on the ionosphere, iris plant and wine datasets

5 Conclusions and Future Work

A framework to design partially-connected ANNs of second and third generation based on Grammatical Evolution to solve supervised classification problems was presented in this paper.

Experimentation results and further statistical analysis confirmed an improvement on the classification performance as the neural model was adjusted on the configurations; third-generation networks displayed indeed their attributed enhancements. Evolutionary Design led to the formulation of solution topologies with fewer connections than those in equivalent fully-connected topologies hence reducing the complexity of the networks, and the absence of an explicit learning process averted

the issues related to it. Several variations of the framework might be considered for future work; varying the search engine, the neural model, or the fitness function may benefit the performance.

Acknowledgements Authors wish to thank *National Technology of Mexico* and *University of Guanajuato*. G. López-Vázquez and A. Rojas-Domínguez thank to the *National Council of Science and Technology of Mexico* (CONACYT) for the support provided by means of the Scholarship for Postgraduate Studies (701071) and Research Grant (CÁTEDRAS-2598), respectively. This work was supported by the CONACYT Project FC2016-1961 “Neurociencia Computacional: de la teoría al desarrollo de sistemas neuromórficos”.

References

1. Amaldi, E., Mayoraz, E., de Werra, D.: A review of combinatorial problems arising in feed-forward neural network design. *Discrete Appl. Math.* **52**(2), 111–138 (1994)
2. Belatreche, A., Maguire, L.P., McGinnity, M., Wu, Q.X.: An evolutionary strategy for supervised training of biologically plausible neural networks. In: *Proceedings of the Sixth International Conference on Computational Intelligence and Natural Computing*, pp. 1524–1527 (2003)
3. Belatreche, A., Maguire, L.P., McGinnity, T.M.: Advances in design and application of spiking neural networks. *Soft Comput.* **11**(3), 239–248 (2007)
4. Dheeru, D., Karra Taniskidou, E.: *UCI Machine Learning Repository* (2017)
5. Ding, S., Li, H., Su, C., Yu, J., Jin, F.: Evolutionary artificial neural networks: a review. *Artif. Intell. Rev.* **39**(3), 251–260 (2013)
6. Elfving, S., Uchibe, E., Doya, K.: Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Netw.* (2018)
7. Elizondo, D., Fiesler, E.: A survey of partially connected neural networks. *Int. J. Neural Syst.* **8**(5–6), 535–558 (1997)
8. Espinal, A., Carpio, M., Ornelas, M., Puga, H., Melin, P., Sotelo-Figueroa, M.: Comparing metaheuristic algorithms on the training process of spiking neural networks. In: *Recent Advances on Hybrid Approaches for Designing Intelligent Systems*, pp. 391–403. Springer (2014)
9. Espinal, A., Carpio, M., Ornelas, M., Puga, H., Melin, P., Sotelo-Figueroa, M.: Developing architectures of spiking neural networks by using grammatical evolution based on evolutionary strategy. In: *Mexican Conference on Pattern Recognition*, pp. 71–80. Springer (2014)
10. Gerstner, W., Kistler, W.: *Spiking Neuron Models: Single Neurons, Populations, Plasticity*. Cambridge University Press (2002)
11. Ghosh-Dastidar, S., Adeli, H.: Spiking neural networks. *Int. J. Neural Syst.* **19**(04), 295–308 (2009)
12. Gnedenko, B.V., Kolmogorov, A.N.: Limit distributions for sums of independent random variables. In: *Predelnye raspredeleniia dlia summ*, No. ix, 264 p. Addison-Wesley Pub. Co., Cambridge, Mass (1954)
13. Hodgkin, A.L., Huxley, A.F.: A quantitative description of membrane current and its application to conduction and excitation in nerve. *J. Physiol.* **117**(4), 500 (1952)
14. Ijspeert, A.J.: Central pattern generators for locomotion control in animals and robots: a review. *Neural Netw.* **21**(4), 642–653 (2008)
15. Judd, J.S.: On the complexity of loading shallow neural networks. *J. Complex.* **4**(3), 177–192 (1988)
16. Judd, J.S.: *Neural Network Design and the Complexity of Learning*. Neural Network Modeling and Connectionism Series. MIT Press, Cambridge, MA (1990)
17. Lapique, L.: Recherches quantitatives sur l’excitation électrique des nerfs traitée comme une polarization. *Journal de Physiologie et de Pathologie Generalej* **9**, 620–635 (1907)

18. Maass, W.: Networks of spiking neurons: the third generation of neural network models. *Neural Netw.* **10**(9), 1659–1671 (1997)
19. Maass, W., Schmitt, M.: On the complexity of learning for spiking neurons with temporal coding. *Inf. Comput.* **153**(1), 26–46 (1999)
20. Markou, M., Singh, S.: Novelty detection: a review-part 2: neural network based approaches. *Sig. Process.* **83**(12), 2499–2521 (2003)
21. McCulloch, W.S., Pitts, W.: A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biology* **5**(4), 115–133 (1943)
22. Montgomery, D.C.: *Design and Analysis of Experiments* (2013)
23. Ojha, V.K., Abraham, A., Snášel, V.: Metaheuristic design of feedforward neural networks: a review of two decades of research. *Eng. Appl. Artif. Intell.* **60**, 97–116 (2017)
24. O’Neill, M., Ryan, C.: Grammatical evolution. *Trans. Evol. Comp.* **5**(4), 349–358 (2001)
25. Quiroz-Ramírez, O., Espinal, A., Ornelas-Rodríguez, M., Rojas-Domínguez, A., Sánchez, D., Puga-Soberanes, H., Carpio, M., Espinoza, L.E.M., Ortíz-López, J.: Partially-connected artificial neural networks developed by grammatical evolution for pattern recognition problems. *Stud. Comput. Intell.* **749**, 99–112 (2018)
26. Rosenblatt, E.: *The Perceptron, A Perceiving And Recognizing Automaton (Project PARA)*. Cornell Aeronautical Laboratory (1957)
27. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back-propagating errors. *Nature* **323**(6088), 533 (1986)
28. Ryan, C., Collins, J., O’Neill, M.: Grammatical evolution: evolving programs for an arbitrary language. In: *Proceedings of Genetic Programming: First European Workshop. EuroGP’98*, Paris, France, 14–15 April 1998, pp. 83–96. Springer, Berlin, Heidelberg (1998)
29. Scarselli, F., Tsoi, A.C.: Universal approximation using feedforward neural networks: a survey of some existing methods, and some new results. *Neural Netw.* **11**(1), 15–37 (1998)
30. Shapiro, S.S., Wilk, M.B.: An analysis of variance test for normality (complete samples). *Biometrika* **52**(3–4), 591–611 (1965)
31. Talbi, E.-G.: *Metaheuristics: from Design to Implementation*. Wiley, Hoboken, NJ, (2009). OCLC: ocn230183356
32. Yao, X.: Evolving artificial neural networks. *Proc. IEEE* **87**(9), 1423–1447 (1999)
33. Yu, J., Tan, M., Chen, J., Zhang, J.: A survey on CPG-inspired control models and system implementation. *IEEE Trans. Neural Netw. Learning Syst.* **3**, 441–456 (2014)
34. Zhang, G.P.: Neural networks for classification: a survey. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **30**(4), 451–462 (2000)

Gegenbauer-Based Image Descriptors for Visual Scene Recognition



Antonio Herrera-Acosta, A. Rojas-Domínguez, J. Martín Carpio, Manuel Ornelas-Rodríguez and Héctor Puga

Abstract Visual scene recognition is an important problem in artificial intelligence with applications in areas such as autonomous vehicles, visually impaired people assistance, augmented reality, and many other pattern recognition areas. Visual scene recognition has been tackled in recent years by means of image descriptors such as the popular Speeded-Up Robust Features (SURF) algorithm. The problem consists in analyzing the scenes in order to produce a compact representation based on a set of so called regions of interest (ROIs) and then finding the largest number of matches among a dataset of reference images that include non-affine transformations of the scenes. In this paper, a new form of descriptors based on moment invariants from Gegenbauer orthogonal polynomials is presented. Their computation is efficient and the produced feature vector is compact, containing only a couple dozens of values. Our proposal is compared against SURF by means of the recognition rate computed on a set of two hundred scenes containing challenging conditions. The experimental results show no statistically significant difference between the performances of the descriptors.

Keywords Local image descriptors · Orthogonal polynomials · Gegenbauer polynomials · Visual scene recognition · Image moment invariants

A. Herrera-Acosta · A. Rojas-Domínguez (✉) · J. M. Carpio · M. Ornelas-Rodríguez · H. Puga
Tecnológico Nacional de México—Instituto Tecnológico de León, Av. Tecnológico S/N, Fracc,
Industrial Julián de Obregón, León 37290, Guanajuato, Mexico
e-mail: Alfonso.Rojas@gmail.com

A. Herrera-Acosta
e-mail: heracost@gmail.com

J. M. Carpio
e-mail: juanmartin.carpio@itleon.edu.mx

M. Ornelas-Rodríguez
e-mail: manuel.ornelas@itleon.edu.mx

H. Puga
e-mail: pugahector@yahoo.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_43

1 Introduction

Nowadays, artificial intelligence (AI) is employed to solve everyday problems such as automated object recognition, guidance of autonomous robots, people tracking, etc. [1, 2]. These problems call for the development of useful image processing and pattern recognition techniques that can extract and employ information derived from the appearance of the objects and scenes depicted in digital images that are processed by the AI software. Appearance based methods rely on the extraction of global and local characteristics or features; local features are more desirable, because it is more efficient to employ only certain distinctive regions of a digital image than a representation of the whole image. Furthermore, many of the modern applications of AI occur within uncontrolled environments, such as outdoors, where illumination may change rapidly, as well as other scenarios where transformations such as scale changes and changes in camera point of view are commonly present. A set of well-designed local features is more robust to these transformations than a single local feature. Thus, local feature descriptors such as the Scale Invariant Feature Transform (SIFT) and its optimized version, the Speeded-Up Robust Features (SURF) have been developed [3].

The SIFT algorithm is one of the most popular, since the descriptors generated by this algorithm are, to certain degree, invariant to scale, rotation, illumination, and other related transformations. Because of this robustness to several conditions, the SIFT is usually considered a benchmark against which other methods are compared. SIFT generated descriptors are composed of 128 attributes corresponding to 8-bin histograms of the gradient orientations in the image [4, 5]. Although SIFT descriptors are highly effective, their computation may be too expensive for real-time applications. Because of this, the SURF algorithm was developed as an approximate version of the SIFT descriptor. The SURF algorithm is able to automatically detect and extract ROIs in images, such as faces, objects, corners, etc. [4]. The detection of ROIs will be further discussed in subsequent sections in this paper.

Many other local descriptors exist, such as the Daisy descriptor, which is inspired by SIFT and the Gradient Location and Orientation Histogram (GLOH) and is said to be 40% faster than SIFT [4, 6]. The descriptor calculates a set of eight orientation maps based on a bank of oriented derivative-of-Gaussian filters [4]. However, instead of detecting ROIs, this algorithm generates a descriptor for each pixel in an image. The process becomes costly for medium and high resolution images, and thus, this feature extraction process is not very attractive. The Pyramid Histogram of Oriented Gradients (PHOG) algorithm was designed for global- and local-region-based image classification, as well as for local feature detection. This applies the Histogram of Gradients (HOG) technique to the complete image and also to the image divided in regions following a pyramid structure with several levels [7]. A compromise between a small amount of levels (efficiency) and a good image representation (accuracy) must be searched for, and thus, a more flexible and automated mechanism for ROI selection such as that of the SURF algorithm is preferred.

In this work, a new proposal of local image feature descriptors based on Gegenbauer moments [8] and computed from automatically extracted ROIs (by means of the SURF algorithm) is presented. From the moment of their introduction by Hu in 1962, moments, moments functions, and invariant moments have been widely used because of their ability to represent global features of an image. They have been used in areas such as: pattern recognition, image processing, artificial vision, etc. Among all the existing moments, the geometric moments are the most used, despite these not being orthogonal. The orthogonal moments are very attractive because these allow one to extract image features with minimal information redundancy [9].

The rest of this paper is organized as follows: Sect. 2 provides theoretical background and contains our proposal. Section 3 presents our experimental methodology. In Sect. 4 our experimental results are reported and discussed. Finally, conclusions and directions for future work are offered in Sect. 5.

2 Theoretical Background and Proposal

In this section, basic theoretical concepts needed, including: Gegenbauer polynomials; Gegenbauer-based image moments; and moment invariants, are reviewed. Afterwards, our proposal of a novel descriptor is presented.

2.1 Gegenbauer Polynomials

The Gegenbauer polynomials belong to the category of orthogonal polynomials within a square (they are orthogonal in a square between $[-1, 1]$) and they are characterized by a degree and a parameter α , known as a scale factor, which distinguishes one polynomial from another one of the same degree [10]. The Gegenbauer polynomials $G_n^{(\alpha)}(x)$ can be computed by several means, such as through the following recurrence relation:

$$G_{n+1}^{(\alpha)}(x) = \frac{2x}{n} \left[(n + \alpha - 1)G_{n-1}^{(\alpha)}(x) - (n + 2\alpha - 2)G_{n-2}^{(\alpha)}(x) \right] \quad (1)$$

where n is the polynomial's degree, and $n > 1$. The polynomials of degrees $n = 0$ and $n = 1$ are: $G_0^\alpha(x) = 1$; $G_1^\alpha(x) = 2\alpha x$ for $\alpha > -0.5$. The Gegenbauer polynomials are orthogonal with respect to a weight function:

$$w^{(\alpha)}(x) = (1 - x^2)^{\alpha-0.5} \quad (2)$$

As said before, the α parameter determines the shape of the polynomials to be produced, such that other known families of polynomials contained by the Gegenbauer

family can be obtained; e.g. $\alpha = 0.5$ produces Legendre polynomials and $\alpha = 1$ produces Chebyshev polynomials of the second kind [11, 12].

2.2 Gegenbauer Polynomials-Based Image Moments

In the formulation of Gegenbauer image moments, the computation of the polynomials by means of (1) is not convenient because the polynomial coefficients cannot be manipulated separately. An alternative is to employ:

$$G_n^{(\alpha)}(x) = \sum_{\substack{k=0 \\ \binom{n-k}{\text{even}}}}^n B_{n,k}^{(\alpha)} x^k \tag{3}$$

where $B_{n,k}^{(\alpha)}$ represent the Gegenbauer coefficients, given by:

$$B_{n,k}^{(\alpha)} = (-1)^{\frac{(n-k)}{2}} \frac{\Gamma(\alpha + \frac{n+k}{2}) 2^k}{k! \binom{n-k}{2}! \Gamma(\alpha)} \tag{4}$$

Use of (4) is costly, due to the factorials and Gamma function involved. However, the following equations allow one to obtain these recursively:

$$\begin{aligned} B_{0,0}^{(\alpha)} &= 1 \\ B_{n,n}^{(\alpha)} &= \frac{2(\alpha+n-1)}{n} B_{n-1,n-1}^{(\alpha)} \\ B_{n,n-t}^{(\alpha)} &= -\frac{(n-t+1)(n-t+2)}{2t(\alpha+n-\frac{t}{2})} B_{n,n-t+2}^{(\alpha)} \end{aligned} \tag{5}$$

where $t = 2, 4, 6, \dots, n$. Besides the polynomial coefficients, scale factors or normalization constants $C_n(\alpha)$ are required. These are defined as follows:

$$C_n \equiv C_n(\alpha) = \frac{2\pi \Gamma(n + 2\alpha)}{2^{2\alpha} n!(n + \alpha) [\Gamma(\alpha)]^2} \tag{6}$$

With this, one is able to define the Gegenbauer two dimensional moments (or Gegenbauer image moments) of order (n, m) as [8]:

$$A_{n,m} = \frac{1}{C_n C_m} \int_{-1}^1 \int_{-1}^1 f(x, y) G_n^{(\alpha)}(x) G_m^{(\alpha)}(y) w^{(\alpha)}(x) w^{(\alpha)}(y) dx dy \tag{7}$$

where n, m are nonnegative integers.

2.3 Invariants of Gegenbauer-Based Moments

In order to be able to take advantage of the well-known regular moment invariant features, Hosny [8] shows that (7) can be modified to produce Gegenbauer invariant moments $\hat{A}_{n,m}$ as follows:

$$\hat{A}_{n,m}(f, \alpha) = \frac{1}{C_n C_m} \sum_{\substack{k=0 \\ \text{even}}}^{\lfloor n/2 \rfloor} \sum_{\substack{l=0 \\ \text{even}}}^{\lfloor m/2 \rfloor} B_{n,k}^{(\alpha)} B_{m,l}^{(\alpha)} I_{k,l}(f) \quad (8)$$

where $I_{k,l}$ are the regular moment invariants of order (k, l) of image f . These can be computed through the traditional regular moments over which invariance to translation, rotation and scale can be introduced [13]. The regular moments are:

$$M_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} i^p j^q f(i, j) \quad (9)$$

where (i, j) are the indexes or coordinates of a pixel within the analyzed image $f(i, j)$ and (p, q) are the corresponding moment orders per dimension. In order to introduce the invariance to translation, it is necessary to first find the image centroid (x_c, y_c) , which is obtained using (9) to compute moments: M_{00} , M_{01} and M_{10} and then applying:

$$x_c = \frac{M_{10}}{M_{00}}, y_c = \frac{M_{01}}{M_{00}} \quad (10)$$

Once the image centroid has been obtained, the central moment, or moment with invariance to translation is computed as [13]:

$$\mu_{pq} = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} (i - x_c)^p (j - y_c)^q f(i, j) \quad (11)$$

A second invariant feature that can be added to a descriptor through the central moments is invariance to scale, but it is necessary to know the scale factor between scenes. For details, see Sonka et al. [13]. In our application, the scale factor is unknown, and because of this, scale invariance is not introduced. However, it has been observed that the descriptors show a certain degree of natural robustness to scale variation (see Sect. 4).

Rotation invariance is another of the features that can be introduced by choosing a coordinate system so that $\mu_{11} = 0$ [13]. An alternative way is to eliminate rotation of the images by means of the Gegenbauer moments $A_{0,1}$ and $A_{1,0}$. This is illustrated

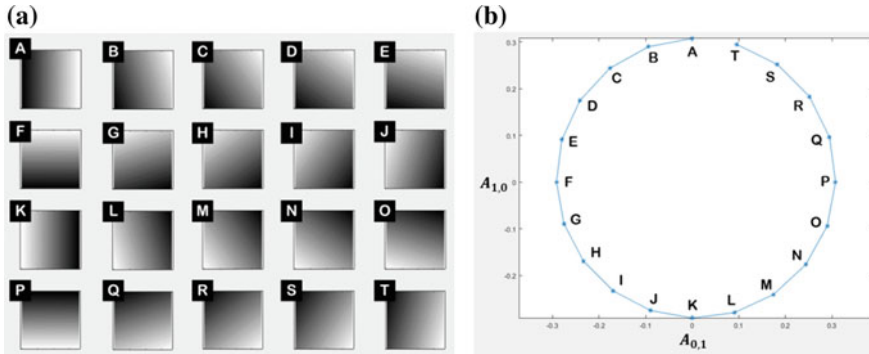


Fig. 1 a The relative degree of rotation between two images is found in the relationship between the corresponding moments $A_{0,1}$ and $A_{1,0}$ taken as coordinates (b)

in Fig. 1 as follows: a synthetic test image (a simple 2D gradient) is shown with different degrees of rotation, with a rotation step of 18° between them and identified with the sequence A to T (Fig. 1a). Plotting the corresponding Gegenbauer moments $A_{0,1}$ versus $A_{1,0}$ shows that the relative rotation between the different versions of the test image can be readily observed from the moment pairs (Fig. 1b). Thus, once the relative rotation between two images is known, the rotation can be eliminated by transforming any of the two images accordingly. In this work, rotation invariance was not introduced for simplicity of the experimental conditions and because a high degree of rotation is not normally expected in our application.

2.4 Gegenbauer Polynomials-Based Image Descriptors

Once the required theoretical background regarding Gegenbauer polynomials, moments and moment invariants has been covered, we are in a position to define a Gegenbauer-based image descriptor, *GID*, as a vector formed by a concatenation of Gegenbauer invariant moments computed according to (8). Formally:

$$GID(f, \alpha) = \left\{ \hat{A}_{n,m}(f, \alpha) \mid \forall n, m \in \mathbb{Z}^+; \alpha > 0.5 \right\} \quad (12)$$

In this paper, for simplicity the scale parameter $\alpha = 1$. In practice, only a few moments need to be included as part of the proposed descriptor. Refer to Fig. 2 and notice that each of the moments $\hat{A}_{n,m}$ corresponds to a 2D kernel $G_{n,m}$ depicted in this figure. The 2D kernels arise from the convolution of two one-dimensional Gegenbauer polynomials. The $G_{0,0}$ kernel is used to compute the average value of the image, and in general, provides little discriminative power (recall that the polynomials of degree zero are simply the constant value 1). Furthermore, the topmost row and leftmost column of kernels in Fig. 2 depict 2D kernels where one of the polynomials is

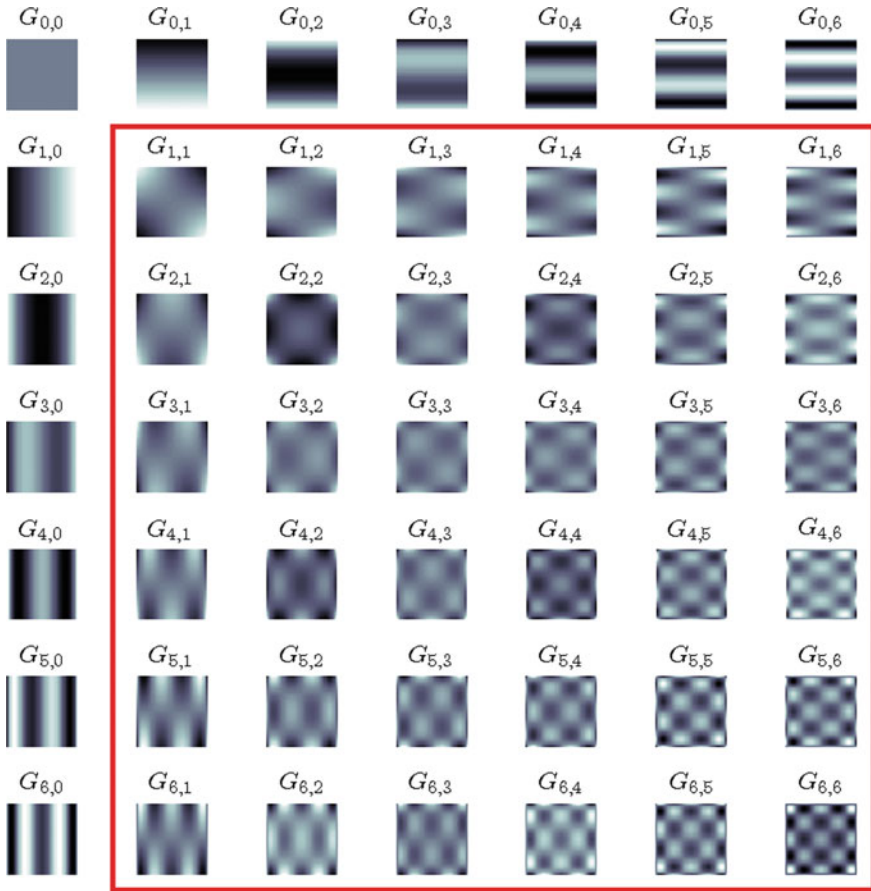


Fig. 2 Two dimensional kernels from convolution of two one-dimensional Gegenbauer polynomials. Kernels in the square are used to form the *GID* descriptor

a zero degree polynomial. Moments derived from these kernels are useful for defining global characteristics of an image, such as its orientation or translation (recall the information provided in Sect. 2.3), but previous experiments tell us that these are not very good features for image recognition, that require a higher level of detail. For this reason, all of these moments are excluded from the definition of the *GID* descriptor in (12).

In this study, only the moments corresponding to the 2D kernels enclosed in the square in Fig. 2 were employed to construct the *GID* descriptor. The Gegenbauer invariant moments are computed over intensity (i.e. grayscale images, Fig. 3a). Since the phase (the orientation of the intensity gradients) often possesses useful information, the *GID* descriptor is alternatively computed on phase images, obtained by convolution of the grayscale images with Sobel filters [14] (Fig. 3b).

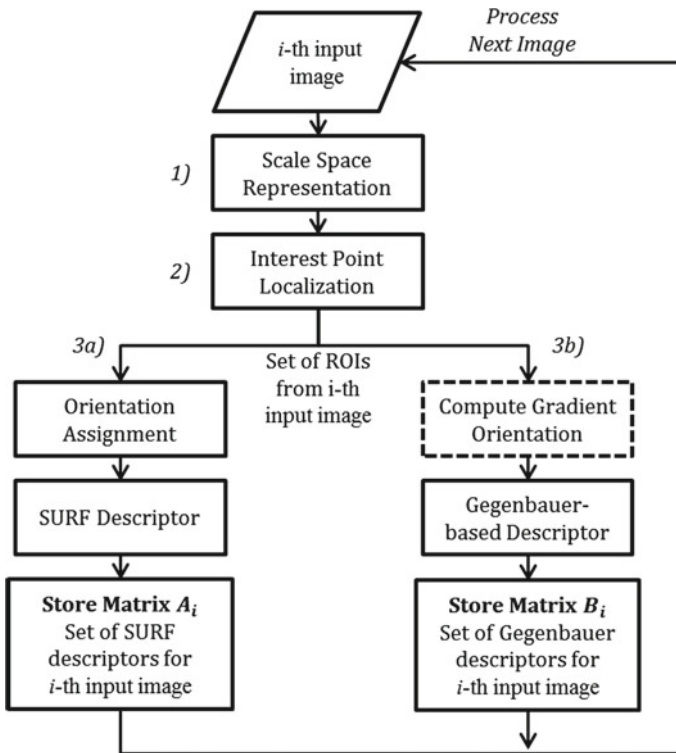


Fig. 3 Extraction of the descriptors. Either the SURF descriptor or the Gegenbauer-based descriptors are extracted at step 3. The dashed box represents an alternative process

3 Experimental Methodology

3.1 Key-Point and ROI Extraction

As mentioned before, the SURF algorithm contains a method for selecting local ROIs in an automated fashion. The main point of this method is to detect repeatable ROIs, in other words, that the same ROIs are found in different images depicting the same scene regardless of the possible changes these have suffered. The algorithm for automated detection of ROIs can be roughly divided into three stages: Scale Space Representation; Interest Point Localization; and Orientation Assignment. A fourth step is reserved for the construction of the descriptor of each ROI [15]. In this paper, we follow the first two stages in order to automatically obtain robust ROIs on which a fair comparison of descriptors can be carried out. Then, the SURF descriptor and the *GID* are extracted separately on the same set of ROIs, so that the comparison uses exactly the same input information. Next, the stages for ROI detection are briefly described (Fig. 3).

Scale Space Representation. To robustly determine if a region should be considered as a ROI, it is necessary to look at the images across different scales. For this purpose, a fixed pyramidal representation is used, called the Scale Space. This space is built by application of Laplacian-of-Gaussian (LoG) filters of different sizes on the input image, obtaining a set of response images. For more details refer to Bay et al. [15].

Interest Point Localization. Using the Scale Space representation, the algorithm locates the points of maximum response to the LoG filters. To declare a point as a maximum response point, each candidate point is compared against a $3 \times 3 \times 3$ neighborhood [15], i.e., it is compared against the 27 surrounding pixels in a cube formed in 3D defined by the horizontal position (x), vertical position (y) and scale (s). If the point is indeed maximum and its value is above a predefined threshold, then an interest point or keypoint is recorded at position x, y, s . Keypoints thus found become the centers of the ROIs that have a particular size, given by their corresponding scale, s .

3.2 Construction of the Descriptors

SURF descriptor of a ROI. Once a ROI has been located, a dominant orientation is determined for identification and to provide rotation invariance. The dominant orientation is computed based on the image gradients. See details in Bay et al. [15]. Guided by the dominant orientation of a ROI, a 4×4 square grid centered on the corresponding keypoint is built over the ROI. Each of the 16 cells so defined is further subdivided into 4 subsections, creating an array of 64 subdivisions. From these, a 64-attribute descriptor is constructed, where the attributes are the sums (per subdivision) of the derivatives in the x and y directions as well as their absolute values: $\Sigma dx, \Sigma |dx|, \Sigma dy$ and $\Sigma |dy|$ [15]. The descriptors corresponding to a set of ROIs from a particular image are stored in one matrix, one descriptor of one ROI per row (see Fig. 3).

Gegenbauer-based descriptor of a ROI. Based on a ROI the *GID* descriptors are computed according to the procedure described in Sect. 2 and in particular, (12). As mentioned earlier, the descriptors are obtained with base on two types of information: the gray scale intensity values and the gradient orientations, separately; the descriptors of each type are also evaluated independently. As with SURF, the *GID* descriptors corresponding to a set of ROIs from a particular image are stored in one matrix, the descriptor of one ROI per row. In the end, for a dataset of N images, N such matrices are produced.

3.3 Experimental Evaluation

The described descriptors (SURF, Gegenbauer from intensity and Gegenbauer from gradient orientation) are used to match the different ROIs extracted from the images

in our test dataset. The matching is carried out with base on a similarity measure $Sim(\mathbf{u}, \mathbf{v})$ that is a function of a pair of descriptors (\mathbf{u}, \mathbf{v}) of a kind:

$$Sim(\mathbf{u}, \mathbf{v}) = \exp\left(-\kappa \sum_{d=1}^D \{(\mathbf{u}_d - \mathbf{v}_d)^2 / (|\mathbf{u}_d| + |\mathbf{v}_d| + \epsilon)\}\right) \quad (13)$$

where κ is a constant used to shape the similarity function and ϵ is a small constant used to prevent a division by zero.

A degree of matching between each image and the rest of the images in the test dataset (with the exception of the same image) is computed. This is done by measuring the similarity between each of the rows of a descriptor matrix from one image against each of the rows of the descriptor matrix from the second image. This experimental procedure is better explained through the pseudocode shown below (Algorithm 1). Many of the descriptors will show a certain degree of similarity, but only the maximum similarity is recorded and if this is above a user-defined recognition sensitivity parameter, δ , a match between the ROIs of the different images is counted. Next, the total amount of ROI-based matches per image is computed, and a hit (a correct correspondence between two images depicting the same scene) is recorded whenever the count of ROI-based matches is the maximum between the pair of images depicting the same scene. Finally, the number of hits is presented as a detection rate associated with the corresponding descriptor. Since the recognition threshold δ is a free parameter, in our experiments we obtained the detection rate of each descriptor type for a range of threshold values.

Algorithm 1. Descriptor extraction and computation of ROIs similarity	
1	Compute descriptors of type "X" for all N images // see Fig. 3
2	For $j = 1$ to N
3	$A_j \leftarrow$ descriptors of image j
4	For $k = 1$ to N
5	If $(j \neq k)$ // do not compare an image against itself
6	$B_k \leftarrow$ descriptors of image k
7	For $l = 1$ to L (number of rows in A_j)
8	$\mathbf{u} \leftarrow$ l -th row of A_j
9	For $m = 1$ to M (number of rows in B_k)
10	$\mathbf{v} \leftarrow$ m -th row of
11	$S(l, m) \leftarrow$ similarity(\mathbf{u}, \mathbf{v}). // Eq. (13)
12	End For
13	End For
14	Store similarity matrix between ROIs of images j and k.
15	End If
16	End For
17	End For



Fig. 4 A selection of test images. The images are organized in pairs, each depicting a local scene. Non-affine transformations are present between the images in each pair

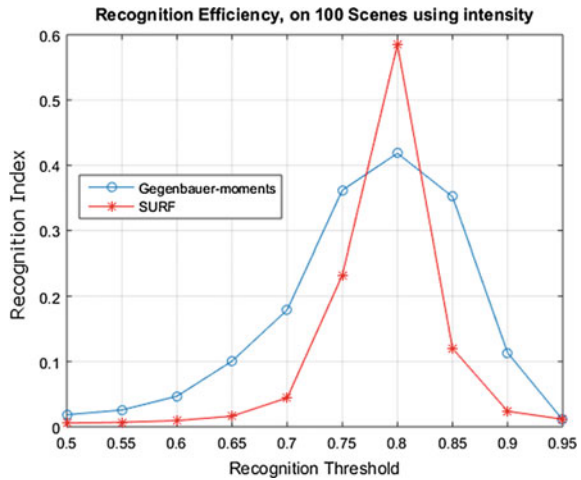
The image bank used to elaborate the experimentation contains a set of 400 images (200 pairs, each pair depicting the same scene with some variations) of 400×300 pixels each; the images were acquired in uncontrolled environments, most of them outdoors under sunlight. The images capture urban art, natural textures, architectural works, objects, etc. The images were categorized into two groups: one where 100 pairs show small changes between the images of a pair, such as translation, illumination changes and little viewpoint changes; in the second group 100 pairs of images present more drastic transformations, such as more significant viewpoint changes, translations combined with scaling, non-affine transformations, stronger illumination changes, etc.

A small selection of the images employed in our experimental evaluation is shown in Fig. 4 for illustrative purposes. The images were acquired in color, but are processed by the algorithms as single-channel (grayscale) images. Some preprocessing is used to prepare the images for descriptor extraction, including symmetrical padding applied to avoid information loss at the margins of the images and slight smoothing through Gaussian filtering before gradient computation.

4 Results and Discussion

Four different experiments were carried out for the purpose of comparatively evaluate the proposed *GID* against the well-known SURF descriptors. In the first experiment, the two types of descriptors were extracted from the group of 100 image pairs in our image bank that present only slight transformations between image pairs corresponding to the same scene. The intensity of the images was used as the input information for the *GID* and the recognition threshold δ was varied in the range 0.5–0.95 in steps of 0.05 units. The detection rate was obtained for each of these values of δ and plotted. The results are shown in Fig. 5.

Fig. 5 Plot of recognition rate as a function of recognition threshold. Experiment 1—100 scenes using image intensity for *GID*



In the second experiment, the conditions were the same as in the first experiment except that this time the 200 image pairs in our image bank (including 100 pairs with slight transformations and 100 with more drastic transformations) were used. The results are shown in Fig. 6.

In the third experiment, the descriptors were extracted from the group of 100 image pairs that present only slight transformations. The phase (gradient orientation) of the images was used as the input for the *GID*. Recognition results are shown in Fig. 7.

Finally, in the fourth experiment the conditions were the same as in the third experiment, except that the descriptors were extracted from the 200 image pairs in the test image bank. Recognition results are shown in Fig. 8.

Fig. 6 Plot of recognition rate as a function of recognition threshold. Experiment 2—200 scenes using image intensity for *GID*

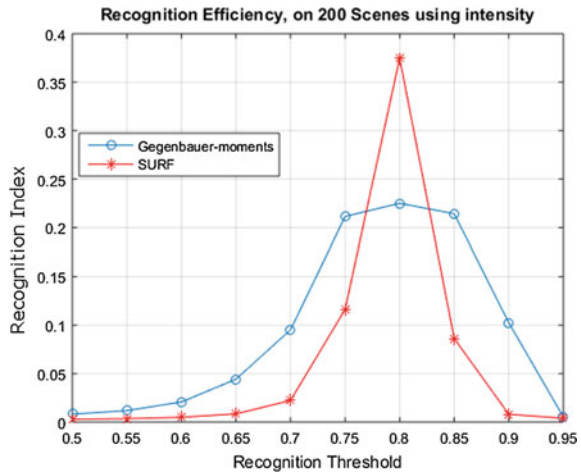


Fig. 7 Plot of recognition rate as a function of recognition threshold. Experiment 3—100 scenes using image phase for *GID*

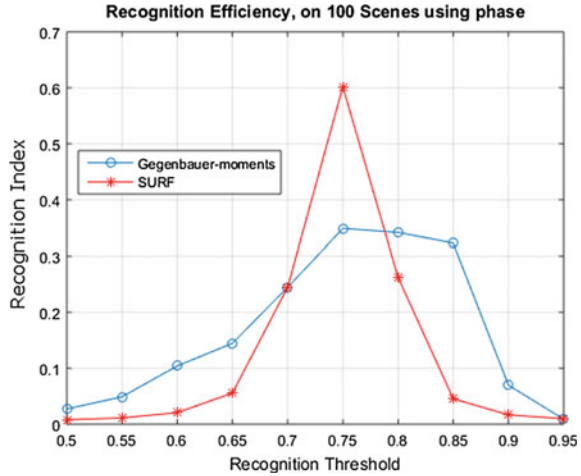
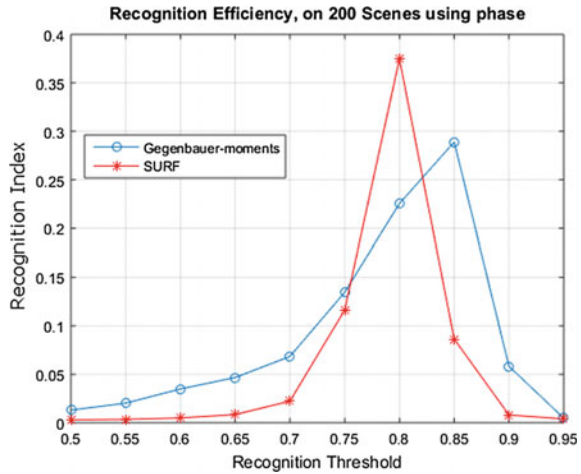


Fig. 8 Plot of recognition rate as a function of recognition threshold. Experiment 4—200 scenes using image phase for *GID*



In general, the following observations can be made from the results provided in Figs. 5, 6, 7 and 8. First, the behavior of the descriptors is just what can be expected: for small values of the recognition threshold δ , the recognition rate is quite poor. This is because a small recognition threshold causes the algorithms to produce many false matches. As the recognition threshold is increased, the matches between ROIs represent increasingly better the true matches between image pairs of the same scene and the recognition rate rises. However, after reaching a maximum at around $\delta = 0.8$, the recognition rate drops again. The explanation is that the recognition threshold becomes too high, causing very few matches between ROIs.

A second general observation is that, across the whole range of values of the recognition threshold, the recognition rate achieved on the whole image bank is about

Table 1 Wilcoxon rank-sum test results

Experiment	Feature for <i>GID</i>	Dataset	<i>p</i> -value	Test result
1	Intensity	100 scenes	0.140	Do not reject H_0
2	Intensity	200 scenes	0.121	Do not reject H_0
3	Phase	100 scenes	0.212	Do not reject H_0
4	Phase	200 scenes	0.140	Do not reject H_0

half of that achieved on the group of images that present only slight transformations between image pairs of a scene. This difference is due to two factors: clearly, when the image transformations are more significant, the recognition rate suffers because the descriptors are not robust enough to deal with that degree of discrepancy between images of a scene. The second factor is that there is twice the number of possible matches to be made.

Finally, the third observation is that in all of the experiments an interesting pattern was identified: for all but one of the values of the recognition threshold, the *GID*s achieve a higher recognition rate than the SURF descriptors. The exception is always the point of peak recognition rate, where SURF always overcome the *GID*. An interpretation is that the SURF descriptors are quite sensitive to the value of the recognition threshold; inside a small range of threshold values the descriptors work very well, but their performance drops dramatically for threshold values outside of that range. The *GID* is less sensitive to threshold values; the transition from the region of lower recognition rates to that of higher recognition rates is smoother, but the peak recognition rate is not as good as that of SURF. Also, there were no substantial differences between using the image intensity and the phase for computation of the *GID*.

In order to test the statistical significance between the results of the two types of descriptors compared, a Wilcoxon rank-sum test was performed on the results corresponding to each of the experiments described above. The results of these tests are reported in Table 1. As can be seen, there is not enough evidence to reject the null hypothesis H_0 (that the results of the different descriptors were sampled from distribution with equal medians) in any of the cases tested. The conclusion is that differences between the descriptors are not statistically significant.

5 Conclusions and Future Work

A novel image descriptor based on Gegenbauer moment invariants was described and a comparative evaluation between this and the very popular SURF descriptor was conducted in the context of visual scene recognition. Different experiments were performed on image pairs obtained from uncontrolled environments and presenting challenging conditions. Although the SURF descriptors consistently achieved

a higher recognition rate within a small range of the recognition-threshold parameter, statistically significant differences between the complete set of results from each of the experiments were not observed in any case. The conclusion is that the performance of the proposed Gegenbauer-based descriptors is similar to that of state-of-the-art descriptors such as SURF. This is a positive result, but more work needs to be carried out.

As future work, a more extensive comparative evaluation between the *GID* and other successful image descriptors will be conducted. More robust invariants will be added and the effect of other parameters such as the scale factor α , which controls the shape of the Gegenbauer polynomials, will be investigated.

Acknowledgements This work was partially supported by the National Council of Science and Technology (CONACYT) of Mexico, through Grant numbers: 416924 (A. Herrera) and CATEDRAS-2598 (A. Rojas).

References

1. Chen, C.H.: Handbook of Pattern Recognition and Computer Vision. World Scientific (2016)
2. Davies, E.R.: Machine Vision: Theory, Algorithms, Practicalities. Morgan Kaufmann (2005)
3. Mur-Artal, R., Montiel, J.M.M., Tardos, J.D.: ORB-SLAM: a versatile and accurate monocular slam system. *IEEE Trans. Rob.* 1147–1163 (2015)
4. Krig, S.: Computer Vision Metrics. Springer (2016)
5. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**(2), 91–110 (2004)
6. Tola, E., Lepetit, V., Fua, P.: DAISY: an efficient dense descriptor applied to wide-baseline stereo. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(5), 815–830 (2010)
7. Bai, Y., Guo, L., Jin, L., Huang, Q.: A novel feature extraction method using pyramid histogram of orientation gradients for smile recognition. In: 16th IEEE International Conference on Image Processing (ICIP) (2009)
8. Hosny, K.M.: New set of Gegenbauer moment invariants for pattern recognition applications. *Arab. J. Sci. Eng.* **39**(10), 7097–7107 (2014)
9. Zhu, H., Shu, H., Zhou, J., Lou, L., Coatrieux, J.: Image analysis by discrete orthogonal dual Hahn moments. *Pattern Recogn. Lett.* **28**(13), 1688–1704 (2007)
10. Chiang, A., Liao, S., Lu, Q., Pawlak, M.: Gegenbauer moment-based applications for chinese character recognition. In: 2002 IEEE Canadian Conference on Electrical & Computer Engineering (2002)
11. Pawlak, M.: Image Analysis by Moments: Reconstruction and Computational Aspects. Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław (2006)
12. Fulsser, J., Suk, T., Zitová, B.: Moments and Moment Invariants in Pattern Recognition. Wiley (2009)
13. Sonka, M., Hlavac, V., Boyle, R.: Image Processing, Analysis, and Machine Vision. Thomson (2008)
14. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Prentice Hall (2002)
15. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **110**(3), 346–359 (2008)

Bimodal Biometrics Using EEG-Voice Fusion at Score Level Based on Hidden Markov Models



Juan Carlos Moreno-Rodriguez, Juan Manuel Ramirez-Cortes,
Rene Arechiga-Martinez, Pilar Gomez-Gil and Juan Carlos Atenco-Vazquez

Abstract This paper presents an experiment on bimodal biometrics based on fusion at score level of electroencephalographic (EEG) and voice signals. The experiments described have been carried out using an open database with commands in Spanish, available to the academic community for research purposes. The accuracy of user identification systems is evaluated using Hidden Markov Models (HMM) as classifiers for both EEG and voice modalities. Feature extraction is implemented using Mel-Frequency Cepstrum Coefficients (MFCC) and Wavelet analysis for voice and EEG, respectively. From the scores generated by each independent system, data fusion is proposed at a score level for multiple cases of weighted sums based on weighted arithmetic means, and for the score product case using a geometric mean scheme. Performance evaluation indicates a recognition rate of 90% in average. The results obtained confirm the expected tendency, validating the convenience and usefulness of multimodal user identification systems while setting the ground for future studies involving data fusion at different levels and using other classifiers.

J. C. Moreno-Rodriguez · J. M. Ramirez-Cortes (✉) · J. C. Atenco-Vazquez
Instituto Nacional de Astrofisica, Optica y Electronica, Electronics Department,
Cholula, Mexico
e-mail: jmram@inaoep.mx

J. C. Moreno-Rodriguez
e-mail: xalatl@inaoep.mx

J. C. Atenco-Vazquez
e-mail: atencovaz@inaoep.mx

R. Arechiga-Martinez
Department of Electrical Engineering, New Mexico Tech, Socorro, USA
e-mail: rene.arechiga@nmt.edu

P. Gomez-Gil
Instituto Nacional de Astrofisica, Optica y Electronica, Computer Science Department,
Cholula, Mexico
e-mail: pgomez@inaoep.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_44

1 Introduction

As defined by Akhtar [1], biometrics is a technique employed to recognize or identify individuals based on the measurement and analysis of biological and behavioral characteristics. A biometric system uses a biosignal, a personal trait or a combination of them, to automatically identify a subject (i.e. identification) or to verify a proposed identity (i.e. verification). Biosignals can be categorized into two groups: bioelectrical, which reflects the electrical activities of tissues such as nerves or muscles and non-bioelectrical, produced by body activities manifesting in a non-electrical manner. Acoustic signals, such as phonocardiogram and optical signals, such as photoplethysmogram are typical examples of the latter category and even though they are not electrical in origin, they are usually converted into electrical signals to be analyzed and processed by automated system [2]. A graphical representation of this classification is shown in Fig. 1. In order to qualify to be a biometric, a biological measurement must comply with the requirements of *universality, distinctiveness, permanence, collectability, performance, acceptability and circumvention* [3].

Figure 2 shows a typical biometric identification system. At stage one, the selected biological characteristic is presented to a data acquisition block. The information produced by this block is usually segmented in order to satisfy certain conditions required by the next stage; these two blocks can be referred as a single step known as

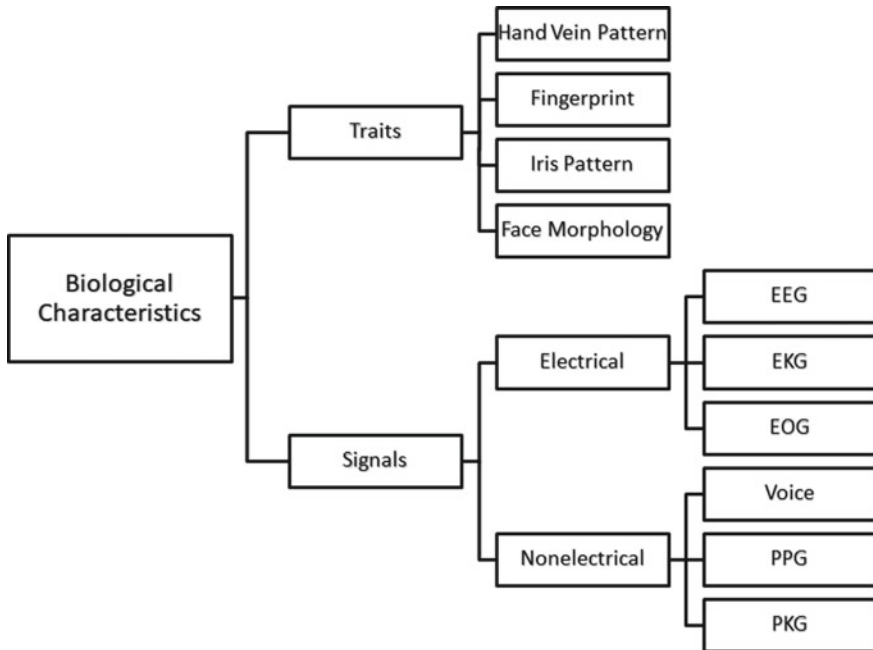


Fig. 1 Biological characteristics classification for biometrics use. The depicted examples are some of the most representative cases among several proposed modalities

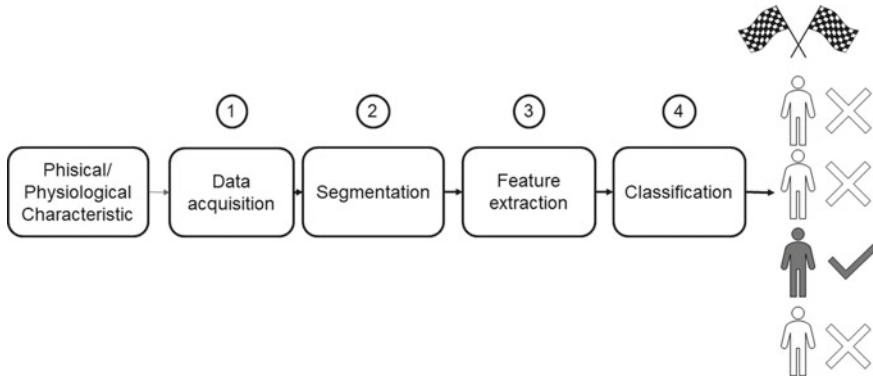


Fig. 2 A typical biometric identification system

the pre-processing stage. The feature extraction block, depicted as stage three, plays a critical role in the identification process. Its goal is to separate the useful information required to discriminate subjects out of the raw signals from the previous stage. Hence, the proper selection of significant features at the design stage of the system is crucial for a trustable and efficient result. These features represent the input to the classification block, which goal is to map the presented information onto a label corresponding to a class, in the case of an identification system, the preregistered users [4].

In [5] Akhtar establishes that, due to factors such as noise in sensed data, intra-class variations, inter-class similarities, non-universality, interoperability issues and spoof attacks, unimodal biometric systems (that is, biometric systems using a single trait or signal to perform identification or authentication) “are not sufficient to meet the variety of requirements, including matching performance, imposed by several large-scale authentication systems”. In order to overcome unimodal limitations, the use of multimodal systems is proposed. A recent review [6] reports the existence of the following multi-modal studies: EEG-EOG, EKG-EMG, EKG-PCG, EKG-face-fingerprint, PCG-EKG, EEG-face, face-palmprint, face-iris, face-ear, face-lips, among others.

In a biometric system, modalities fusion can be accomplished at five different levels: sensor-level, feature-level, score-level, rank-level and decision level fusion. According to Tistarelli [7] score-level fusion “is the most commonly discussed approach in the biometric literature primarily due to the ease of accessing and processing match scores”.

When proposing a multi-modal system, an important criterion to be considered is that a modality weakness should be compensated by the other modality strength, in accordance with the previously mentioned compliance requirements. EEG signals have emerged as a potential biometric signal according to their condition of presenting moderate to high performance, however, collectability of the signals still needs the use of specialized devices and a level of cumbersome which might be impractical in real life scenarios [8, 9]. This situation is rapidly evolving with recent technological

advances oriented to development of simple, low-cost, yet efficient sensor devices, many of them arising in the context of Internet of Things (IoT). Another interesting condition associated to EEG signals is their robustness to circumvention given the fact that spoofing is out of the people control, and liveness detection is implicit when EEG signals are sensed. Among some relevant techniques reported recently in the literature with good results we find an EEG based login system by Chen et al. [10], biometric systems based on hand synergies and their neural representations by Patel et al. [11] or a multimodal approach based on signature and EEG signals, developed by Saini et al. [12].

On the other hand, voice-based biometric is an attractive modality in our current life immersed in a plethora of communication systems, telephone services, internet, social networks, gaming, etc. Nowadays, there are many techniques for speaker verification which provide an excellent performance, based on voice password [13], text dependent [14] and text independent [15] speaker verification, in addition to multilevel person authentication systems, like the one reported by Das et al. [16], as well commercial applications worldwide available such as Nuance Voice Biometrics Solution [17] or LumenVox Multifactor Authentication [18]. Nevertheless, spoofing attacks using fraudulent signals relying on synthetic speech, digital recordings, digital voice conversion, or even imitators, are still a threat to voice-based biometric. Is in this context that bimodal or multimodal biometric systems are receiving attention in search of robust and convenient complementary techniques. After reviewing the advantages and disadvantages of some of the most frequently used modalities [6] attention is brought to the use of voice and EEG signals.

2 Proposed Method

This article proposes a bimodal EEG-voice identification system with a score-level data fusion scheme, as shown in Fig. 3. Fusion is calculated by means of weighted

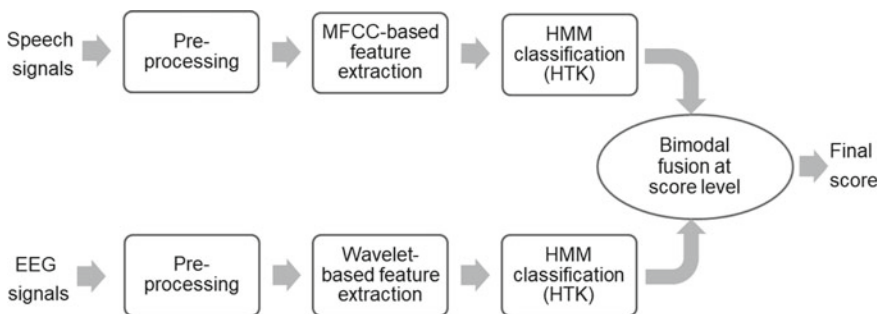


Fig. 3 The proposed scheme for a bimodal EEG-Voice biometrics system with a score-level fusion stage

sums and scores product. The system is based on four main steps: (a) a voice recognition score generator using HMM's, (b) an EEG recognition score generator, (c) a data processing and fusion stage and (d) a decision stage as output. Each stage will be detailed in further sections.

To test the scheme the database generated by Coretto et al., available to the research community for academic purposes [19], is employed. This database is made up of voice and EEG signals recorded during imagined speech, from fifteen Argentinian subjects, eight male, seven female, with a mean age of 25 years old, one left-handed, the rest right-handed. The database was originally intended to be used on BCI research so it consists of a set of commands, nevertheless, the audio and EEG samples can be used for biometric purposes. Two sets of commands were defined to be imagined/pronounced by the subjects. The first set was formed by the five vowels, Spanish pronounced and represented as /a/, /e/, /i/, /o/and /u/. The second set included six common orders for an electrical wheel chair or displacement device: *adelante*, *atrás*, *arriba*, *abajo*, *derecha*, *izquierda* (forward, backward, up, down, right, left, respectively) During recording sessions, subjects were seated in a chair in front of an LCD display at a distance of approximately one meter, were target words where shown to be recorded under two conditions: during imagined speech and pronounced speech. For the purposes of the experiments described in this article only the pronounced samples were used. In accordance with the 10-20 EEG system for electrode placement, the available signals correspond to channels F3, F4, C3, C4, P3 and P4. Further details on the experimental protocol and data collection can be consulted in [19].

The procedure was applied separately for each command, in order to compare the difference in results according to the complexity of the pronounced vocal or word. Therefore, 11 different sets of samples from 15 users were generated from the original dataset to perform the evaluation. The original intention was to use 10 samples per subject for all 11 cases, however, for some users and some commands, the available samples of pronounced signals (that is, the EEG signal corresponding to a particular voice signal) were insufficient. Table 1 shows the number of samples used for test and the total samples used for each command; in all cases the number of samples used for training was 105.

Table 1 Samples used by command

Command	Test	Total	Command	Test	Total
/a/	45	150	<i>Arriba</i>	45	150
/e/	45	150	<i>Abajo</i>	43	148
/i/	44	149	<i>Adelante</i>	42	147
/o/	44	149	<i>Atrás</i>	43	148
/u/	45	150	<i>Derecho</i>	42	147
			<i>Izquierda</i>	44	149

(A) *Voice Recognition*

Voice signal have been widely used for biometric purposes. One of the most used procedure for feature extraction is the calculation of Mel-Frequency Cepstrum Coefficients (MFCC) [20] to form a feature vector to be fed as input for a selected classifier. The common procedure to compute the MFCC's for a speaker recognition system consists of the following six steps: (1) framing of the original signal to be treated as a stationary signal, in our case the signal is framed applying a 30 ms Hamming window with an overlap of 15 ms; (2) calculation of the periodogram-based power spectral estimate for the each frame; (3) application of the Mel filter bank to the power spectra and calculation of the sum in each filter, a typical value of 26 filter was used; (4) computation of the logarithm of the sums obtained in the previous step; (5) evaluation of the Discrete Cosine Transform of the values calculated in step 4; (6) forming of the feature vector with coefficients 2–13, discarding the rest. Further information on MFCCs can be consulted in several sources, e.g. the work of Agrawal et al. [21].

A typical speaker recognition system was generated using the Hidden Markov Model Toolkit (HTK) [22]. Firstly, a finite state grammar was constructed, defining a state for each user, namely *uno, dos, tres... quince*. A word network was created based on the proposed grammar, using HTK's HParse. Next, the audio files from the database were parametrized to obtain the Mel-frequency cepstral coefficients (MFCCs). The first seven MFCC files for each user were used to initialize the 15 users' models using HTK's HCompV command, each one with one input state, three hidden states and an output state. Figure 4 shows a typical representation of a Model like the mentioned before; as it can be seen by the representation, a HMM can be perceived as a finite state machine defined by (1) a set of states $\Omega = \{1,2,3,\dots, N\}$; (2) a set of possible observations $\mathbf{O} = \{o_1, o_2, o_3,\dots, o_M\}$; (3) a transition probability matrix $\mathbf{A} = [a_{ij}]$ and (4) an output probability distribution $\mathbf{B} = \{b_i(k)\}$. The stochastic nature of HMM's is manifested in the transition of states governed by the transition matrix \mathbf{A} as it is in the observation's generation given by \mathbf{B} , been both probabilistic events. More illustrative information on Hidden Markov Models can be found in [23]. Model re-estimation was performed via three iterations using HERest.

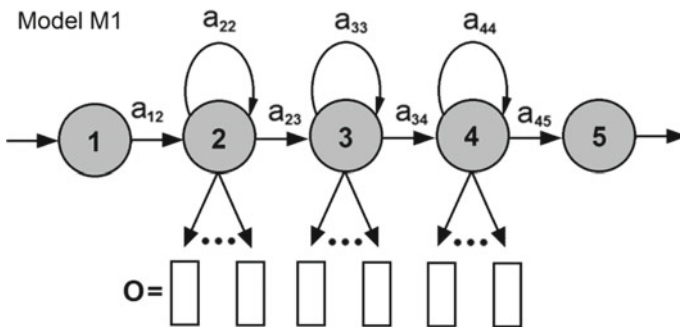


Fig. 4 Hidden Markov Model representation: three hidden states

(B) *EEG Recognition*

EEG signals represent one of the most promising biometrics. The fact that a matching signal is hardly obtained by coercion or insistence, liveness is a required condition that prevents the user to be murdered to obtain access to the required biometric, it is an invisible signal for the naked eye which makes it uncapturable by third parties, its uniqueness, its complexity to be forged and some other advantages [24, 25], makes them a robust modality in terms of security for the user and the asset or installation related with the biometry system. Beside these advantages, Abo-Zahhad et al. in [26] review and confirm compliance with universality, distinctiveness, permanence, collectability, performance, acceptance and circumvention.

In terms of practical implementation, three channels were proposed to be used. Considering that (a) localization of speech has been established to be a left hemispheric function [27] and (b) the available channels in the working database are F3, F4, C3, C4, P3 and P4, the left-side available channels (namely, F3, C3 and P3) were selected. The information contained in the three selected channels was synthesized in a single signal, via a spatial semi-Laplacian given by the function (1)

$$R = 2C3 - (F3 + P3) \quad (1)$$

The resulting signal was then used to perform EEG user identification following the same procedure as in the previous section. The substantial difference lies on the feature extraction stage; while for the voice signal MFCCs were used, for EEG a feature vector must be defined. In a similar way as in [28], the synthesized signal was decomposed using a five level third order Daubechies decomposition wavelet transform to obtain five sets of detail coefficients (D1–D5) and one set of approximation coefficients, corresponding to specific sub-bands of the EEG spectra. Figure 5 shows an example of the described decomposition using Matlab.

These coefficients were used to calculate the following statistical features: (V1) Mean of the absolute values of the coefficients in each sub-band, (V2) the average power of the wavelet coefficients of each sub-band, (V3) the standard deviation of the coefficients in each band and (V4) the ratio of the absolute mean values of adjacent sub-bands. This defines a 24-dimension feature vector: four V values for each one of the six bands contemplated. Once the feature vector is defined, the HMMs scores are generated, as mentioned before, following the same procedure as for the speaker recognition section. For the EEG signals only one hidden state was used, instead of the three employed for voice signals.

(C) *Data Processing and Fusion*

An output score-level data fusion scheme was chosen to test the expected accuracy improvement of a bimodal system when compared to a unimodal one. The fundamental idea to choose this level arises from the observation of the mismatched samples' scores of both unimodal systems. Through that analysis, evidence becomes visible pointing to the fact that in several of the wrongfully classified samples, the correct choice was the second or the third best rated score. So, it would be reasonable to

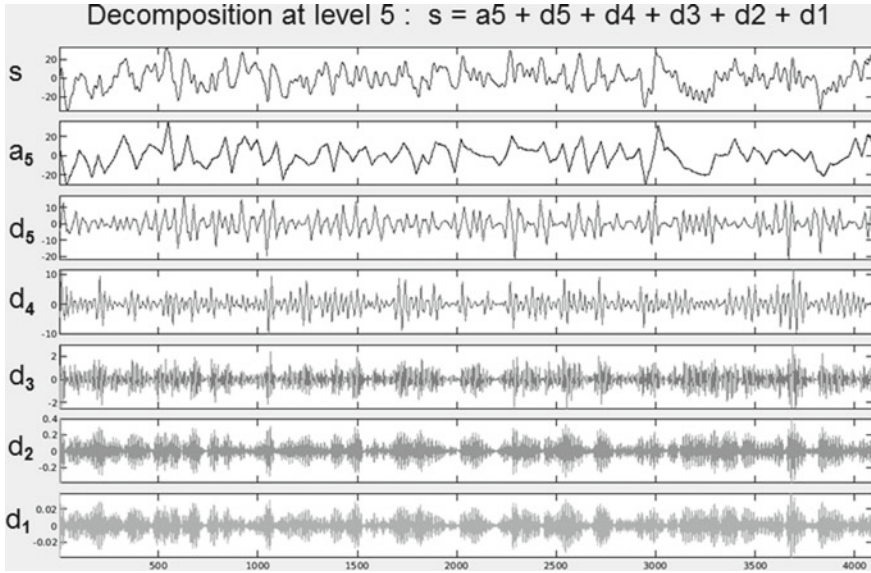


Fig. 5 Level 5 wavelet decomposition for an EEG signal using a third order Daubechies wavelet

suppose that a small amount of additional information would be enough to contribute in the correction of the subject identification. The proposed way in which the fusion is generated is by a direct weighted sum of the scores from both EEG and voice samples. However, due to the difference in the nature of the signals and the proposed number of states for the Hidden Markov Models in each modality, it is expected to obtain scores differing in magnitude order for voice an EEG; this difference would produce an undesirable weighting effect in favor of the higher order modality.

According to [22], the computation of likelihoods performed by HVite’s Viterbi algorithm uses log likelihoods to avoid possible underflow expected when performed by direct calculation. Then, as these log scores are calculated by (2) the results span from negative to positive values.

$$\psi_j(t) = \max_i \{ \psi_i(t - 1) + \log a_{ij} \} + \log(b_j(o_t)) \tag{2}$$

To eliminate the effect of mismatched magnitude order, a magnitude normalization must be executed prior to the weighted addition. The procedure intended to normalize scores into the [0, 1] range consists of two steps: (1) shifting all scores to the natural numbers domain by adding the absolute value of the minimum score to all the 15 available scores and (2) dividing the resulting scores by the maximum of the set. Once done, both EEG and voice sets will be contained between the defined limits and will be available to be used in the selected weighted sum. Depending on the selection of the weighting coefficients, the produced synthetic signal will have a predominant EEG component, a balanced mix or a predominant voice component.

D. Decision Stage

The classification of the resulting sum preserves the classification criteria used by HTK’s HResults: the sample will be tagged as belonging to the model with the higher logarithmic probability. In this case, as scores are already normalized prior to the addition process, the sample will be identified as belonging to the subject’s model with a score closer to one.

3 Experimental Results

According to the procedure detailed in the previous section, recognition rates were calculated for each of the eleven available commands. Beside the single modality rates and their product, nine different weighted sums for are considered for each command, computing the expression (3).

$$WS(a) = aEEG + (1 - a)Voice$$

for $a = \{1, 0.9, 0.8 \dots 0.2, 0.1, 0\}$ (3)

Figure 6 shows the performance evolution of the recognition rate as the proportion of the mix is modified for the case of the /i/ command. Notice that this case is presented given that both pure signal rates (EEG and voice) are the lowest ones registered, making it hence the worst evaluated case. However, even with the minimal 10% voice contribution, the recognition rate increases more than one hundred percent when compared to pure EEG, going from 36 to 73%. Moreover, for 60%/40% mix

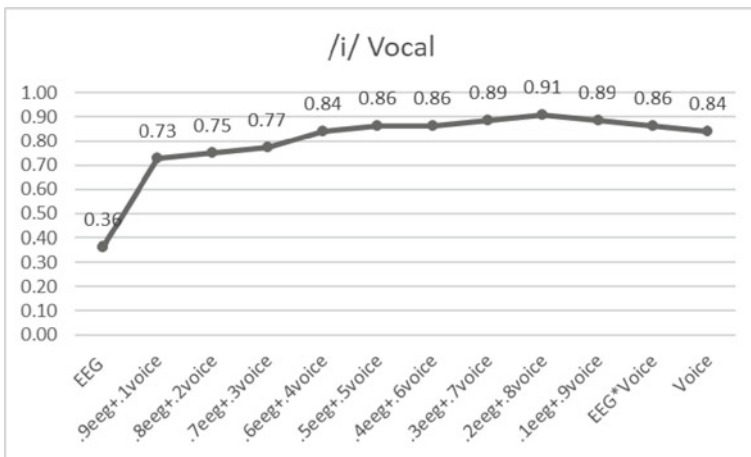


Fig. 6 Accuracies obtained for the proposed mixes, /i/ vocal case. Notice that the first value corresponds to the EEG modality, EEG * Voice to the geometric mean and right-most to the Voice modality

proportion, results equals the performance of the best rated single modality, in this case, voice, accomplishing the best performance (91%) with a 20%/80% proportion.

As for a directional command, Fig. 7 shows the accuracies obtained for the /Adelante/ command. Attention is brought to the following facts: (a) again, even with a minimal contribution of 0.1, the voice component boosts the accuracy from 55% for pure EEG to 81% for the 0.9EEG to 0.1 Voice mix; (b) the best-performance single modality accuracy is surpassed by at least one bimodal case, in this case almost all of them; c) the score product equals the best rated weighted sums.

As it can be seen in Figs. 8 and 9, consistent results are observed for the rest of the cases. Figure 8 shows a summary for the vocal commands, while Fig. 9 depicts the tendency for the directional commands.

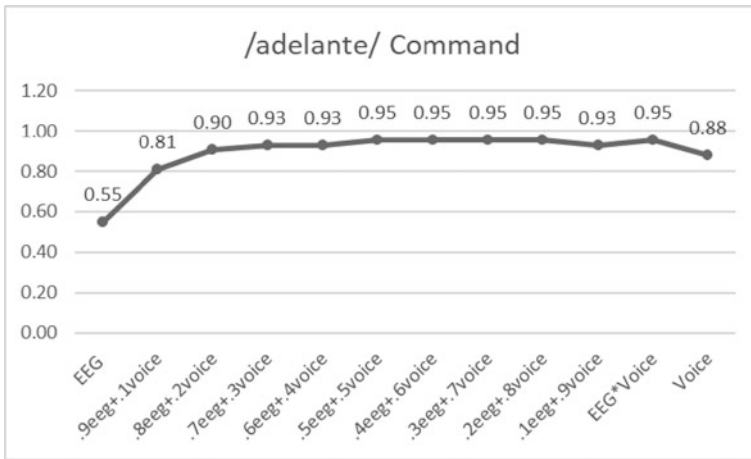


Fig. 7 Accuracies obtained for the proposed mixes, /adelante/ case

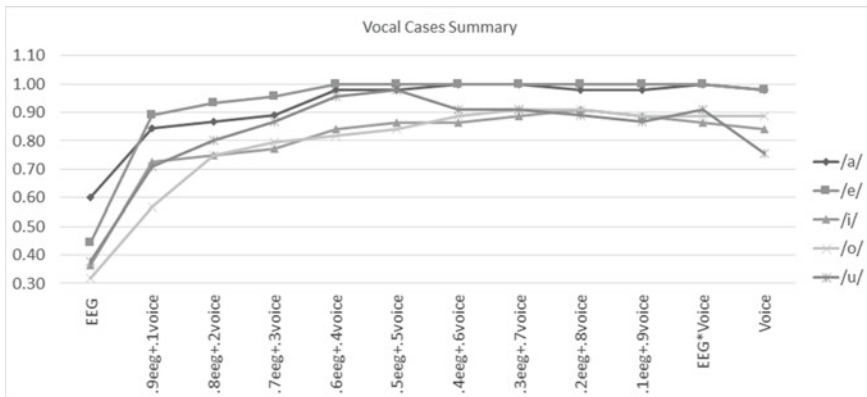


Fig. 8 Results summary: vocal commands

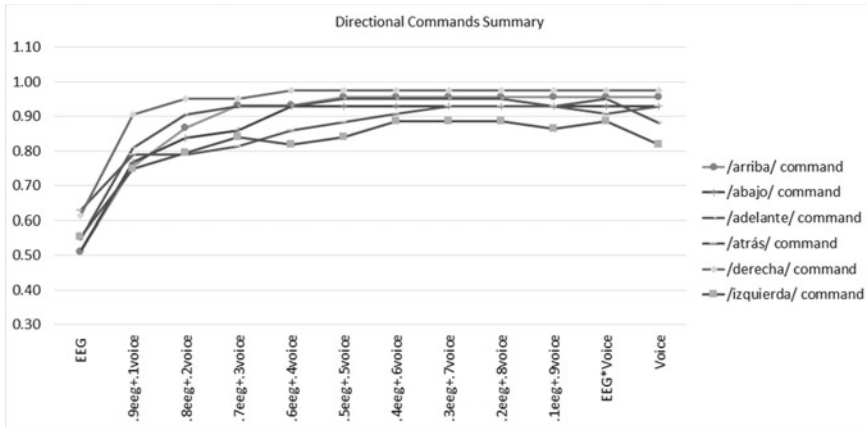


Fig. 9 Results summary: directional commands

4 Conclusions

The proposed method represents a simple yet effective solution to the aforementioned problem of increasing the recognition rates of an EEG biometric system. The inclusion of the user’s voice signal boosts the performance of recognizer in comparison to the pure EEG signal. Besides, given the fact that no additional signal conditioning was performed over the original database signals, it would be expected to obtain even better performance after the deputation of the signal’s quality, e.g. blinking artifact removal by ICA for EEG and truncated audio edition for voice. However, the preliminary results generated by this bimodal scheme are promising.

Acknowledgements J.C. Moreno-Rodriguez and J.C. Atenco-Vazquez acknowledge the financial support from the Mexican National Council for Science and Technology (CONACYT) to pursue doctoral studies.

Glossary

As it was previously detailed HTK software [22] was employed to generate de HMMs and to evaluate the classifiers performances. This section presents a brief description of the specific tools used in the proposed experiments, to provide the uninitiated reader a general perspective of the toolkit operation.

HCompV Program to calculate the global mean and covariance of a set of data, in order to be used as a starting point or initialization of all models.

HCOPY The function of this program is to copy a single file or a set of them to a parametrized file format which HTK can recognize. A useful function of this

program is the conversion of waveform file to a MFCC file by the calculation of the cepstral coefficients.

HERest The function of this program is to perform a single re-estimation of the parameters of a set of HMMs using a version of the Baum-Welch algorithm. After initialized, the models are presented with labeled training files to produce the particular re-estimations.

HParse Generates lattice files from a syntax description in a text file. It generates the internal representation of the finite-state network.

HResults Analysis tool employed for an easy visualization and interpretation of the generated results.

HTK Portable toolkit for building and manipulating hidden Markov models.

HVite General-purpose Viterbi recognizer. It matches an input file against a network of HMMs and output a transcription for each. When performing N-best recognition, multiple hypotheses can be produced.

References

1. Akhtar, Z., Hadid, A., Nixon, M.S., Tistarelli, M., Dugelay, J.L., Marcel, S.: Biometrics: in search of identity and security (Q A). *IEEE Multimed.* **25**(3), 22–35 (2018)
2. Supratak, A., Wu, C., Dong, H., Sun, K., Guo, Y.: Survey on feature extraction and applications of biosignals. *Mach. Learn. Heal. Informatics* **9605**, 161–182 (2016)
3. Jain, A.K., Ross, A., Prabhakar, S.: An introduction to biometric recognition. *IEEE Trans. Circuits Syst. Video Technol.* **14**(1), 4–20 (2004)
4. Carvalho, S.N., et al.: Comparative analysis of strategies for feature extraction and classification in SSVEP BCIs. *Biomed. Signal Process. Control* **21**, 34–42 (2015)
5. Akhtar, Z., Kale, S.: Security analysis of multimodal biometric systems against spoof attacks. *Commun. Comput. Inf. Sci.* **191**(Part 2), 604–611 (2011)
6. Zapata, J.C., Duque, C.M., Rojas-Idarraga, Y., Gonzalez, M.E., Guzmán, J.A., Becerra Botero, M.A.: Data fusion applied to biometric identification—a review. In: *Communications in Computer and Information Science*, vol. 735, pp. 721–733 (2017)
7. Tistarelli, M. (ed.): *Handbook of Remote Biometrics*, vol. 6256. Springer, London (2010)
8. Khalifa, W., Salem, A., Roushdy, M.: A survey of EEG based user authentication schemes. In: *2012 8th International Conference on Informatics and Systems (INFOS)*, pp. 55–60 (2012)
9. Goverdovsky, V., Looney, D., Kidmose, P., Mandic, D.P.: In-ear EEG from viscoelastic generic earpieces: robust and unobtrusive 24/7 monitoring. *IEEE Sens. J.* **16**(1), 271–277 (2016)
10. Chen, Y., et al.: A high-security EEG-based login system with RSVP stimuli and dry electrodes. *IEEE Trans. Inf. Forensics Secur.* **11**(12), 2635–2647 (2016)
11. Patel, V., Burns, M., Chandramouli, R., Vinjamuri, R.: Biometrics Based on Hand Synergies and their Neural Representations. *IEEE Access* (2017)
12. Saini, R., et al.: Don't just sign use brain too: a novel multimodal approach for user identification and verification. *Inf. Sci. (Ny)* **430–431**, 163–178 (2018)
13. Latvala, O.-M., Peng, C., Honkamaa, P., Halunen, K.: “Speak, Friend, and Enter”—secure, spoken one-time password authentication. In: *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5 (2018)
14. Heigold, G., Moreno, I., Bengio, S., Shazeer, N.: End-to-end text-dependent speaker verification. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 5115–5119 (2016 May)

15. Zhang, C., Koishida, K.: End-to-end text-independent speaker verification with triplet loss on short utterances. In: Proceedings of Annual Conference of the International Speech Communication Association (INTERSPEECH 2017), pp. 1487–1491 (2017 August)
16. Das, R.K., Jelil, S., Mahadeva Prasanna, S.R.: Development of multi-level speech based person authentication system. *J. Signal Process. Syst.* **88**(3), 259–271 (2017)
17. Beranek, B., Warren, R.: Authenticating Customers with Nuance Voice Biometrics Solutions, Redbooks REDP-4903-00, IBM Corporation, 1–18 (2012)
18. LumenVox.: Multifactor Authentication—LumenVox. [Online]. Available <https://www.lumenvox.com/multifactor-authentication/>. Accessed 19 Feb 2019
19. Pressel Coretto, G.A., Gareis, I.E., Rufiner, H.L.: Open access database of EEG signals recorded during imagined speech. In: 12th International Symposium on Medical Information Processing and Analysis, vol. 10160, p. 1016002 (2017)
20. Muda, L., Begam, M., Elamvazuthi, I.: Voice recognition algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) techniques. *J. Comput.* **2**(3), 138–143 (2010)
21. Agrawal, S., Mishra, D.K.: Speaker verification using mel-frequency cepstrum coefficient and linear prediction coding. In: 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE), pp. 543–548. IEEE (2018 January)
22. Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Valtchev, V.: The HTK book, n.p.: Cambridge University Engineering Department, **3**(1), 1–175 (2002)
23. Visser, I.: Seven things to remember about hidden Markov models: a tutorial on Markovian models for time series. *J. Math. Psychol.* **55**(6), 403–415 (2011)
24. Kaur, B., Singh, D.: Neuro signals: a future biomertic approach towards user identification. In: Proceedings of the 7th International Conference Confluence 2017 on Cloud Computing, Data Science and Engineering, pp. 112–117 (2017)
25. Jayarathne, I., Cohen, M., Amarakeerthi, S.: Survey of EEG-based biometric authentication. In: Proceedings of 2017 IEEE 8th International Conference on Awareness Science Technology (iCAST 2017), pp. 324–329 (2018 January)
26. Abbas, S.N., Abo-Zahhad, M., Ahmed, S.M.: State-of-the-art methods and future perspectives for personal recognition based on electroencephalogram signals. *IET Biometrics* **4**(3), 179–190 (2015)
27. Ross, E.D.: Cerebral localization of functions and the neurology of language: fact versus fiction or is it something else? *Neuroscientist* **16**(3), 222–243 (2010)
28. Subasi, A.: EEG signal classification using wavelet feature extraction and a mixture of expert model. *Expert Syst. Appl.* **32**(4), 1084–1093 (2007)

Towards a Quantitative Identification of Mobile Social Media UIDPs' Visual Features Using a Combination of Digital Image Processing and Machine Learning Techniques



Viviana Yarel Rosales-Morales, Nicandro Cruz-Ramírez,
Laura Nely Sánchez-Morales, Giner Alor-Hernández,
Marcela Quiroz-Castellanos and Efrén Mezura-Montes

Abstract User Interface Design Patterns (UIDPs) improve the interaction between users and e-applications through the use of interfaces with a suitable and intuitive navigability without restrictions on the size of the screen to show the content. Nowadays, UIDPs are frequently used in the development of new mobile apps. In fact, mobile apps are ubiquitous: in education through learning platforms; in medicine through health self-care apps and in a social dimension, of course, through social networks. Social media networks have become one of the main channels of communication and dissemination of content; however, surprisingly, UIDPs have not been deeply analyzed in the design and development process of social media apps. In this sense, we propose the use of a combination of digital image processing and machine learning techniques to both quantitatively identify the main visual features of UIDPs in social media apps and assess the goodness of those features for building

V. Y. Rosales-Morales (✉) · N. Cruz-Ramírez · M. Quiroz-Castellanos · E. Mezura-Montes
Centro de Investigación en Inteligencia Artificial, Universidad Veracruzana, Sebastián Camacho
no. 5 Col. Centro, Xalapa, Veracruz 91000, Mexico
e-mail: viviana_rosales@outlook.com

N. Cruz-Ramírez
e-mail: ncruz@uv.mx

M. Quiroz-Castellanos
e-mail: qc.marcela@gmail.com

E. Mezura-Montes
e-mail: emezura@gmail.com

L. N. Sánchez-Morales · G. Alor-Hernández
Division of Research and Postgraduate Studies, Tecnológico Nacional de México/I. T. Orizaba,
Av. Oriente 9 no. 852, Col. Emiliano Zapata, 94320 Orizaba, Veracruz, Mexico
e-mail: lauransanchezmorales@gmail.com

G. Alor-Hernández
e-mail: galor@itorizaba.edu.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_45

highly accurate classifiers. Our results suggest that such a combination seems sensible not only for explicitly unveiling patterns shared by different users but also for constructing such kind of classifiers.

Keywords Mobile applications · Social media apps · User interface design patterns (UIDPs) · Digital image processing · Machine learning · Decision trees

1 Introduction

Although researchers have different definitions of social media sites, all definitions reveal the same meaningful function [1–3]: social media sites are web-based sites for social communication where Internet users can create online communities to share information with one another. On the other hand, software engineering helps establish mechanisms for software development through models, methodologies or patterns [4]. UIDPs are recurring solutions that solve common design problems. The UIDPs are standard benchmarks for the experienced UI designer [5]; i.e., UIDPs are solutions to build man-machine interfaces through a graphical interface. Due to the above, different areas of software engineering have used UIDPs to solve common design problems; one of them being software development process. The UIDPs are important because they improve human-computer interaction by using intuitive interfaces, with a suitable navigability and without restrictions on the size of the screen to adequately show the content. However, there is a lack of an in-depth analysis to identify the most appropriate UIDPs for the development of social media apps. Thus, in this paper, we present some steps towards such an analysis that combine techniques from digital image processing and machine learning which allow us not only to identify and extract important quantitative visual features of UIDPs in social media apps but also to build a highly accurate classifier based on those characteristics. We argue that this combination may provide a guide to encourage the use of best practices during the design phase. To that end, we first review a set of well-known social media apps in order to identify the UIDPs that are most frequently used. Then, we carry out a digital image processing of some of the most common interfaces that are present in those apps. Finally, we build a classifier, based on decision trees, to assess whether the features extracted in the previous step can faithfully and compactly represent these interfaces.

The rest of the paper is organized as follows: Section 2 reviews the state-of-the-art about UIDPs in mobile apps and development and assessment of mobile apps using digital image processing and machine learning techniques. In Sect. 3, the classification of mobile social media apps is presented; Sect. 4 reports the set of UIDPs identified in mobile social media apps while Sect. 5 presents the materials and methods for the use of decision trees to identify and classify UIDPs. Finally, in Sect. 6 we formulate concluding remarks and provide recommendations for future work.

2 State-of-the-Art

The social media apps are very broad and diverse and their use has increased considerably in recent years. This section is then divided into two parts, a) UIDPs in mobile apps and b) Development and assessment of mobile apps using digital image processing and machine learning techniques.

(A) UIDPs in Mobile Apps

User Interface Design Patterns have been used to solve design problems in multiple domains since they facilitate the development of mobile applications [5]. In this sense, Cortes-Camarillo et al. [6] carried out a comparative analysis of UIDPs to generate contexts of use oriented to the educational environment. They designed contexts of use to establish which UI design patterns are recommended for a specific device by facilitating the development of multi-device educational applications. Videla-Rodríguez et al. [7] presented a research work to determine the elements, components and key factors for the design of interactive user interfaces for augmented reality environments. Particularly, they focused on the development of virtual environments for educational applications. Quispe Rodríguez [8] carried out an assessment about users with problems in the recognition of colors, known as color-blind users. From this perspective, he proposed to use interfaces with custom structures according to the needs of the users mentioned, without harming usability of the website. Figeac and Chaulet [9] presented a neural machine translator that combines recent advances in computer vision and machine translation for translating a UI design image into a GUI skeleton. The translator learns to extract visual features in UI images, encode these features' spatial layouts, and generate GUI skeletons in a unified neural network framework, without requiring manual rule development. For training the translator, the authors developed an automated GUI exploration method to automatically collect large-scale UI data from real-world applications. Sánchez-Morales et al. [10] presented a software component to generate user interfaces for mobile applications using pattern recognition techniques, image processing and neural networks. The process of generating user interfaces consists of three phases: (1) image analysis, (2) configuration and (3) generation of source code. The component identifies the elements in a freehand generated image to transform each element to its equivalent source code. Cortes-Camarillo et al. [11] presented Atila, a generator of educational applications based on UIDPs. The applications developed are compatible with 4 operating systems: Android™, Firefox® OS, iOS™, Windows® Phone and Web. The ability to quickly generate applications is their main advantage. Markkula and Mazhelis [12] presented a generic architectural model approach to organize patterns. In the approach proposed, the identification of relevant patterns is considered as the process of reducing the set of candidate patterns by implicit restrictions in the mobile domain. These restrictions can be incorporated into a domain-specific generic architectural model that reflects the common characteristics in the particular domain solutions.

(B) **Development and Assessment of Mobile Apps Using Digital Image Processing and Machine Learning Techniques**

Machine learning and digital image processing are artificial intelligence techniques that have been used to improve the performance and characteristics of mobile applications, even in their development phase. In this sense, Moran et al. [13] presented an approach that automates this process by enabling accurate prototyping of GUIs via three tasks: detection, classification, and assembly. (1) Logical components of a GUI are detected from a mock-up artifact using either computer vision techniques or mock-up metadata. (2) The software repository mining, automated dynamic analysis, and deep convolutional neural networks are utilized to accurately classify GUI-components into domain-specific types. (3) A data-driven, K-nearest-neighbors algorithm generates a suitable hierarchical GUI structure from which a prototype application can be automatically assembled. They implemented this approach for Android in a system called REDRAW, which achieves an average GUI-component classification accuracy of 91% and assembles prototype applications that closely mirror target mock-ups in terms of visual affinity while exhibiting reasonable code structure. Umair et al. [14] explored the potential of machine learning in mobile forensics, and specifically in the context of Facebook messenger artifact acquisition and analysis. They demonstrated how one can acquire Facebook messenger app artifacts from mobile devices. Based on the acquired evidence, they created 199 data-instances to train WEKA classifiers (i.e. ZeroR, J48 and Random tree) with the aim of classifying the device owner's contacts and determine their mutual relationship strength. Basavaraju and Varde [15] gave a comprehensive review of a few useful supervised learning approaches along with their implementation in mobile apps, focusing on Androids as they constitute over 50% of the global smartphone market. This review includes description of the approaches and portrays interesting Android apps deploying them, addressing classification and regression problems. Moreover, they discussed the contributions and critiques of the apps to be useful to students, researchers and developers in mobile computing, human computer interaction, data mining and machine learning. Waranusast et al. [16] assure that judging the size, and therefore weight, of an egg is often important in many food recipes. Hence, they proposed an image processing algorithm for classifying eggs by size from an image displayed on an Android device. A coin of known size is used in the image as a reference object. The coin's radius and the egg's dimensions are automatically detected and measured using image processing techniques. Egg sizes are classified based on their features computed from the measured dimensions using a support vector machine (SVM) classifier. The experimental results show the measurement errors in egg dimensions were low at 3.1% and the overall accuracy of size classification was 80.4%. Singh et al. [17] claimed that extracting patterns and features from a large corpus of data requires the use of machine learning (ML) tools to enable data assimilation and feature identification for plants stress phenotyping. They reported that, four stages of the decision cycle in plant stress phenotyping and plant breeding activities where different ML approaches can be deployed are (i) identification, (ii)

classification, (iii) quantification, and (iv) prediction (ICQP). Moreover, they provided a comprehensive overview and user-friendly taxonomy of ML tools to enable the plant community to correctly and easily apply the appropriate ML tools and best-practice guidelines for various biotic and abiotic stress traits. Poostchi et al. [18] gave an overview of the techniques and discussed the current developments in image analysis and machine learning for microscopic malaria diagnosis. They organized the different approaches published in the literature according to the techniques used for imaging, image preprocessing, parasite detection and cell segmentation, feature computation, and automatic cell classification.

To the best of our knowledge and based on the previous revision, it is possible to mention that none of the previous works performed an analysis of the development and assessment of UIDPs for mobile applications development using digital image processing and machine learning techniques. For this reason, the state-of-the-art was made in two ways, (1) on the use of UIDPs in mobile apps and (2) development and assessment of mobile apps using digital image processing and machine learning techniques in different areas.

Then, Sect. 3 presents an analysis of UIDPs for mobile devices grouped by author and a classification of mobile social media apps.

3 Classification of Mobile Social Media Apps

According to the NetMarketShare [19] statistics service, the most popular operating systems for mobile devices are Android™, Firefox® OS, iOS™ and Windows® Phone. Each operating system has its own identity that is reflected in the appearance and behavior of each graphic element. However, all share fundamental points of view that are manifested in the design of their interfaces such as navigation, dialog boxes and notifications, among others. Different categories of UIDPs proposed by different authors were identified. For mobile devices, there are categories proposed by Neil [20], Tidwell [21], Mari Sheibley [22], Toxboe [5] and UNITiD [23]. In the case of Smart TVs, there are categories proposed by LG Developer [24], Android™ TV [25] and Apple TV [26].

To develop an app, it is firstly necessary to determine the device and type of application to facilitate the human-computer interaction process.

The social media are online communication platforms where the content is created by the users themselves through the use of Web 2.0 technologies, which facilitate the editing, publication and information exchange. There are different types of Social Media applications, based on their content or functionality as well as various classifications. Kaplan and Haenlein [1] and Dao [27] presented a similar social media classification based on social presence and wealth of media, in which they defined six types: (1) collaborative projects, (2) blogs, (3) content communities, (4) sites of social networks, (5) virtual game worlds and (6) virtual social worlds. On the other hand, the DelValle Institute for Emergency Preparedness Learning Center [28] presented a

Table 1 Classification of social media apps

Category	Application considered
Social Network	Facebook®
	Instagram™
Microblogging	Twitter®
	Tumblr™
Streaming	Youtube™
	Netflix™
	Spotify™
	Deezer™

classification and general description of social media platforms by purpose and function, in which they defined seven types: (1) Social Networking, (2) Microblogging, (3) Blogging (Using Publishing Websites), (4) Photo Sharing, (5) Video Sharing or Streaming, (6) Crowdsourcing and (7) Tools for Managing Multiple Social Media Platforms. In our case, the classification used was made based on the apps function and was divided into Social Network, Microblogging and Streaming applications. Table 1 shows the proposed classification and the social media apps considered in each category.

A set of social media apps was analyzed to identify the UIDPs involved in the user graphic interfaces of each app. The social media apps presented were selected as the most popular according to the study presented in [29] and carried out by IAB Spain [30] on trends in Social Media, with the aim of quantifying the evolution of its use, understanding knowledge and use, and evaluating the level of advertising saturation and linkage with other brands. The social media apps analyzed are presented in Table 1.

Next, Sect. 4 presents the result of the analysis of UIDPs in those social media apps reviewed.

4 UIDPs Identified in Social Media Apps

This section presents the result of the analysis of UIDPs in the social media apps. For each social media app analyzed, all its interfaces were reviewed to identify the UIDPs used in its development. Table 2 shows the applications focused on social

Table 2 UIDPs identified in social media apps focused on social networks

Application	UIDPs
Facebook®	Login, video, gallery, menu (horizontal and vertical), comments
Instagram™	Login, video, gallery, ranking, datalist, menu (horizontal and vertical)

Table 3 UIDPs identified in social media apps focused on microblogging

Application	UIDPs
Twitter®	Login, video, menu (horizontal & vertical), gallery, datalist
Tumblr™	Video, search, login, menu, panel, gallery, comments

Table 4 UIDPs identified in social media apps focused on streaming

Application	UIDPs
Youtube™	Video, ranking, gallery, datalist, chat, comments, login
Netflix™	Login, menu, picker, video, gallery, carousel, ranking
Spotify™	Gallery, datalist, login, menu, carousel, music player
Deezer™	Datalist, login, menu, gallery, music player

networks. Table 3 shows the social media applications focused on microblogging. Table 4 shows the social media applications focused on streaming.

Some examples of the interfaces of these applications are depicted in Fig. 1.

As can be seen, social media apps are varied and do not have standards that define the design and development of these kind of apps yet. From this perspective, this work presents a significant contribution by providing an analysis aimed at identifying the most UIDPs used in social domain. This analysis has the objective of establishing design standards and facilitating both the use of best practices and development of mobile applications for social purposes. This analysis reveals that the most used UIDPs in social media apps are Login, Gallery, Video, Carousel and music player, to name a few. Each UIDPs was used according to the characteristics and functionality that the application has. Since the applications are of social nature, the features are common among them. The basic functionalities of these applications share content or resources with other people or, in some cases, consume resources, which can be photographs, music or videos. Hence, the UIDPs most used by this type of applications are Video, Carousel and Galleries. On the other hand, other basic functionalities of the social media applications connect people and share resources among them. That is why among of the most used UIDPs is Login, which allows to identify the users and thus knowing who shared a content or with whom we are interacting.

The main idea of the analysis of social media apps and the identification of the UIDPs used in their development is to identify the most common UIDPs in each type of application, and also to have a mental image or conceptualization of each UIDP, that is to say, if someone thinks of a video player it will always be something similar: a square with the play button in the middle and the progress bar at the bottom, as can be seen in Fig. 2. For each UIDP, its visual representation was established. In order to extract its characteristics and thus be able to fully identify each one of the UIDPs, Fig. 2 shows some examples of the representation of the UIDPs made by freehand.

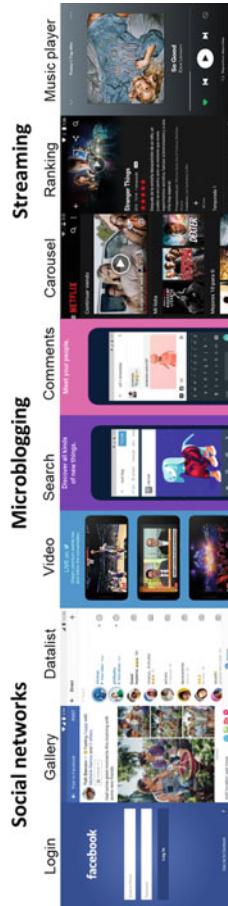


Fig. 1 Interfaces of social media apps

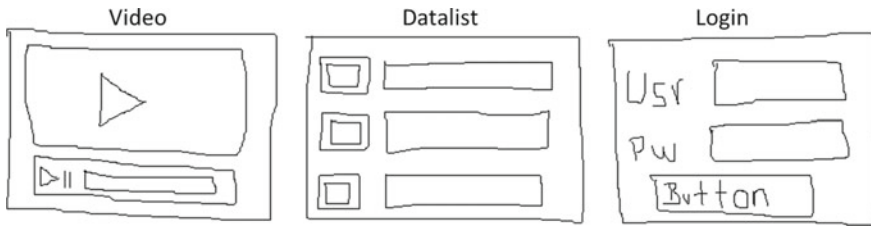


Fig. 2 UIDPs examples

The characterization of the images was carried out in order to obtain characteristics that could help us to classify the UIDPs that will be later analyzed using image processing and decision trees.

The following section explains how to perform the classification of UIDPs using decision trees.

5 Materials and Methods

This section describes all the materials and methods necessary to reproduce the study carried out.

5.1 Decision Trees

A decision tree [31, 32] is a set of rules for the classification of data; a path or blueprint for the solution of a problem. It consists of nodes, branches and leaves. Classification starts at the root node, and proceeds through internal nodes corresponding to the questions about a particular attribute of the problem. Each of these nodes has outgoing branches representing their possible values. Finally, leaf nodes represent a decision about the class variable. Decision-tree building follows two stages: (1) induction and (2) classification. In the first stage, training data are used for the construction of the tree and the assignment of attributes to each node. In the second stage, once the tree has been built, each new object is classified starting at the root node and proceeding through the branches and leaves. The path followed within the tree determines this classification. There are several algorithms for the generation of decision trees, among which are ID3 [33] and C4.5 [34]. C4.5 is one of the most widely used and practical methods for inductive inference [34]. It is a method for approximating discrete-valued functions that is robust to noisy data and capable of learning disjunctive expressions [33, 35]. Learned trees can also be represented as sets of if-then rules to improve human readability. These learning methods are among the most popular of inductive inference algorithms and have been successfully applied to a broad range of tasks

from diagnosing medical cases to assessing credit risk of loan applicants. C4.5 uses as a heuristic a one-step look ahead (hill climbing) procedure, non-backtracking search through the space of all possible decision trees [33, 35, 36].

(a) **Evaluation method: k-fold cross-validation**

We follow the definition of the cross-validation method given by Kohavi [37]. In k-fold cross-validation, we split the database D in k mutually exclusive random samples called the folds: D_1, D_2, \dots, D_k , where such folds have approximately equal size. We train this classifier each time $i \in \{1, 2, \dots, k\}$ using $D \setminus D_i$ and test it on D_i (the symbol \setminus denotes set difference). The cross-validation accuracy estimation is the total number of correct classification divided by the sample size (total number of instances in D). Thus, the k-fold cross validation estimate is:

$$acc_{cv} = \frac{1}{n} \sum_{(v_i, y_i) \in D} \partial(I(D \setminus D_{(i)}, v_i), y_i)$$

where $I(D \setminus D_{(i)}, v_i)$, y_i represents the label assigned by the inducer I to an unlabeled instance v_i , in the set $D \setminus D_{(i)}$, y_i is the class of the instance v_i , n is the size of the complete data set and $\partial(i, j)$ is a function where $\partial(i, j) = 1$ if $i = j$ and 0 if $i \neq j$. In our experiments, the value we use for k is 10.

(b) **Dataset**

The data set was obtained from the extraction of characteristics of the user interface images that represent each UIDP made freehand as mentioned above, for this investigation the analyzed UIDPs are: **Dashboard, Datalist, Login, Video** and **Carousel**, and the characteristics that were obtained from these images are the following:

- (1) Four Hu moments: these are statistical parameters, invariant to the size and position of a figure in an image.
- (2) Number of objects per pattern: this parameter does not consider possible letters.
- (3) The number of objects including letters: the total number of figures in each UIDP, plus letters.
- (4) Number of signatures from the circle figure: the number of objects in the UIDP that are a circle element.
- (5) Number of signatures from the rectangle figure: the number of objects in the UIDP that are a rectangle element.
- (6) Number of signatures from the square figure: the number of objects in the UIDP that are a square element.
- (7) Number of signatures from the triangle figure: the number of objects in the UIDP that are a triangle element.
- (8) Number of signatures from the line figure: the number of objects in the UIDP that are a line element.
- (9) Number of signatures from Accordion contour element: the number of skeletons in the UIDP that are identical to the accordion contour element.

- (10) Number of signatures from Contextual Menu contour element: the number of objects in the pattern-shaped contour that are equal to the contextual menu element.
- (11) Number of signatures from List contour element: the number of objects that are identical to the list contour element.
- (12) Number of signatures from oval figure: the number of objects with a structure similar to that of an oval.
- (13) Number of signatures from a right arrow figure: the number of objects in a UIDP that are similar to the contour of a right arrow figure.
- (14) Number of signatures from a left arrow figure: the number of objects in a UIDP that have a structure identical to that of a left arrow contour figure.
- (15) Number of signatures from contour of letter L: the number of UIDP objects that are similar to the contour of a letter L.

These characteristics were obtained from a set of 500 images, 100 for each UIDP analyzed, the images are freehand images, which represent each UIDP. The extraction was carried out and from these the data set.

(c) **Weka**

Waikato Environment for Knowledge Analysis (WEKA) [38] is an open source software developed at the University of Waikato, located in New Zealand, under a general public license of GNU (General Public License). It consists of a set of machine learning algorithms that are generally used for data mining. WEKA has tools for preprocessing, classification, regression, clustering, selection and visualization.

This software facilitates the classification through decision trees with the J48 algorithm, which is an implementation of the C4.5 algorithms. In addition, WEKA offers other algorithms such as Bayesian networks and Artificial Neural Networks, to name a few. In this research we made use of WEKA versions 3.6.15 and 3.8.3.

5.2 *Methodology*

The objective of the research is a quantitative identification of mobile social media UIDPs' visual features using a combination of digital image processing and machine learning techniques, the following steps are proposed to achieve the objective:

- (1) Data set selection: the data is taken from the resulting characteristics explained in the Dataset section, and the class takes 5 possible values for this case, each value corresponds to an analyzed UIDP, Dashboard, Datalist, Login, Video and Carousel.
- (2) Analysis of data properties: 500 cases were prepared, 100 for each of the classes. Data are studied in order to identify important characteristics, as well as the presence of outliers and null values.

- (3) Transformation of the input data set: according to the previous analysis, the pre-processing and transformation of the data is carried out in order to prepare them for the application of the machine learning techniques.
- (4) Application of the techniques of Machine Learning: the models are constructed making use of WEKA (Waikato Environment for Knowledge Analysis). Since it provides the necessary tools for data modeling and the application of the algorithms, in this case J48 which is the implementation of C4.5.
- (5) Extraction of knowledge: once the technique of Decision Trees is applied, the models are obtained, which will represent the patterns of behavior detected in the variables or the relations between them.
- (6) Interpretation and evaluation of the data: once the models have been obtained, they are validated to verify that the conclusions they provide are valid and satisfactory. In this particular case, the characteristics that take more relevance in the identification and classification of the UIDPs are observed.

Figure 3 shows the sequence of steps followed by the methodology.

5.3 Results and Discussion

The graphic representation of the decision tree obtained as part of the results is shown in Fig. 4. And in Table 5 its respective confusion matrix. The image can be easily seen the most relevant features for classifying UIDPs.

In this work we carried out a quantitative identification of mobile social media UIDPs' visual features using a combination of digital image processing and machine learning techniques. The characteristics of the images were extracted using image processing, the set of characteristics was proposed by us, demonstrating that they are really effective since the accuracy obtained when evaluating the classifier using decision trees was 97.4% (of the 500 cases used were correctly classified 487). For the Dashboard class, 97 cases were correctly classified, for the Datalist class, the total number of cases was correctly classified (100), for the Login class, 94 cases were correctly classified, for the Video class, 98 cases were correctly classified and finally, for the Carousel class, 98 cases were correctly classified.

This helps us to better identify the most relevant characteristics that allow us to more accurately classify the UIDPs using decision trees. And in any case add other features or eliminate the less relevant in the future.

In addition, it is important to mention that although the main objective of the research is a quantitative identification of the visual characteristics of UIDP of mobile social media through a combination of digital image processing and automatic learning techniques, this identification will be used later to generate mobile social applications of automatically, that is, from the automatic recognition of UIDPs in a freehand image, the UIDPs that will compose the social media apps to be generated can be identified.



Fig. 3 Methodology steps

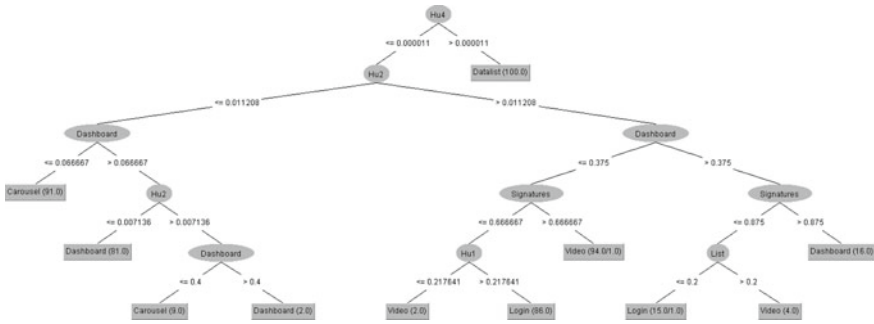


Fig. 4 Decision tree obtained by J48

Table 5 Confusion matrix for J48

Classified as	a	b	c	d	e
a = Dashboard	97	0	0	2	1
b = Datalist	0	100	0	0	0
c = Login	0	0	94	6	0
d = Video	0	0	2	98	0
e = Carousel	1	0	0	1	98

6 Conclusions and Future Work

Mobile social media apps have become very important today, especially in entertainment and in the way people interact. Due to this, it is important to investigate the use of these applications in order to develop appropriate graphical user interfaces for this domain, as well as the implementation of UIDPs to solve existing problems in the design of mobile social media apps. The analysis of UIDPs presented in this paper provides a guide to facilitate choosing the appropriate UIDPs to develop a mobile social media app considering the kind of app to develop. By considering these aspects, the advantage of using UIDPs is that the apps are developed without losing quality in the design, but, above all, without affecting the functionality. Perform a quantitative identification of mobile social media UIDPs’ visual features using a combination of digital image processing and machine learning techniques is of great importance, since thanks to this it is possible to identify and correctly classify a UIDP drawn by freehand, which will allow us to generate the mobile social media app automatically with the component corresponding to that UIDP, that is, if the identified UIDP is a Video then a mobile app with a video player will automatically be generated. As future work, we are considering to expand the analysis presented by including more mobile social media apps that allow evaluating other UIDPs in the social domain. We will also seek to perform an analysis in other knowledge domains such as electronic commerce (e-commerce). Also, we will seek to cover more devices in order

to increase the number of UIDPs analyzed to promote the development of apps in the social domain. And finally, implement the analysis and classification algorithms in a tool that allows the automatic code generation from the digital image processing and machine learning techniques.

Acknowledgements This work was supported by Tecnológico Nacional de México (TecNM) and Artificial Intelligence Research Center of the Universidad Veracruzana, and sponsored by the National Council of Science and Technology (CONACyT) through the program “Estancias Posdoctorales 1er Año 2018-1” and the Mexican Secretariat of Public Education (in Spanish Secretaría de Educación Pública, SEP) through PRODEP (Programa para el Desarrollo Profesional Docente).

References

1. Kaplan, A.M., Haenlein, M.: Users of the world, unite! The challenges and opportunities of Social Media. *Bus. Horiz.* **53**, 59–68 (2010)
2. Bradley, A.J.: A New Definition of Social Media. https://blogs.gartner.com/anthony_bradley/2010/01/07/a-new-definition-of-social-media/
3. Curtis, A.: The Brief History of Social Media. <http://www2.uncp.edu/home/acurtis/NewMedia/SocialMedia/SocialMediaHistory.html>
4. Pressman, R.S.: *Ingeniería del software. Un enfoque práctico* (2010)
5. Toxboe, A.: UI patterns-user interface design pattern library. 15 July 2015, <http://ui-patterns.com/patterns> (2012)
6. Cortes-Camarillo, C.A., Alor-Hernández, G., Olivares-Zepahua, B.A., Rodríguez-Mazahua, L., Peláez-Camarena, S.G.: Análisis comparativo de patrones de diseño de interfaz de usuario para el desarrollo de aplicaciones educativas. *Res. Comput. Sci.* **126**, 31–41 (2016)
7. Videla-Rodríguez, J.-J., Pérez, A.S., Costa, S.M., Seoane, A.: Diseño y usabilidad de interfaces para entornos educativos de realidad aumentada. *Digit. Educ. Rev.* 61–79 (2017)
8. Quispe Rodríguez, A.M.J.: Web usability for colorblind users. *Bridge* **8**, 71–78 (2017)
9. Figeac, J., Chaulet, J.: Video-ethnography of social media apps’ connection cues in public settings. *Mob. Media Commun.* **6**, 407–427 (2018)
10. Sánchez-Morales, L.N., Alor-Hernández, G., Miranda-Luna, R., Rosales-Morales, V.Y., Cortes-Camarillo, C.A.: Generation of User Interfaces for Mobile Applications Using Neuronal Networks BT. In: García-Alcaraz, J.L., Alor-Hernández, G., Maldonado-Macías, A.A., Sánchez-Ramírez, C. (eds.) *New Perspectives on Applied Industrial Tools and Techniques*, pp. 211–231. Springer International Publishing, Cham (2018)
11. Cortes-Camarillo, C.A., Rosales-Morales, V.Y., Sanchez-Morales, L.N., Alor-Hernández, G., Rodríguez-Mazahua, L.: Atila: A UIDPs-based educational application generator for mobile devices. In: 2017 International Conference on Electronics, Communications and Computers, CONIELECOMP 2017 (2017)
12. Markkula, J., Mazhelis, O.: A generic architectural model approach for efficient utilization of patterns: application in the mobile domain. In: *Application Development and Design: Concepts, Methodologies, Tools, and Applications*. pp. 501–529. IGI Global (2018)
13. Moran, K., Bernal-Cárdenas, C., Curcio, M., Bonett, R., Poshyanyk, D.: Machine learning-based prototyping of graphical user interfaces for mobile apps. *arXiv Prepr. arXiv1802.02312* (2018)
14. Umair, A., Nanda, P., He, X., Choo, K.-K.R.: User Relationship Classification of Facebook Messenger Mobile Data using WEKA BT. In *International Conference on Network and System Security* (2018)
15. Basavaraju, P., Varde, A.S.: Supervised learning techniques in mobile device apps for Androids. *SIGKDD Explor. Newsl.* **18**, 18–29 (2017)

16. Waranusast, R., Intayod, P., Makhod, D.: Egg size classification on Android mobile devices using image processing and machine learning. In: 2016 Fifth ICT International Student Project Conference (ICT-ISPC), pp. 170–173 (2016)
17. Singh, A., Ganapathysubramanian, B., Singh, A.K., Sarkar, S.: machine learning for high-throughput stress phenotyping in plants. *Trends Plant Sci.* **21**, 110–124 (2016)
18. Poostchi, M., Silamut, K., Maude, R.J., Jaeger, S., Thoma, G.: Image analysis and machine learning for detecting malaria. *Transl. Res.* **194**, 36–55 (2018)
19. NetMarketShare: Market Share Statistics for Internet Technologies. <https://www.netmarketshare.com>
20. Neil, T.: *Mobile Design Pattern Gallery: UI Patterns for Smartphone Apps*. O'Reilly Media, Inc. (2014)
21. Tidwell, J.: *Designing Interfaces: Patterns for Effective Interaction Design*. O'Reilly Media (2010)
22. Sheibley, M.: *Mobile Patterns* (2013)
23. UNITiD: *Android Patterns*. (2016)
24. LG ELECTRONICS: *LG Developer*. LG (2013)
25. *Android TV: Android TV Patterns* (2015)
26. *Apple TV: Human Interface Guidelines* (2016)
27. Dao, D.V.: Social media classification scheme in online teaching and learning activities: a consideration for educators. *Int. J. Educ. Soc. Sci.* **2**, 85–94 (2015)
28. DelValle Institute for Emergency Preparedness Learning Center: SOCIAL MEDIA PLATFORMS. <https://delvalle.bphc.org/mod/wiki/view.php?pageid=65>
29. Rodríguez Cid, S.: Tendencias en Redes Sociales en 2018. <https://www.cice.es/noticia/tendencias-redes-sociales-2018/>
30. IAB Spain: *Estudio anual de redes sociales 2018* (2018)
31. Breiman, L.: *Classification and Regression Trees*. Routledge (2017)
32. Rokach, L., Maimon, O.: *Data Mining with Decision Trees: Theory and Applications*. World Scientific (2014)
33. Quinlan, J.R.: Induction of decision trees. *Mach. Learn.* **1**, 81–106 (1986)
34. Quinlan, J.R.: *C4.5: Programs for Machine Learning*, Morgan Kaufmann series in machine learning. Morgan Kaufmann Publishers (1993)
35. Mitchell, M.T.: *Machine Learning*. McGraw-Hill, Singapore (1997)
36. Liu, Y., Zheng, Y.F.: One-against-all multi-class SVM classification using reliability measures. In: *Proceedings of the IEEE International Joint Conference on Neural Networks IJCNN'05*, vol. 2, pp. 849–854 (2005)
37. Kohavi, R.: A study of cross-validation and bootstrap for accuracy estimation and model selection. *Ijcai* **14**, 1137–1143 (1995)
38. Hall, M.A., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The WEKA data mining software: an update. *SIGKDD Explor.* **11**, 10–18 (2009)

Fuzzy Modular Neural Model for Blinking Coding Detection and Classification for Linguistic Expression Recognition



Mario I. Chacon-Murguía, Carlos E. Cañedo-Figueroa
and Juan A. Ramirez-Quintana

Abstract EEG signal analysis provides a new alternative to implement brain computer interfaces. Among the possible signals that can be used for brain computer interfaces are signals generated during blinking. This chapter presents a novel fuzzy modular neural model for linguistic expression recognition using blinking coding detection and classification. The linguistic expressions are coded into blinking sequences, and the proposed system analyzes these sequences to detect possible existence of events, expression codes. The blinking signals are first preprocessed to eliminate possible offset and to limit their bandwidth. Then, a new processing step obtains statistical information of the signals and makes them invariant to future users. If a code expressions is detected, it is processed to generate a feature vector with statistical and frequency features. The feature vector is classified for a set of specialized modular neural networks, and finally an output analysis scheme is used to reduce an improper decision. The fuzzy detection systems was tested with signals corresponding to blinking codes, involuntary blinking, and noise and achieved 100% of correct detection, and the final classification of expression with the modular network was 94.26%. Regarding these results, the propose system is considered suitable for applications with seriously impaired persons to establish a basic communication with other person through blinking code generation.

Keywords Brain computer interface · Artificial neural network · Fuzzy logic · Fuzzy neural model · Blinking recognition

M. I. Chacon-Murguía (✉) · C. E. Cañedo-Figueroa · J. A. Ramirez-Quintana
Visual Perception Applications on Robotic Lab, Tecnológico Nacional de Mexico/I. T.
Chihuahua, Mexico City, Mexico
e-mail: mchacon@itchihuahua.edu.mx

C. E. Cañedo-Figueroa
e-mail: cecanedo@itchihuahua.edu.mx

J. A. Ramirez-Quintana
e-mail: jaramirez@itchihuahua.edu.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_46

1 Introduction

Man-machine interfaces, MMI, have been of great interest for researchers during several decades. They intend to establish a simple and fluid communication between people and machines. Recent advances in neurosciences and brain technology have a positive impact in new MMI by processing EEG information. These new MMIs are termed brain computer interfaces, BCI, and they open a new era of opportunities between humans and machines. We can consider BCI as systems that allow detecting, analyzing, and translating brain signals and or their artifacts into commands and messages with the purpose of generating an interaction between a persona with his/her environment or a machine. A more modern conception of BCI defines it like the technology that through brain computer signal processing, it enables a person to influence other persons as well as his/her environment. This last concept includes the capacity of analyzing emotions as well as cognitive states [1].

Some works on BCI reported in the literature are described next. Performance variations over long periods of time is an important issue in BCI. An alternative to reduce this issue is to provide different levels of assistance to the user. Saeedi et al. report in [2] a method to adapt the level of assistance in a motor imagery (MI-BCIs) system. Computational intelligence, CI, theories have been also used considering their inherent organic relationship between artificial neural networks and BCI [3]. Fuzzy logic may play a relevant role because of its natural potential to manage ambiguity, uncertainty, and linguistic processing [4]. Learning is a paramount issue on BCI, a learning scheme that transfer knowledge in a BCI system is documented in [5]. This learning system conveys knowledge in session-to-session or subject to subject scenarios. For more information on BCI applications, the reader is encourage to read [3].

One major area of impact of BCI is rehabilitation systems [6], and another important area of application of BCI is related to provide communication to impaired people, especially people with severe motion, and language impairment or not enable to communicate through normal channels [7]. Examples of specific applications of BCI in these area are; interaction with electronic devices [8], using smart phone, [9], rehabilitation for movement impairment people [10]. Mikołajewska and Mikołajewski [11] reported a study on analysis of BCI in neuro-rehabilitation applied to children. The purpose of the study is to investigating how the BCI technology is applied in this scenario, and what is its scope. The study concludes that in some severe cases of children with neurological deficits, BCIs' may represent the only solution for communication Other works related to blinking aimed to determine mental states of humans are reported in the literature. For example in [12], a method is reported to detect blinking with the purpose to help characterize fatigue states. This method is based on source separation algorithm, and temporal, spatial and frequency features. The work presented by Zhang et al. [13] describes a method to detect blink and heart rate using multi-channel ICA, and it is oriented to determine physical and mental state of humans.

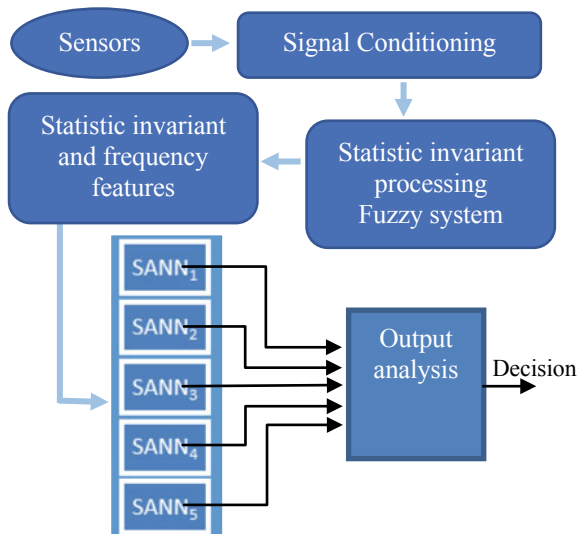
Even many works have been reported in the literature, BCI is considered as an emergent research area, where more research is required considering the potential of brain-generated signals.

This chapter presents the design of a novel fuzzy modular neural model to recognize linguistic expression. The linguistic expressions are coded into blinking sequences. The signals analyzes are not properly EEG signal but signal immerse into the EEG signal termed artifacts. The artifact approach was used because our purpose is to develop a communication system for people who only can blink as a medium of communication [14]. The motivation of this work is to design a system to provide to a seriously impaired person the possibility to establish a basic communication with other person through blinking code generation. The main technical contributions of the work are a fuzzy system to detect the occurrence of events based on linguistic variables, and the neural recognition system based on a modular approach with an output analysis scheme to reduce an improper decision. In addition, the system does not require any prior training or tuning when applied to a new subject, which makes it subject-independent.

2 General Description of the System

The proposed system is illustrated in Fig. 1, and a brief explanation is provided next. An EEG sensor acquires the events representing the possible expressions. Then the signals are conditioned to eliminate an amplitude offset and to limit their bandwidth. The next stage is a new processing step to obtain statistical information of the signals and to make them invariant to future users. These features are then used by the fuzzy

Fig. 1 General scheme of the proposed system including the fuzzy detection and modular neural network systems



system to detect possible code expressions. If code expressions are detected, they are analyzed for feature extraction, yielding a statistical and frequency feature vector. The feature vector is classified for each one of the specialized neural networks, SANN, and finally each network output is examined to reduce an improper decision when the final decision is taken.

3 Data Acquisition

The signals of interest in this work are the signals generated during human blinking. People in the age of 18–28 years old blink 10.3 ± 3.1 times per minute, and apparently, there is not difference between woman and man. The blinking time may change according to external or internal conditions of the person. External factors that may increase blinking may be wind and heat. Instead, cold and a null or low presence of wind may decrease blinking. Among the internal factors that may affect blinking are; mood, fatigue, depression, stress, brain diseases or spasmodic motion (tics) [15]. In this work, blinking signals are coded to represent four linguistic expressions. These selected expressions were selected considering the most basic expression for communication. However, they may be changed to other expressions. The signals were obtained from five subjects, 3 males and 2 females around 25 years old. The total number of samples was 279 signals of different events. The signals used to represent blinking were acquired with a commercial 14 channel wireless EEG equipment. The signals of interest were obtained from the electrodes corresponding to the channels AF3, AF4, F7 and F8 according to the international system 10–20 because they convey blinking information. These signals are represented in this work as xAF3, xAF4, xF7, xF8. The sessions to obtain the signals were achieved in a room with semi-controlled environment where people were talking, listening music in a moderated volume, there were working PCs, and there was a normal illumination condition. The test subjects were sat in a chair with his/her hands and arms resting on the armrests, and looking at the monitor of the PC where the test was performed. The time duration of each session was different because several tests were performed to acquire signals corresponding to one linguistic expression, which takes between 1 and 5 min. In addition, sessions to acquire signals of combined linguistic expressions with duration of 2–10 min were performed. The codification blinking—expression involves time series signals which include eyelid closing time and eyelid open time. The linguistic expressions and their corresponding codes are in Table 1. Any other situation present in the signals different from the events described in Table 1 is considered noise, c_{no} . It is important to specify that each blink must be similar to a natural blink, which duration is approximately 400 ms. Open and close eyes must be approximately an event of 1 s. The time specified is defined as approximate because the blinking codes are generated by different persons, and they may vary from one person to another.

Table 1 Linguistic expressions, codes

Expression	Notation	Code
<i>Assertion</i>	c_y	5 blinks
<i>Negation</i>	c_n	1 blink - 1st sec, open eyes - 2nd, 1blink - 3er, eyes open - 4th, 1blink - 5sh
<i>Not know</i>	c_{nk}	2 blinks - 1st and the 2nd seconds, eyes close - 3er, and 2 blinks - 4th and the 5th
<i>More or less</i>	c_{mol}	2 blinks in the 1st and the 2nd seconds, eyes open - 3er, and 2 blinks - 4th and 5th

4 Signal Conditioning

Before the signal $x_{AF3}, x_{AF4}, x_{F7}, x_{F8}$ can be used for blink event detection, and code recognition, they must be preprocessed. The preprocessing stage involves three processes, signal offset removal, filtering and windowing. The offset removal is needed because the signals provided by the EEG equipment include an offset of 4100 mV. Thus, the offset is removed from the original signal x_{EEGi}

$$x_i = x_{EEGi} - 4100 \text{ mV} \quad \text{for } i = \{AF3, AF4, F7, F8\}. \tag{1}$$

In addition, the EEG signal contains frequency information beyond the frequencies of interest for blinking analysis. Therefore, x_i is filtered out to frequencies of interest 16 Hz, with a low pass filter

$$H : x_i \rightarrow x_f \tag{2}$$

where

$$|H| = \frac{1}{\sqrt{1 + (\omega/\omega_c)^{2p}}} \tag{3}$$

is a first order Butterworth filter $p = 1$, and cutoff frequency $\omega_c = 16$ Hz. A value of 16 Hz is used due to the transition band. Then, x_f is normalized with respect its mean

$$\tilde{x}_f = x_f - \mu_{x_f} \tag{4}$$

where μ_{x_f} is the mean of x_f .

N is the number of samples of x_f . Finally, since the main purpose of the first stage of the proposed system is to detect possible events corresponding to expressions, it is necessary to perform a windowing process on \tilde{x}_f . A window of $K = 258$ samples with a proposed overlap of 64 samples is defined. This window corresponds to 2 s since the sampling frequency of the EEG system is 128 samples/second. The window size

is determined considering that there may be persons that may blink in times greater than 400 ms [16], and the window size is sufficient to include low rate blinking. This windowing process is defined by

$$W : \tilde{x}_f \rightarrow \tilde{x}_{fk} \quad k = 1, \dots, K \quad (5)$$

where W is a sampling process using a rectangular window.

5 Mamdani Fuzzy Event Detection System

Detection of expression events is achieved through different stages. The first stage consists on computing the maximum variance of each \tilde{x}_{fk} in order to normalize the variance information. This normalization makes the variance information subject-independent, regarding different variance values, and the system does not need to be adjusted to each subject. This step is important for the expression recognition system so the proposed system can be used with any person. This computation must be done in \tilde{x}_{fk} which includes the expression *Not know* because in this expression the variance reaches maximum values. The variance and mean per window per channel are

$$\sigma_{\tilde{x}_{fk}}^2 = \frac{1}{K-1} \sum_{j=1}^K (\tilde{x}_{fjk} - \mu_{\tilde{x}_{fk}})^2 \quad \mu_{\tilde{x}_{fk}} = \frac{1}{K} \sum_{j=1}^K \tilde{x}_{fjk} \quad (6)$$

where the maximum channel variance is

$$\sigma_{\tilde{x}_{fk}}^2 \max = \max(\sigma_{\tilde{x}_{fk}}^2) \quad \text{for } i = \{AF3, AF4, F7, F8\} \quad (7)$$

The variance of each windowed signal of each channel normalized to its maximum variance becomes

$$\hat{\sigma}_{\tilde{x}_{fki}}^2 = \frac{\sigma_{\tilde{x}_{fk}}^2}{\sigma_{\tilde{x}_{fk}}^2 \max} \quad \text{for } i = \{AF3, AF4, F7, F8\} \quad (8)$$

and the values $\hat{\sigma}_{\tilde{x}_{fki}}^2$ will be the input linguistic variables for the Mamdani fuzzy event detection system, *MFEDS*. In order to use the $\hat{\sigma}_{\tilde{x}_{fki}}^2$ information in the fuzzy system it is fuzzified as follows

$$\hat{\sigma}_{\tilde{x}_{fki}}^2 = \{Low, Medium, High\} \quad (9)$$

where *Low* is a zeta function $z(w; a, b) = z(w; 0.06, 0.09)$, *Medium* is a trapezoidal function $t(w; a, b, c, d) = z(w; 0.05, 0.08, 0.3, 0.4)$, and *High* is a *s* function $s(w; a, b) = z(w; 0.3, 0.5)$. The linguistic output variable is

$$Detection = \{No\ event, Seems, Event\} \tag{10}$$

where *No event* is a zeta membership function $z(w; a, b) = z(w; 0.01, 0.2)$, *Seems* is a Gaussian membership function $g(w; a, b) = z(w; 0.1, 0.4)$, and *Event* is a *s* membership function $s(w; a, b) = z(w; 0.5, 0.7)$. The five membership functions were determined by statistical analysis.

Thus, the Mamdani fuzzy event detection system *MFEDS* is defines by

$$MFEDS : \phi_c \left(\hat{\sigma}_{\tilde{x}_{fjAF3}}^2, \hat{\sigma}_{\tilde{x}_{fjAF4k}}^2, \hat{\sigma}_{\tilde{x}_{fjAF7k}}^2, \hat{\sigma}_{\tilde{x}_{fjAF8k}}^2 \right) \Rightarrow \{No\ event, Seems, Event\} \tag{11}$$

where ϕ_c is the max-min composition operator. The Mamdani system is expressed by rules of the type

$$R_m : \text{if } \hat{\sigma}_{\tilde{x}_{fjAF3}}^2 \text{ is } A_{m1} \wedge, \dots, \hat{\sigma}_{\tilde{x}_{fjAF8k}}^2 \text{ is } A_{mn} \Rightarrow Detection \text{ is } B_m \tag{12}$$

where A_m and B_m correspond the possible fuzzy values of the input and output variables previously defined. The system consists of 27 fuzzy rules. The rules to determine event, *Seems*, or *Not event* were determined considering that an event is detected if at least two or more signal \tilde{x}_{fjkj} , have values of *Mean* or *High*. Otherwise, *No event* will be reported. This criterion avoids false detection in situations where noise is present in one channel. The defuzzification method used is the centroid. Some of the Mamdani rules are

- $R_1 : \text{if } \hat{\sigma}_{\tilde{x}_{fjAF3}}^2 \text{ is } Low \wedge \hat{\sigma}_{\tilde{x}_{fjAF4}}^2 \text{ is } Low \wedge \hat{\sigma}_{\tilde{x}_{fjAF7}}^2 \text{ is } Low \wedge \hat{\sigma}_{\tilde{x}_{fjAF8}}^2 \text{ is } Low \Rightarrow Detection \text{ is } Not\ event$
- $R_3 : \text{if } \hat{\sigma}_{\tilde{x}_{fjAF3}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF4}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF7}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF8}}^2 \text{ is } High \Rightarrow Detection \text{ is } Event$
- ⋮
- $R_{25} : \text{if } \hat{\sigma}_{\tilde{x}_{fjAF3}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF4}}^2 \text{ is } Medium \wedge \hat{\sigma}_{\tilde{x}_{fjAF7}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF8}}^2 \text{ is } High \Rightarrow Detection \text{ is } Event$
- $R_{26} : \text{if } \hat{\sigma}_{\tilde{x}_{fjAF3}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF4}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF7}}^2 \text{ is } Medium \wedge \hat{\sigma}_{\tilde{x}_{fjAF8}}^2 \text{ is } High \Rightarrow Detection \text{ is } Event$
- $R_{27} : \text{if } \hat{\sigma}_{\tilde{x}_{fjAF3}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF4}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF7}}^2 \text{ is } High \wedge \hat{\sigma}_{\tilde{x}_{fjAF8}}^2 \text{ is } Medium \Rightarrow Detection \text{ is } Event$

The fuzzy value of the linguistic output variable *Detection* indicates at what level the linguistic input variable values represent an expected event, a blinking code. Based on an analysis of the values of the variable *Detection* it was observed that a value of 0.2 corresponds to the linguistic expression *Not know*, and values around 0.1 are associated to the linguistic expressions *Assertion*, *Negation*, and *More or less*. Therefore, a final threshold of 0.09 over the defuzzification value was used to

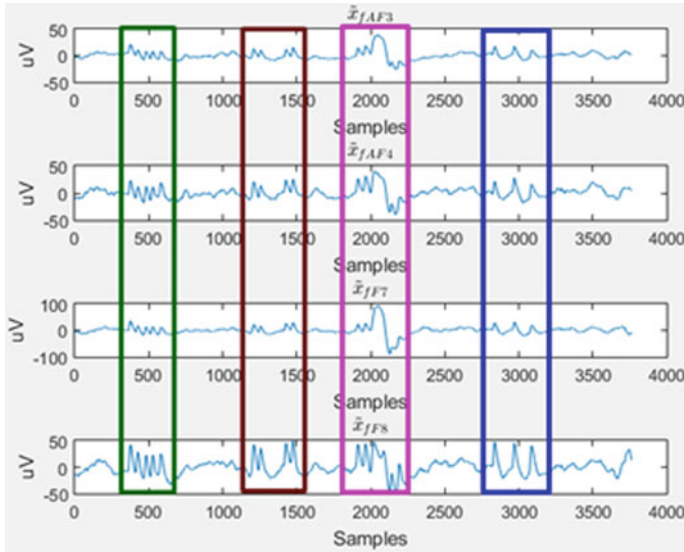


Fig. 2 Illustration of signals $\tilde{x}_{fik}, i = \{AF3, AF4, F7, F8\}$ of the events, *Assertion* (green), *More or less* (brown), *Not know* (pink), and *Negation* (blue)

determined occurrence of an event. Figure 2 illustrates the plots of the signals $\tilde{x}_{fik}, i = \{AF3, AF4, F7, F8\}$ corresponding to the four events marked in colored rectangles, *Assertion* (green), *More or less* (brown), *Not know* (pink), and *Negation* (blue).

If the fuzzy system detects an event, the information corresponding to the window k of each $\tilde{x}_{fik}, i = \{AF3, AF4, F7, F8\}$ is used to generate a new signal that will be used to achieve the linguistic expression recognition process. That is, considering the output of the fuzzy system

$$D_k = \phi_c \left[\hat{\sigma}_{\tilde{x}_{fkAF3}}^2, \hat{\sigma}_{\tilde{x}_{fkAF4}}^2, \hat{\sigma}_{\tilde{x}_{fkF7}}^2, \hat{\sigma}_{\tilde{x}_{fkF8}}^2 \right] \tag{13}$$

the new signal E_k is defined by

$$E_k = \begin{cases} \frac{(\tilde{x}_{fkAF3} + \tilde{x}_{fkAF4} + \tilde{x}_{fkF7} + \tilde{x}_{fkF8})}{4} & D_k \geq 0.09 \\ 0 & D_k < 0.09 \end{cases} \text{ for } k = 0, \dots, K \tag{14}$$

The *MFEDS* system was tested with signals corresponding to blinking codes, involuntary blinking, and noise where 113 true events were present. The performance of the *MFEDS* turned to be 100% hits without any false alarm. Figure 3 shows an example of event detection indicated with rectangular pulses in the last plot. The width of the pulse corresponds to the time duration of the event expression.

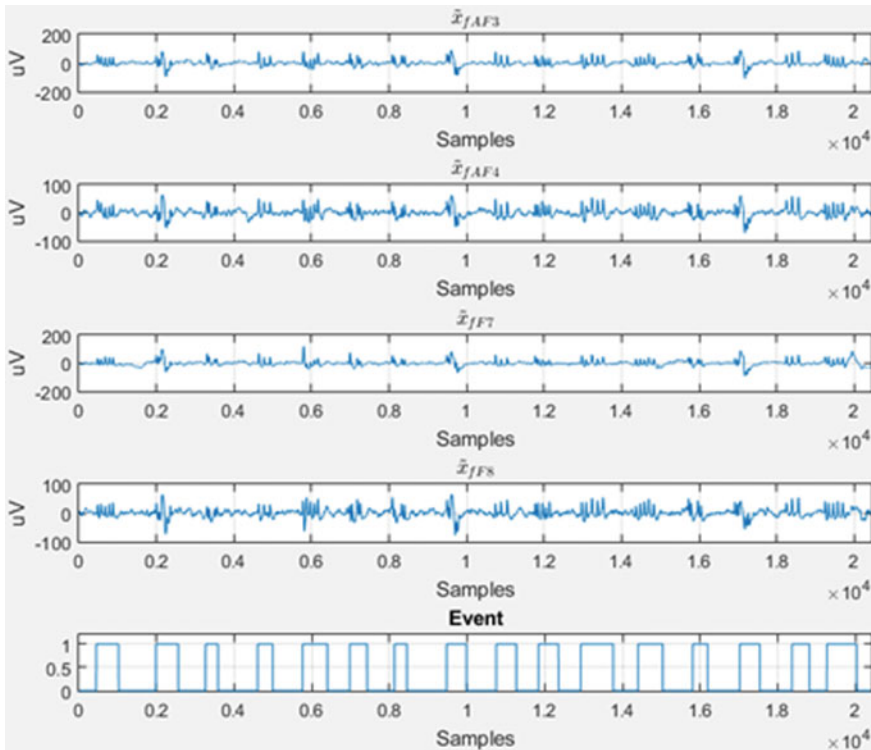


Fig. 3 Example of event detection by the MFEDS. Last plot indicates with pulses the occurrence of event

6 Mamdani Fuzzy Event Detection System

This section describes the design of the modular neuronal linguistic expression recognition system including the feature extraction process, the modular neural network design, and the output analysis scheme used to reduce an improper decision.

6.1 Feature Extraction

The feature extraction stage starts with a normalization of the signal E_k for $k = 0, \dots, K$ in order to make it invariant to amplitude for different persons

$$E_{nk} = E_k / \max(E_k) \tag{15}$$

then, the features are extracted from E_{nk} yielding the feature vector, which is a combination of time, statistical, and frequency features.

$$\mathbf{X}_E = [\sigma_{E_{nk}} \ E_{E_{nk}} \ M_{E_k} \ \mu_\Delta \ \mu_\theta \ \mu_\alpha]^T \quad (16)$$

The first feature is the statistic

$$\sigma_{E_{nk}} = \sqrt{\frac{1}{N-1} \sum_{i=1}^n (E_{nk} - \mu_{E_{nk}})^2} \quad (17)$$

The second is the energy feature [17]

$$E_{E_{nk}} = \sum_{n=0}^N |E_{nk}|^2 \quad (18)$$

The third feature corresponds to the number of maximums in E_{nk} , which correspond to the blinking peaks. The threshold value of 0.3 was selected because the blinking peaks are greater than this value.

$$M_{E_k} = |\{p : p \text{ is a maximum of } E_{nk} \geq 0.3\}| \quad (19)$$

The last three features are obtained from the frequency domain. The frequency features correspond to the mean energy of the waves delta, theta, and alpha

$$\mu_\Delta = \frac{1}{N} \sum_{\omega=0}^N |E_{nk}(\omega)|^2 \quad \text{for } 0 \leq \omega \leq 4 \text{ Hz} \quad (20)$$

$$\mu_\theta = \frac{1}{N} \sum_{\omega=0}^N |E_{nk}(\omega)|^2 \quad \text{for } 4 < \omega \leq 7 \text{ Hz} \quad (21)$$

$$\mu_\alpha = \frac{1}{N} \sum_{\omega=0}^N |E_{nk}(\omega)|^2 \quad \text{for } 7 < \omega \leq 13 \text{ Hz} \quad (22)$$

where $E_{nk}(\omega)$ is the discrete Fourier transform of E_{nk} , and the delta, theta, and alpha waves correspond to the bands $\omega < 4$ Hz, $4 \text{ Hz} < \omega < 7$ Hz, and $8 \text{ Hz} < \omega < 13$ Hz, respectively [18]. Based on several experiments we discover a correlation between the increment of energy on these waves and the presence of blinking. Therefore, the features of these waves were incorporated to the feature vector because they provide extra information of blinking occurrence.

6.2 Modular Neural Network Design

EEG signal suffer from different types of variations due to noise during signal acquisition, intra-person and inter-person signal changes, environment changes, among others. Therefore, designing a system to recognize events based on EEG signal or artifacts is not a trivial task. As it was stated in this work, the proposed system must recognize EEG events artifacts and must be robust to subjects and environment issues. Considering the complexity of this system, it was decided to follow a modular approach instead the monolithic one. The modular neural network, MNN, designed consists of 5 specialized networks. The first network is trained to recognize the expression *assertion*, yes, the second for *negation*, the third network recognize *more or less*, the fourth *not know*, and the fifth is trained to recognize noise, incorrect codes, and involuntary blinks. Thus the classes are $C = \{C_y, C_n, C_{nk}, C_{mol}, C_{no}\}$. The modular neural network system is defined by

$$\psi : \mathbf{X}_E \rightarrow C = \{c_y, c_n, c_{nk}, c_{mol}, c_{no}\} \tag{23}$$

where,

$$\psi = D_{O\varphi}(\varphi_y(\mathbf{X}_E), \varphi_n(\mathbf{X}_E), \varphi_{nk}(\mathbf{X}_E), \varphi_{mol}(\mathbf{X}_E), \varphi_{no}(\mathbf{X}_E)) \tag{24}$$

D_{op} is the output analysis scheme to reduce an improper decision, and $\varphi(\bullet)$ are the specialized neural networks, $\varphi_y(\bullet), \varphi_n(\bullet), \varphi_{nk}(\bullet), \varphi_{mol}(\bullet), \varphi_{no}(\bullet)$, for the classes $c_y, c_n, c_{nk}, c_{mol}$, and c_{no} respectively.

The architecture of each specialized neural network is a 3-layer feedforward perceptron network with 6 inputs, and two output neurons. One network output measures the membership to the corresponding class and the other output measures the not membership. The training algorithm used was scaled conjugate gradient backpropagation. The activation function of the hidden neurons is tan-sigmoid. The activation function of the output neurons is of the type softmax or normalized exponential function. This type of function is employed to keep the largest values and discriminate values, which are far from the maximum value. The normalized exponential function is defined as

$$\vartheta(\mathbf{Z})_j = \frac{e^{z_j}}{\sum_{l=1}^L e^{z_l}} \quad \text{for } j = 1, \dots, L \tag{25}$$

where \mathbf{Z} correspond to the outputs of the hidden neurons of each network $\varphi(\bullet)$.

A series of experiments involving the change of the number of neurons in the hidden layer between 4 and 21 neurons determined the design of the MNN. The experiments were done with 295 samples, 59 for each class, 70% was used for training, and 15% for validation and testing. The best performance metric considering the training, validation and test, determines best network, turned to be the one with 13 hidden neurons. Since each network has two outputs indicating membership and

Table 2 Performance on hits of the five networks

Performance in %				
Network	Training (%)	Validation (%)	Testing (%)	Average (%)
$\varphi_y(\bullet)$	97.3	93.4	100	96.9
$\varphi_n(\bullet)$	98.1	97.7	97.7	98.0
$\varphi_{nk}(\bullet)$	96.1	95.2	95.5	95.6
$\varphi_{mol}(\bullet)$	91.3	90.9	90.9	91.2
$\varphi_{no}(\bullet)$	96.1	95.8	100	97.3

non-membership to the class the network represents, the specification of the targets are 59 out of the 295 targets indicate membership and 236 non-membership.

Table 2 shows the performance of each network to recognize de classes $C = \{C_y, C_n, C_{nk}, C_{mol}, C_{no}\}$, and Fig. 4 illustrates the total confusion matrices per specialized neural network. Results shown in Table 2, and in the confusion matrices indicate that each one of the networks presents a high performance as expect due to their specialization.

Since the decision of the system is of paramount importance to the person using the system, it was decided to implement a verification of the neuron output information on each network before the final decision. The scheme used is similar to one reported in [19] where the decision of a system is critical. The main idea is to make a decision on relevant difference outputs and not just using the greatest output win criterion. If we just implement the greatest output win criterion, there may be situations where we decide on one class with a small amount of information, let say one output is 0.9 and other output is 0.95. In this case and cases where both values are very small, the system may take an incorrect decision. In order to assure a correct final decision, the following output analysis scheme for each network output is proposed

$$D_{O_{\varphi_y}} = \begin{cases} O_{\varphi_y} & \text{if } (O_{\varphi_y} > 0.5) \text{ AND } (\bar{O}_{\varphi_y} > \bar{O}_{\varphi_y} + 0.2) \\ 0 & \text{otherwise} \end{cases} \quad (26)$$

$$D_{O_{\varphi_n}} = \begin{cases} \bar{O}_{\varphi_n} & \text{if } (\bar{O}_{\varphi_n} > 0.5) \text{ AND } (O_{\varphi_n} > \bar{O}_{\varphi_n} + 0.2) \\ 0 & \text{otherwise} \end{cases} \quad (27)$$

$$D_{O_{\varphi_{nk}}} = \begin{cases} O_{\varphi_{nk}} & \text{if } (O_{\varphi_{nk}} > 0.5) \text{ AND } (O_{\varphi_{nk}} > \bar{O}_{\varphi_{nk}} + 0.2) \\ 0 & \text{otherwise} \end{cases} \quad (28)$$

$$D_{O_{\varphi_{mol}}} = \begin{cases} O_{\varphi_{mol}} & \text{if } (O_{\varphi_{mol}} > 0.5) \text{ AND } (O_{\varphi_{mol}} > \bar{O}_{\varphi_{mol}} + 0.2) \\ 0 & \text{otherwise} \end{cases} \quad (29)$$

$$D_{O_{\varphi_{no}}} = \begin{cases} O_{\varphi_{no}} & \text{if } (O_{\varphi_{no}} > 0.5) \text{ AND } (O_{\varphi_{no}} > \bar{O}_{\varphi_{no}} + 0.2) \\ 0 & \text{otherwise} \end{cases} \quad (30)$$

where O_φ represents the membership output of the network to the corresponding class, and \bar{O}_φ the non-membership to the class, and DO_φ is the final decision. Finally

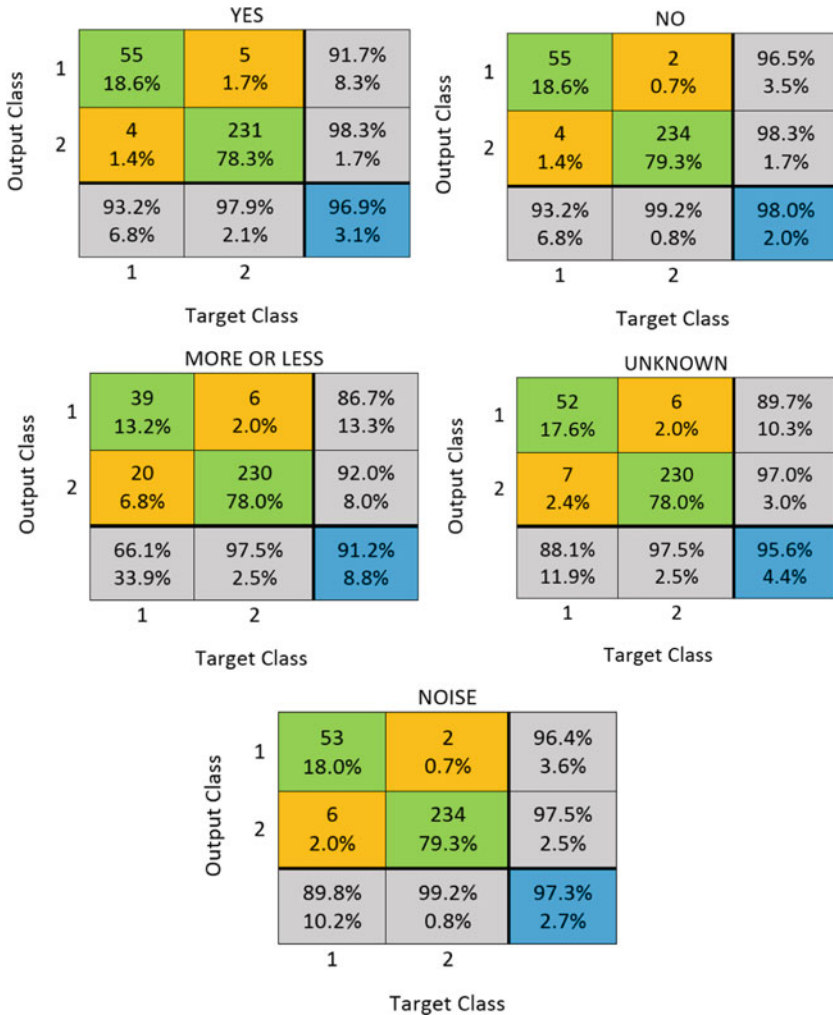


Fig. 4 Confusion matrices of each specialized network at recognizing the classes *assertion*, *yes*, *negation*, *more or less*, *not know*, and *noise*

the class C assigned for the expression detected will be

$$C = c_i \text{ if } D_{O\varphi_i} > D_{O\varphi_j} \quad \forall i, j = \{y, n, nk, mol, no\} \quad i \neq j \quad (31)$$

In order to determine the performance of the proposed model, it was tested with the signals from 5 different subjects, 3 males and 2 females around 25 years old. The total number of samples was 279 of different events achieving a performance of 94.26% of correct classification.

7 Results and Conclusions

This paper presented a fuzzy modular neural model for linguistic expression recognition using blinking coding detection and classification. The points in the design is that the system is tolerant to noise during signal acquisition, intra-person and inter-person signal, and environment changes. The system also does not require any prior training or tuning when applied to a new subject, which makes it subject-independent. The system was designed considering the synergism of fuzzy and neural systems. The fuzzy system detects event expressions considering statistical information represented as linguistic variables. The fuzzy system achieves a performance 100% of correct blink detection, which outperforms the results of new methods like in [12] which reports a true detection rate of 89% and a false alarm rate of 3%. This result shows the potential to process linguistic information on EEG signals or artifacts. Once the events were detected, their classification was done with a modular neural network with five specialized networks. The use of the modular network with the output analysis scheme reduced an improper decision, allowed a high performance of classification of 95.8%. Regarding these results, the proposed system is considered suitable for applications with seriously impaired persons to establish a basic communication with other person through blinking code generation.

Acknowledgements The authors greatly appreciate the support of Tecnológico Nacional de Mexico under grant 5684.16-P to develop this work.

References

1. Lance, B.J., Kerick, S.E., Ries, A.J., Oie, K.S., McDowell, K.: Brain-computer interface technologies in the coming decades. *Proc. IEEE* **100**(3), 1585–1599 (2012)
2. Saeedi, S., Chavarriaga, R., Leeb, R., Millán, J.R.: Adaptive assistance for brain-computer interfaces by online prediction of command reliability. *IEEE Comput. Intell. Mag.* **11**(1), 32–39 (2016)
3. All in the Mind Engineering & Technology Magazine. Available at <http://eandt.theiet.org/magazine/2015/06/index.cfm>. Accessed 15 Jan 2017
4. Abbass, H., Guan, C., Chen-Tan, K.: Computational intelligence for brain computer interface. *IEEE Comput. Intell. Mag.* **11**(1), 18 (2016)
5. Cavrini, F., Bianchi, L., Quitadamo, L.R., Saggio, G.: A fuzzy integral ensemble method in visual P300 brain-computer interface. *Comput. Intell. Neurosci.* **2016**(8), 1–9 (2016)
6. Formaggio, E., Masiero, S., Bosco, A., Izzi, F., Piccione, F., Del Felice, A.: Quantitative EEG evaluation during robot-assisted foot movement. *IEEE Trans. Neural Syst. Rehabil. Eng.* **25**(9), 1633–1640 (2017)
7. Zhang, W., Sun, F., Tan, C., Liu, S.: Low-rank linear dynamical systems for motor imagery EEG. *Comput. Intell. Neurosci.* **2016**(12), 1–7 (2016)
8. Abdulkader, N.S., Ayman, A., Mostafa-Sami, M.: Brain computer interfacing: applications and challenges. *Egypt. Inform. J.* **16**(2), 213–230 (2015)
9. Jayaram, V., Alamgir, M., Altun, Y., Schölkopf, B., Grosse-Wentrup, M.: Transfer learning in brain-computer interfaces. *IEEE Comput. Intell. Mag.* **2015**(12), 1–20 (2015)

10. Amali-Rani, B.J., Umamakeswari, A., Sree-Madhubala, J.: An approach toward wireless brain-computer interface system using EEG signals: a review. *Natl. J. Physiol. Pharm. Pharmacol.* **5**(5), 350–356 (2015)
11. Soman, S., Murthy, B.K.: Using brain computer interface for synthesized speech communication for the physically disabled. *Proc. Comput. Sc.* **46**, 292–298 (2015)
12. Roy, R.N., Charbonnierb, S., Bonnet, S.: Eye blink characterization from frontal EEG electrodes using source separation and pattern recognition algorithms. *Biomed. Signal Process. Control* **14**(11), 256–264 (2014)
13. Zhanga, C., Wua, X., Zhanga, L., Hea, X., Lva, Z.: Simultaneous detection of blink and heart rate using multi-channel ICA from smart phone videos. *Biomed. Signal Process. Control* **33**(3), 189–200 (2017)
14. Singh, B., Wagatsuma, H.A.: Removal of eye movement and blink artifacts from EEG data using morphological component analysis. *Comput. Math. Methods Med.* **2017**(2), 1–17 (2017)
15. Kruis, A., Slagter, H.A., Bachhuber, D.R.W., Davidson, R.J.: Effects of meditation practice on spontaneous eye blink rate. *Soc. Psychophysiological Res.* **53**(5), 749–758 (2016)
16. Kompoliti, K.: Cognitive assessments and Parkinson's disease. *The Encyclopedia of Movement Disorders*, pp. 231–236. Elsevier, Academic Press, Chicago (2015)
17. Sepulveda, R., Montiel, M.O., Díaz, G., Gutierrez, D., Castillo, O.: EEG signal classification using artificial neural networks. *Comput. Syst.* **19**(1), 69–88 (2015)
18. Manasi, S.D., Trivedi, A.R.: Gate/source-overlapped heterojunction tunnel FET-based LAM-STAR neural network and its application to EEG signal classification. In: *Proceedings of International Joint Conference on Neural Networks* (2016)
19. Chacon-Murguia, M.I., Quezada-Holguín, Y., Rivas-Perea, P., Cabrera, S.: Dust storm detection using a neural network with uncertainty and ambiguity output analysis. *Lecture Notes in Computer Science.* 21587(2_33), pp 305–313 (2011)

Hybrid Intelligent Systems

A Genetic Algorithm Based Approach for Word Sense Disambiguation Using Fuzzy WordNet Graphs



Sonakshi Vij, Amita Jain and Devendra Tayal

Abstract Due to the ever-evolving nature of human languages, the ambiguity in it needs to be dealt with by the researchers. Word sense disambiguation (WSD) is a classical problem of natural language processing which refers to identifying the most appropriate sense of a given word in the concerned context. WordNet graph based approaches are used by several state-of-art methods for performing WSD. This paper highlights a novel genetic algorithm based approach for performing WSD using fuzzy WordNet graph based approach. The fitness function is calculated using the fuzzy global measures of graph connectivity. For proposing this fitness function, a comparative study is performed for the global measures edge density, entropy and compactness. Also, an analytical insight is provided by presenting a visualization of the control terms for word sense disambiguation in the research papers from 2013 to 2018 present in Web of Science.

Keywords Genetic algorithm · Fuzzy WordNet graph · Fuzzy graph connectivity measures · WordNet · Word sense disambiguation

1 Introduction

WSD is a well known problem in natural language processing which refers to identifying the most appropriate sense of a given word in the concerned context. Several approaches have been discussed in the literature to deal with this problem and many algorithms are suggested. Graph based approaches have recently become very popular for serving this purpose.

WordNet graph based methods are used by several state-of-art methods for performing unsupervised and semi-supervised WSD [1–3].

S. Vij (✉) · D. Tayal

Department of CSE, Indira Gandhi Delhi Technical University for Women, New Delhi, India
e-mail: sonakshi.vij92@gmail.com

A. Jain

Department of CSE, Ambedkar Institute of Advanced Communication Technologies and Research, New Delhi, India

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_47

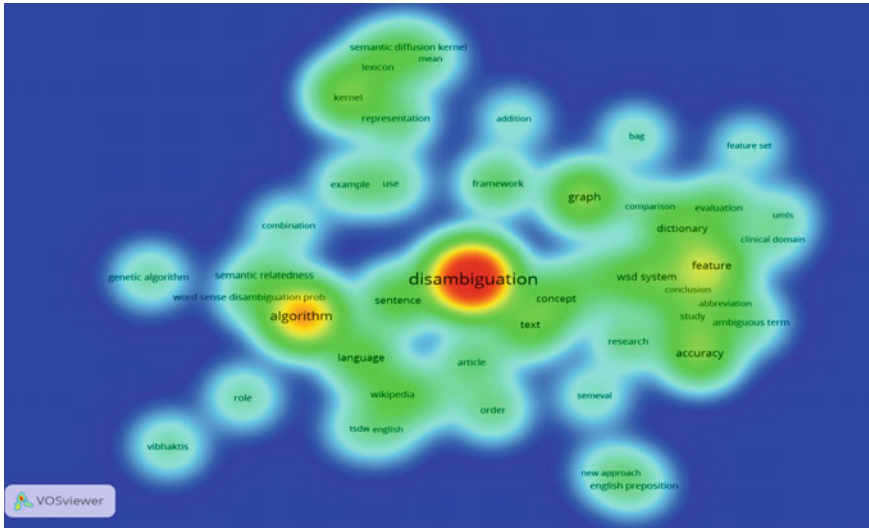


Fig. 1 Control terms in word sense disambiguation

In order to gain more understanding about this topic, the research papers present in Web of Science in the field of word sense disambiguation in the last 5 years i.e. 2013–2018 were analyzed to find out the control terms occurring in them. These terms highlight the most significant concepts covered/mentioned in these research papers. This is visualized in VOSViewer and shown in Fig. 1.

Major applications of WSD include:

- Question Answering [4]
- Processing Bio-Medical Documents [5, 6]
- Machine Translation [7]
- Query Expansion [8]
- Data Processing in Clinical databases [9–11].

A closer look at the state-of-art reveals that metaheuristic approaches like genetic algorithm, ant colony optimization etc. have played a crucial role in resolving issues of WSD [12–17]. Hence, in this paper we have adopted genetic algorithm for performing graph based WSD. The rest of the paper discusses the proposed work with proper detailed implementation and highlights the results.

2 Proposed Work

This section highlights the proposed method of fuzzy WordNet graph based WSD that uses concepts of genetic algorithm. In a nutshell, the process is summarized as shown in Fig. 2.

The process initiates when the user inputs the text. The query is divided into individual tokens and is tagged according to the relevant part-of-speech. The population initiation process begins when a Fuzzy WordNet graph has to be created for the content words using semantic relations hypernym, hyponym, holonyms and meronyms with their corresponding edge weights [1]. The following local graph connectivity measures are to be calculated for every node in the graph [2]:

a. Fuzzy Degree(N_{FD}):

$$N_{FD} = \frac{degf^{(v)}}{|v| - 1} \tag{1}$$

b. Fuzzy Closeness(N_{FC}):

$$N_{FC} = \sum_{k=1}^{tot} \frac{1}{w_k} \tag{2}$$

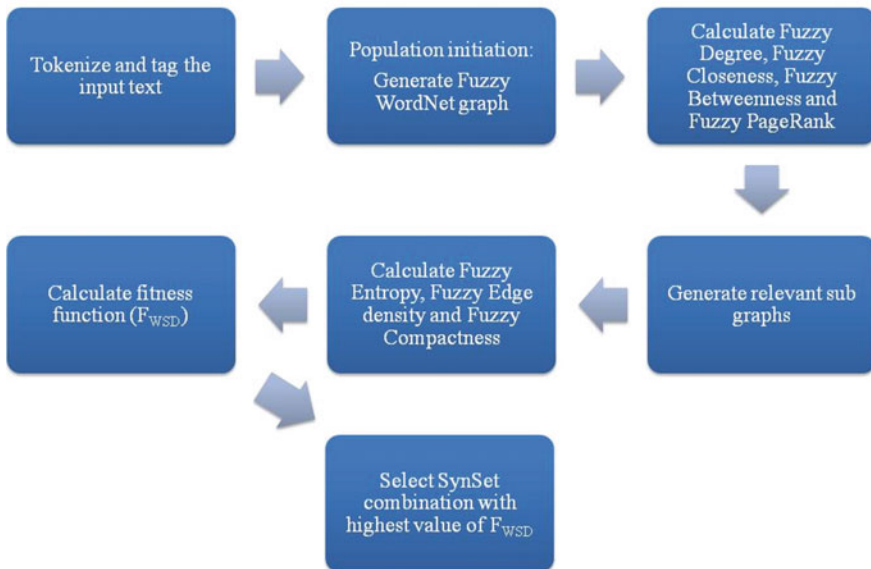


Fig. 2 Proposed work

c. Fuzzy Betweenness(N_{FB}):

$$N_{FB} = \frac{betweennessf^{(v)}}{(|V| - 1)(|V| - 2)} \tag{3}$$

d. Fuzzy PageRank(N_{FP}):

$$N_{FP} = \frac{(1 - d)}{|V|} + d \sum_{(v_a, v_b) \in E} \frac{\mu_{ba}}{\sum_{(v_b, v_c) \in E^{\mu_{bc}}} } PageRank_{f(v_b)} \tag{4}$$

Identify the node with the highest and second highest values of all these fuzzy local centrality measures to generate sub graphs for paths leading from one node to the other [3]. For the concerned sub-graphs, following fuzzy global measures needs to be calculated [2]:

i. Fuzzy Compactness(N_{FCO}):

$$N_{FCO} = \frac{2|V|^2|V - 1| - \sum_{u \in V} \sum_{v \in V} d(u, v)}{2|V|^2|V - 1| - |V||V - 1|} \tag{5}$$

ii. Fuzzy Entropy(N_{FEN}):

$$N_{FEN} = - \sum_{v \in V} \frac{\sum_{u \neq v} \mu_{uv}}{2|E|} \log \frac{\sum_{u \neq v} \mu_{uv}}{2|E|} \tag{6}$$

iii. Fuzzy Edge Density(N_{FED}):

$$N_{FED} = \frac{\sum_{u \in V} \sum_{v \in V} \mu_{uv}}{|V|C_2} \tag{7}$$

So as to define the proposed fitness function, a comparative study is performed for these global measures and the observations are tabulated as in Table 1.

On the basis of Table 1 and an intuitive logic, we propose the following significance order for these measures:

$$Entropy > Edge Density > Compactness$$

Since we have used Fuzzy WordNet graph hence the order in our case becomes:

$$Fuzzy Entropy > Fuzzy Edge Density > Fuzzy Compactness$$

Hence the fitness function (F_{WSD}) for the proposed approach is defined as follows:

$$F_{WSD} = \{(N_{FCO}) + (3 * N_{FEN}) + (2 * N_{FED})\}/3 \tag{8}$$

Table 1 Comparative study for global measures of graph connectivity

Parameters	Compactness	Graph entropy	Edge density
What the high values indicate	High values indicates that the vertices are connected with small distance i.e. Reach ability between vertices is good	High value indicates that vertices are almost equally relevant and significant	High value indicates great amount of connectivity
What the low values indicate	Low value indicates that the vertices are disconnected or connected with very large distance i.e. Insufficient links	Low value indicates that only a few of our vertices are significant	Low value indicates disconnected/less connected graph
Performance	Average performance with reference to WordNet(measured by the mean between precision and recall) [3]	Best performing global measure with reference to WordNet [3]	Average performance with reference to WordNet (measured by the mean between precision and recall) [3]

Calculate F_{WSD} for the concerned sub graphs. The SynSet combination with the highest value of F_{WSD} gives the most relevant sense of that word in the given context. In the state-of-art, if a node combination has higher values of any two global measures, it is considered for disambiguation. In the proposed approach, each fuzzy global measure is given a significance for consideration.

3 Results and Discussion

Let us consider the implementation of the above mentioned method using an example. Let the user query be as follows:

Query = “I like music and song”.

The text is tokenized and tagged as follows:

Tagged Words = [(‘Song’, ‘NN’), (‘Music’, ‘NN’)]

Where: NN → Noun

The words “Song” and “Music” are to be disambiguated. The Fuzzy WordNet graph using (‘Song’, ‘NN’) sand (‘Music’, ‘NN’) as the content words is shown as in Fig. 3. The total number of nodes in the Fuzzy WordNet graph is 20.

The results of the local fuzzy graph connectivity measures for this graph are as follows:

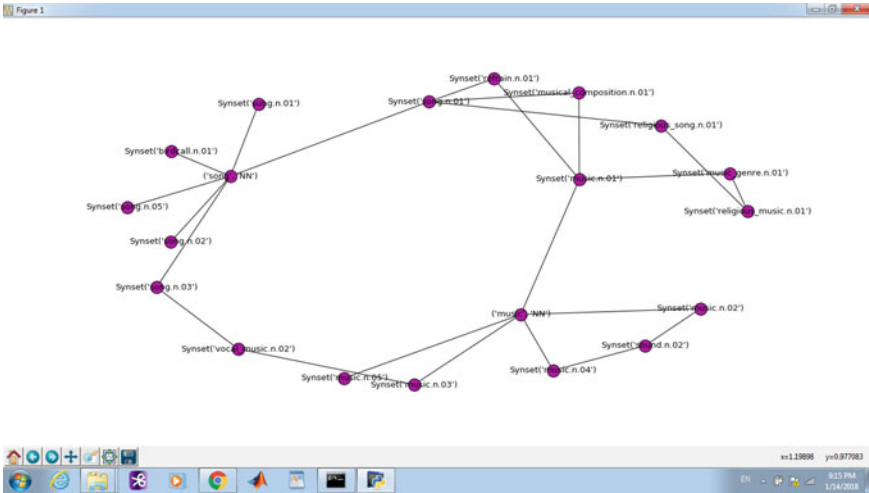


Fig. 3 Fuzzy WordNet graph for example text

- Fuzzy_Degree_Centrality** = {“Synset(‘Music.N.02’): 0.10526315789473684, “Synset(‘Refrain.N.01’): 0.10526315789473684, “Synset(‘Religious_Song.N.01’): 0.10526315789473684, “Synset(‘Music.N.01’): 0.21052631578947367, “Synset(‘Song.N.02’): 0.05263157894736842, “Synset(‘Song.N.03’): 0.10526315789473684, (“‘Music’, ‘NN’): 0.2631578947368421, “Synset(‘Music.N.03’): 0.10526315789473684, “Synset(‘Song.N.01’): 0.21052631578947367, “Synset(‘Music.N.05’): 0.05263157894736842, “Synset(‘Religious_Music.N.01’): 0.10526315789473684, “Synset(‘Musical_Composition.N.01’): 0.10526315789473684, “Synset(‘Sound.N.02’): 0.10526315789473684, “Synset(‘Vocal_Music.N.02’): 0.10526315789473684, “Synset(‘Music.N.04’): 0.10526315789473684, (“‘Song’, ‘NN’): 0.3157894736842105, “Synset(‘Birdcall.N.01’): 0.05263157894736842, “Synset(‘Song.N.05’): 0.05263157894736842, “Synset(‘Sung.N.01’): 0.05263157894736842, “Synset(‘Music_Genre.N.01’): 0.10526315789473684}
- Fuzzy_Betweenness_Centrality** = {“Synset(‘Music.N.02’): 0.049707602339181284, “Synset(‘Refrain.N.01’): 0.1111111111111111, “Synset(‘Religious_Song.N.01’): 0.06286549707602337, “Synset(‘Music.N.01’): 0.3674463937621832, “Synset(‘Music.N.05’): 0.0, “Synset(‘Song.N.03’): 0.13109161793372318, (“‘Music’, ‘NN’): 0.4327485380116959, “Synset(‘Music.N.03’): 0.13109161793372315, “Synset(‘Song.N.01’): 0.3674463937621832, “Synset(‘Song.N.02’): 0.0, “Synset(‘Religious_Music.N.01’): 0.03216374269005847, “Synset(‘Musical_Composition.N.01’): 0.1111111111111111, “Synset(‘Sound.N.02’): 0.0029239766081871343, “Synset(‘Vocal_Music.N.02’): 0.1198830409356725, “Synset(‘Music.N.04’): 0.0029239766081871343}

- 0.049707602339181284, (“Song’, ‘NN’): 0.4473684210526315,
 “Synset(‘Birdcall.N.01’): 0.0, “Synset(‘Song.N.05’): 0.0,
 “Synset(‘Sung.N.01’): 0.0, “Synset(‘Music_Genre.N.01’):
 0.06286549707602337}
- **Fuzzy_Closeness_Centrality** = {“Synset(‘Music.N.02’):
 0.27941176470588236, “Synset(‘Refrain.N.01’): 0.38,
 “Synset(‘Religious_Song.N.01’): 0.3064516129032258,
 “Synset(‘Music.N.01’): 0.3958333333333333, “Synset(‘Music.N.05’):
 0.2714285714285714, “Synset(‘Song.N.03’): 0.3392857142857143, (“‘Music’,
 ‘NN’): 0.36538461538461536, “Synset(‘Music.N.03’): 0.3392857142857143,
 “Synset(‘Song.N.01’): 0.3958333333333333, “Synset(‘Song.N.02’):
 0.2714285714285714, “Synset(‘Religious_Music.N.01’): 0.296875,
 “Synset(‘Musical_Composition.N.01’): 0.38, “Synset(‘Sound.N.02’):
 0.2261904761904762, “Synset(‘Vocal_Music.N.02’): 0.3275862068965517,
 “Synset(‘Music.N.04’): 0.27941176470588236, (“‘Song’, ‘NN’):
 0.36538461538461536, “Synset(‘Birdcall.N.01’): 0.2714285714285714,
 “Synset(‘Song.N.05’): 0.2714285714285714, “Synset(‘Sung.N.01’):
 0.2714285714285714, “Synset(‘Music_Genre.N.01’): 0.3064516129032258 }
 - **Fuzzy_PageRank_Centrality** = {“Synset(‘Music.N.02’):
 0.043483351280936805, “Synset(‘Refrain.N.01’):
 0.04083724267179972, “Synset(‘Religious_Song.N.01’):
 0.04304701702576124, “Synset(‘Music.N.01’): 0.07740664387255262,
 “Synset(‘Song.N.02’): 0.026465513560724324, “Synset(‘Song.N.03’):
 0.04589236891688578, (“‘Music’, ‘NN’): 0.10051711972448235,
 “Synset(‘Music.N.03’): 0.04401440063883744, “Synset(‘Song.N.01’):
 0.07947105414723951, “Synset(‘Music.N.05’): 0.02458754528267599,
 “Synset(‘Religious_Music.N.01’): 0.04390279032779826,
 “Synset(‘Musical_Composition.N.01’): 0.04083724267179972,
 “Synset(‘Sound.N.02’): 0.04446157807566113,
 “Synset(‘Vocal_Music.N.02’): 0.045711108878297706,
 “Synset(‘Music.N.04’): 0.0434833512809368, (“‘Song’, ‘NN’):
 0.13387680165992172, “Synset(‘Birdcall.N.01’): 0.026465513560724324,
 “Synset(‘Song.N.05’): 0.026465513560724324, “Synset(‘Sung.N.01’):
 0.026465513560724324, “Synset(‘Music_Genre.N.01’):
 0.0426083293015159}.

The SynSets with the highest and second highest values for these measures are Music.N.02, Music.N.01, Music.N.03, Song.N.03, Music.N.04 and Song.N.01. Sub graphs are generated for their combination.

The results for the fitness function for the sub graphs are tabulated as in Table 2.

It can be seen that “Music.N.01-Song.N.01” yields the highest value of fitness function hence it would be the best suited combination. This means that the disambiguated senses for Music and Song in the given context are Music.N.01 and Song.N.01.

Table 2 Fitness Function for the relevant sub-graphs

S. No.	SynSet combination	N_{FCO}	N_{FEN}	N_{FED}	F_{WSD}
1	Music.N.02-Song.N.03	0.90	0.63	0.24	1.09
2	Music.N.01-Song.N.03	0.91	0.75	0.24	1.21
3	Music.N.03-Song.N.03	0.90	0.71	0.26	1.18
4	Music.N.04-Song.N.03	0.92	0.64	0.24	1.12
5	Music.N.02-Song.N.01	0.90	0.69	0.29	1.18
6	Music.N.01-Song.N.01	0.98	0.88	0.38	1.46
7	Music.N.03-Song.N.01	0.93	0.68	0.29	1.18
8	Music.N.04-Song.N.01	0.94	0.59	0.28	1.09

So as to test the feasibility of the proposed approach, the algorithm was applied to a synthetic dataset consisting of 20 ambiguous words. Senses were obtained using it and the results were compared with the senses provided by the human annotators. It was observed that 13 out of 20 senses were found to be same as the ones provided by the human annotators i.e. the results show an accuracy of 65%.

4 Conclusion

This paper depicts a novel genetic algorithm based approach for performing WSD using Fuzzy WordNet graph. The fitness function is calculated using fuzzy global measures of graph connectivity. For proposing the fitness function, a comparative study is performed for the global measures edge density, entropy and compactness to assign them significance. An analytical analysis for WSD is done using research papers from Web of Science. In the future, the proposed approach can be extended by using other semantic relations like troponymy.

References

1. Vij, S., Jain, A., Tayal, D., Castillo, O.: Fuzzy logic for inculcating significance of semantic relations in word sense disambiguation using a WordNet graph. *Int. J. Fuzzy Syst.* (2017). (Springer) <https://doi.org/10.1007/s40815-017-0433-8>
2. Jain, A., Lobiyal, D.K.: Fuzzy Hindi WordNet and word sense disambiguation using fuzzy graph connectivity measures. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* **15**(2), 8 (2016)
3. Navigli, R., Lapata, M.: An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(4), 678–692 (2010)
4. Hung, J.C., Wang, C.S., Yang, C.Y., Chiu, M.S., Yee, G.: Applying word sense disambiguation to question answering system for e-learning. In: 19th International Conference on Advanced Information Networking and Applications, AINA 2005. vol. 1, pp. 157–162. IEEE, March 2005

5. Duque, A., Martinez-Romo, J., Araujo, L.: Can multilinguality improve biomedical word sense disambiguation? *J. Biomed. Inf.* **64**, 320 (2016). (Academic Press Inc Elsevier Science) <https://doi.org/10.1016/j.jbi.2016.10.020>
6. Ellenius, J., Bergvall, T., Dasgupta, N., Hedfors, S., Pierce, C., Noren, G.N.: Medication name entity recognition in tweets using global dictionary lookup and word sense disambiguation. *Pharmacoepidemiol. Drug Safety* **25**, 414 (2016) (Wiley-Blackwell)
7. Carpuat, M., Wu, D.: Improving statistical machine translation using word sense disambiguation. In: *Emnlp-Conll*, vol. 7, pp. 61–72, June 2007
8. Jain, A., Mittal, K., Tayal, D.K.: Automatically incorporating context meaning for query expansion using graph connectivity measures. *Prog. Artif. Intell.* **2**(2–3), 129–139 (2014)
9. Moon, S., McInnes, B., Melton, G.B.: Challenges and practical approaches with word sense disambiguation of acronyms and abbreviations in the clinical domain. *Healthcare Inform. Res.* **21**(1), 35 (2015). (Korean Soc Medical Informatics) <https://doi.org/10.4258/hir.2015.21.1.35>
10. Chasin, R., Rumshisky, A., Uzuner, O., Szolovits, P.: Word sense disambiguation in the clinical domain: a comparison of knowledge-rich and knowledge-poor unsupervised methods. *J. Am. Med. Inform. Assoc.* **21**(5), 842 (2014). (BMJ Publishing Group) <https://doi.org/10.1136/amiajnl-2013-002133>
11. Chen, Y.K., Cao, H.X., Mei, Q.Z., Zheng, K., Xu, H.: Applying active learning to supervised word sense disambiguation in MEDLINE. *J. Am. Med. Inform. Assoc.* **20**(5), 1001 (2013). (BMJ Publishing Group) <https://doi.org/10.1136/amiajnl-2012-001244>
12. Alsaeedan, W., Menai, M.E., Al-Ahmadi, S.: A hybrid genetic-ant colony optimization algorithm for the word sense disambiguation problem. *Inf. Sci.* **417**, 20 (2017). (Elsevier Science Inc) <https://doi.org/10.1016/j.ins.2017.07.002>
13. Abualhaija, S., Zimmermann, K.H.: D-Bees: a novel method inspired by bee colony optimization for solving word sense disambiguation. *Swarm Evol. Comput.* **27**, 188 (2016). (Elsevier Science Bv) <https://doi.org/10.1016/j.swevo.2015.12.002>
14. Abed, S.A., Tiun, S., Omar, N.: Word sense disambiguation in evolutionary manner. *Connection Sci.* **28**(3), 226 (2016). (Taylor & Francis Ltd) <https://doi.org/10.1080/09540091.2016.1141874>
15. AlSaidi, B.K.: Automatic approach for word sense disambiguation using genetic algorithms. *Int. J. Adv. Comput. Sci. Appl.* **7**(1), 41 (2016) (Science & Information Sai Organization Ltd)
16. Abed, S.A., Tiun, S., Omar, N.: Harmony search algorithm for word sense disambiguation. *PLOS ONE* **10**(9) (2015). (Public Library Science) <https://doi.org/10.1371/journal.pone.0136614>
17. Menai, M.E.B.: Word sense disambiguation using evolutionary algorithms—application to Arabic language. *Comput. Human Behav.* **41**, 92 (2014). (Pergamon-Elsevier Science Ltd) <https://doi.org/10.1016/j.chb.2014.06.021>

Configuration Module for Treating Design Anomalies in Databases for a Natural Language Interface to Databases



Grigori Sidorov, Rodolfo A. Pazos R., José A. Martínez F., J. Martín Carpio and Alan G. Aguirre L.

Abstract Natural language interfaces for databases (NLIDBs) are tools that allow obtaining information from a database (DB) through natural language queries. Currently, domain-independent NLIDBs (interfaces capable of working with several DBs) have not had a favorable performance due to the complexity for devising an approach that can solve all the problems that occur in NLIDBs. Another important problem is the existence of design anomalies in DBs, which have not been addressed in the approaches proposed by various authors. This chapter describes the implementation of a module for dealing with four design anomalies: absence of primary and foreign keys, use of surrogate keys, Columns for storing aggregate function calculations, and columns repeated in two or more tables. This module was implemented in an experimental NLIDB. The purpose of this is to allow the NLIDB to be configured so that it perceives an anomalous DB as a DB without design anomalies so the NLIDB can function correctly. Experimental tests of the NLIDB with the module for treating design anomalies show that it can correctly answer 100% of the test queries; whereas, the NLIDB without the module only generates 25% correct answers.

Keywords Database anomaly · Database schema · Natural language interfaces to databases · Relational database

G. Sidorov

Centro de Investigación en Computación, Instituto Politécnico Nacional, Mexico City, Mexico
e-mail: sidorov@cic.ipn.mx

R. A. Pazos R. (✉) · J. A. Martínez F. · A. G. Aguirre L.

Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Madero, Mexico
e-mail: r_pazos_r@yahoo.com.mx

J. A. Martínez F.

e-mail: jose.mtz@gmail.com

A. G. Aguirre L.

e-mail: li.aguirre.lam@hotmail.com

J. M. Carpio

Tecnológico Nacional de México, Instituto Tecnológico de León, León, Mexico
e-mail: jmcarpio61@hotmail.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_48

1 Introduction

Relational databases are used in businesses for storing important information. There exist many software tools that facilitate obtaining information from DBs. Among the easiest to use by inexperienced users are NLIDBs. These interfaces allow users to formulate queries in their own language [1], facilitating access to information to those who have no knowledge of a formal language for querying databases or programming.

There are two types of NLIDBs according to the number of DBs that can be queried using the interface: domain-specific interfaces and domain-independent interfaces. The first ones can be used for querying only one DB, the one for which they were designed. The second ones can be used for querying any DB.

In the development of NLIDBs, it has not been considered what behavior they would have if they were used with DBs with design anomalies [2]. Regarding this problem, no publication on NLIDBs mentions having addressed this issue until now.

Domain-specific NLIDBs have been developed to adapt to the design anomalies that their DBs have. In contrast, domain-independent NLIDBs are not designed in the same way, so they have a poor performance when they are used with DBs with design anomalies.

The survey on NLIDBs presented in [2] shows that some of the most recent [3–6] and most popular interfaces [7–9] do not consider the problems related to design anomalies in DBs.

Nowadays, there are many DBs that have design anomalies [10, 11]. Therefore, many of the domain-independent NLIDBs do not perform correctly with a large number of DBs that have this problem.

In this chapter we present a configuration module for a domain-independent NLIDB prototype [12], which allows it to perceive an anomalous DB as if it had no design anomalies. Before explaining the characteristics of the configuration module developed, it is important to mention some aspects of the interface used in this project.

One of the main components of the NLIDB is the Semantic Information Dictionary (SID) [12], which stores DB metadata and relates it to the semantic information of the DB domain. The SID is configured by the database administrator (DBA). To configure the dictionary, the DBA uses the Domain Editor of the interface, which is a graphical interface that allows the administrator to visualize the information stored in the SID. However, the configuration that can be performed using the Domain Editor does not allow the NLIDB to correctly answer natural language questions when querying DBs with design anomalies.

When the NLIDB is used for querying a new DB, it performs an automatic configuration of the SID. However, this configuration includes the design anomalies of the original DB schema; this degrades the performance of the interface when translating a natural language query to its equivalent in SQL.

2 Database Design Anomalies

Design anomalies in DBs make the design of the database inconsistent with good DB design practices.

Most of the time, DB anomalies are introduced in the design of a database because the applications that use it require them.

The NLIDB [12], for which the module was implemented for treating design anomalies, is a domain-independent interface; therefore, it is important that it can work correctly with any DB, which implies that it should be able to work with DBs with design anomalies.

The design anomalies considered in this chapter are the following:

- **Absence of primary and foreign keys.** This problem prevents the NLIDB from performing joins between tables when a query involves several tables, since the NLIDB uses the foreign keys for that task.
- **Use of surrogate keys instead of primary keys.** The use of substitute primary keys can cause duplicate or inconsistent information, because this allows inserting several identical rows in a table, whose only difference is the value of the substitute primary key.
- **Columns for storing aggregate function calculations.** The value of these columns can be calculated by means of an aggregate function applied to a column of another table. However, this problem requires the DB administrator to decide whether the aggregate function should be applied or to use the value of the column with such anomaly.
- **Repeated columns in two or more tables.** These columns are not foreign keys nor primary keys; in addition, information regarding these columns should only be stored in one table. Therefore, it can cause the NLIDB to obtain inconsistent information.

3 Configuration Module

The configuration module implemented in the NLIDB allows it to perceive the DB as if it had no design anomalies and thus function correctly (Fig. 1).

For this purpose, the configuration module was implemented in such a way that the DBA can treat the design anomalies present in the DB schema one by one. The module has a section for each design anomaly, and in each section it guides the user through the SID configuration process.

It is important to mention that the configuration module presented in this chapter does not modify the structure of the DB in use; instead, it modifies the configuration of the SID so that it represents a DB schema without design anomalies, so the NLIDB can have a perception of the DB according to this schema.

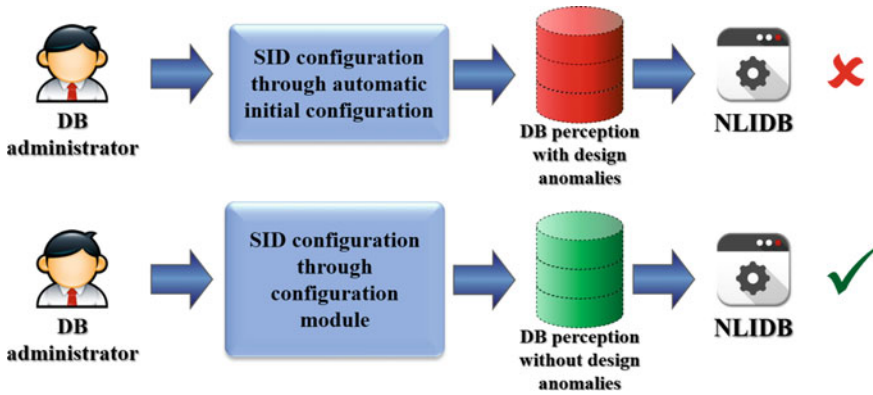


Fig. 1 Perception of the DB through different configurations

3.1 Treating the Absence of Primary and Foreign Keys

To deal with the absence of primary and foreign keys, a graphic interface was designed that allows the DBA to define primary and foreign keys in the SID (Fig. 2).

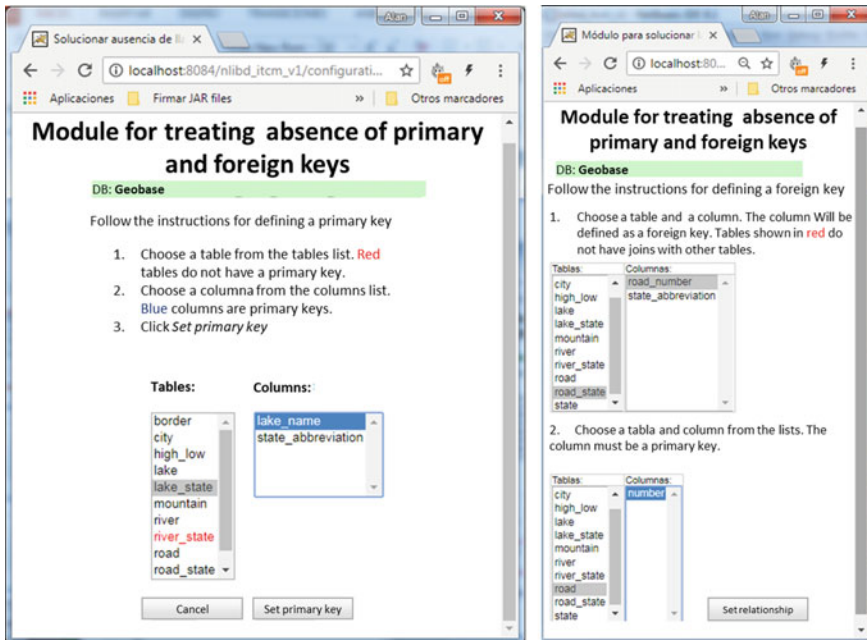


Fig. 2 Module for treating the absence of primary and foreign keys

This section of the module consists of two parts. In the first, the DBA can define a column of a table as a primary key in the SID (Fig. 2, left side). In the second part (Fig. 2, right side) the administrator can define a relationship between two tables by selecting the table that would be the foreign key and then selecting the primary key that would be related to the column selected previously.

After carrying out this process, the NLIDB will be able to use the relationships defined to create the joins that are required for the translation of a query.

3.2 *Treating the Use of Surrogate Keys*

The treatment of substitute primary keys consists of identifying the surrogate key and determining if it has any relationship with another table through a foreign key. It is important that the DBA consider the following situations:

- If a table has a surrogate key (T_{SK}), this table must have a natural primary key (candidate key).
- If table T_{SK} has a natural primary key, then it is necessary to identify if it is related to other tables through foreign keys.
- Tables related to table T_{SK} must have a set of columns that may constitute a foreign key that is related to T_{SK} 's natural primary key.

Once the preceding situations have been identified, the DBA should use the configuration module for dealing with this design anomaly. To this end, the administrator must first define in the SID the column that is a surrogate key using an interface similar to the one shown in Fig. 2.

Subsequently, the DBA must define the relationships between T_{SK} and other tables. To do this, the administrator must use the interface shown in Fig. 3, where the tables that have a surrogate key are shown, and he must select table T_{SK} ; from this table, a column that is a natural primary key of the table must be selected. Subsequently, the tables that could be related to T_{SK} are shown and the DBA must select the column that is the foreign key that is related to the column selected at the beginning (natural primary key of T_{SK}). Optionally, the administrator may indicate if he wishes that the NLIDB use the relationship defined in the translation process.

3.3 *Treating Columns for Storing Aggregate Function Calculations*

The treatment for this design anomaly consists of identifying the column that has this problem and defining which aggregate function should be used instead of the column.

The DBA can use the interface shown in Fig. 4, which shows a list of all the tables in the DB and its columns. From this list, the column with the anomaly should be

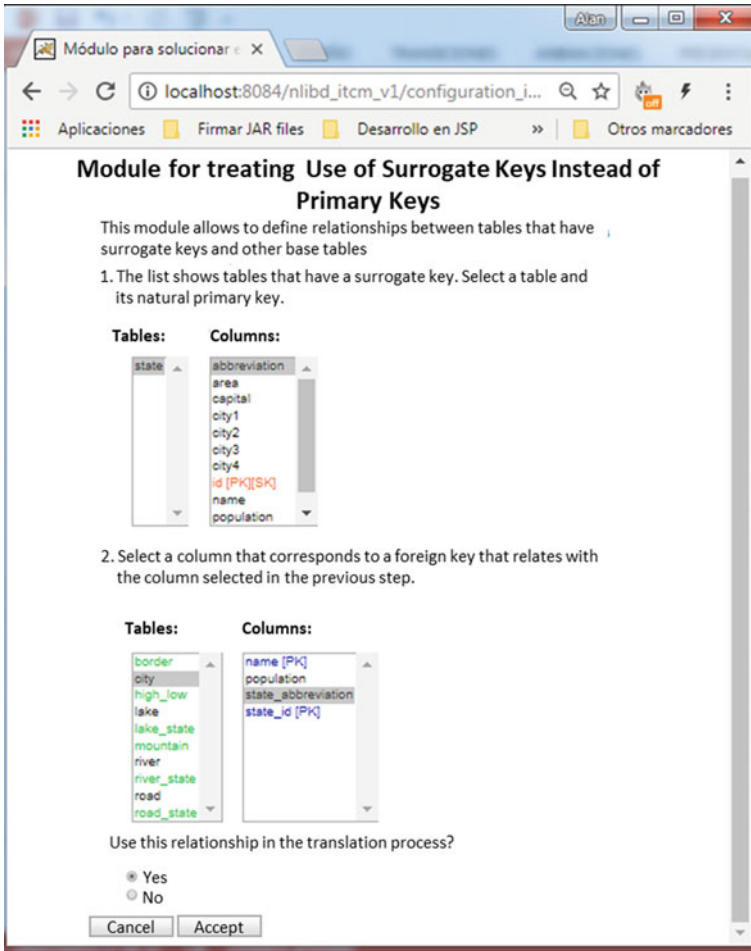


Fig. 3 Module for treating the use of surrogate keys

selected. Subsequently, the aggregate function must be indicated and the column to which the aggregate function will be applied.

3.4 Treating Repeated Columns in Two or More Tables

Most of the time, columns that have this anomaly might contain incorrect information due to information insertion errors. Therefore, when the NLIDB retrieves this information, it will be wrong.

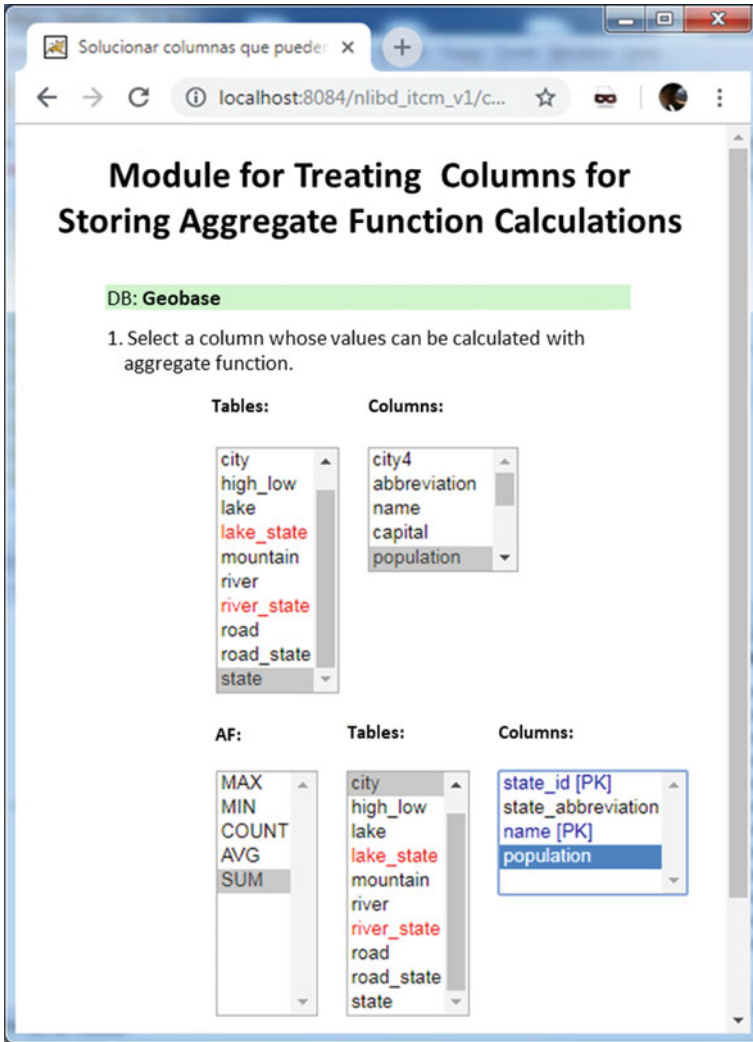


Fig. 4 Module for treating columns for storing aggregate function calculations

To solve the problems related to this anomaly, the DBA must first indicate the column that stores the correct information (usually a column that corresponds to an attribute of a *strong entity*). Then, the DBA must specify the columns that are replicas of the column selected in the first step (Fig. 5).

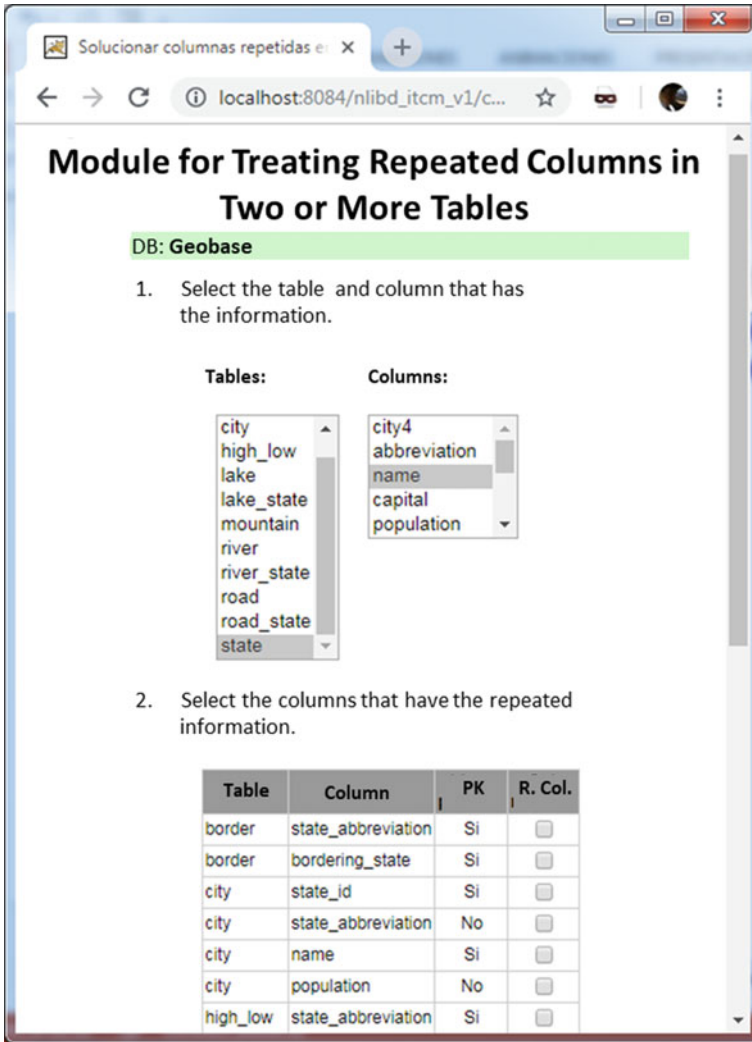


Fig. 5 Module for treating repeated columns

4 Experimental Results

The main objective of this experiment is to prove that the configuration of the SID performed using the configuration module allows the NLIDB to correctly answer natural language (NL) queries for DBs with design anomalies.

The tests carried out consisted of introducing 20 anomalies in the ATIS and Geobase DBs schemas [7]. The 20 anomalies were separated into groups of 5 anomalies, giving a total of 4 groups of 5 anomalies each. Each group corresponds to a type

of anomaly (absence of foreign keys, use of surrogate keys, etc.). For each anomaly, a NL query of the ATIS or Geobase corpus was chosen [7]. These queries involve the tables where the design anomalies were introduced.

For this experiment it was necessary to create new DB schemas that include the anomalies. Subsequently, the automatic configuration was performed by the NLIDB. The SID configuration was then carried out using the modules mentioned in the previous sections for treating the design anomalies. Finally, the 20 queries of the corpus were input to the NLIDB configured to treat anomalies and also to the NLIDB without treatment for anomalies, giving the results shown in Table 1.

In Table 1, for queries corresponding to the absence of foreign keys, the new version of the NLIDB was able to correctly answer 5 queries (second column of Table 1). This is because the absence/presence of foreign keys affects the join construction process carried out by the NLIDB; therefore, for any query that involves several DB tables that do not have foreign keys, the previous version of the interface cannot generate the joins between the tables (third column of Table 1).

For example, the NL query *Give me the one-way fare for flight number 18* involves tables *flight*, *flight_fare* and *fare* of the ATIS database. The design anomaly for this query consisted in eliminating the integrity constraint defined in *flight_fare.flight_code* (FK) and *flight.flight_code* (PK). The results for this example where the following:

Query 1. SQL translation without design anomaly treatment

```

SELECT fare.one_way_cost
FROM fare, flight, restriction, restrict_carrier, airline
WHERE fare.restrict_code = restriction.restrict_code AND
        restriction.restrict_code = restrict_carrier.restrict_code AND
        restrict_carrier.airline_code = airline.airline_name AND
        airline.airline_code = flight.airline_code AND
        flight.flight_number = 18
    
```

Table 1 Experimental results for NLIDB configured for treating DB design anomalies

DB design anomaly	Correct translations w/o DA treatment	Correct translations w/o DA treatment
Absence of primary and foreign keys	5	0
Use of surrogate keys	5	0
Columns for storing aggregate function calculations	5	5
Repeated columns in two or more tables	5	0
Total	20	5

Query 2. SQL translation with design anomaly treatment

```

SELECT fare.one_way_cost
FROM fare, flight, flight_fare
WHERE fare.fare_code = flight_fare.fare_code AND
        flight_fare.flight_code = flight.flight_code AND
        flight.flight_number = 18

```

As can be observed in Query 1, since there are no foreign keys in *flight_fare.flight_code*, the NLIDB cannot build the joins between *flight* and *flight_fare*; therefore, the interface looks for a set of tables that are related by foreign keys that connect *flight* and *fare* tables, finding the set *flight*, *airline*, *restrict_carrier*, *restriction*, *fare*. Therefore, the interface proceeds to create joins between them, giving an erroneous result.

In the queries that involve the use of surrogate keys, an error occurs in the translation of these queries, because, for this test a surrogate key and foreign keys that refer to it were introduced. Therefore, when obtaining information about this segment of the DB, the NLIDB returns a set of repeated values.

For example, the NL query *List the fare for flight number 19* involves tables *flight*, *flight_fare* and *fare*. For this test, a surrogate key called *flight.flight_id* was created, which is a numeric column, also a column called *flight_fare.flight_id* which refers to the said column was added. Additionally, several identical rows were inserted in the *flight* table with different values in the surrogate key of the table. The SQL translation generated using the NLIDB without treatment of anomalies, 30 rows are obtained with repeated information 6 times, while the SQL translation generated by the NLIDB with treatment of anomalies obtained 5 rows, which correspond to the 5 different fares of flight number 19.

Queries involving columns that can be calculated with aggregate functions were considered correct for both NLIDBs, since the DBA must decide whether to use an aggregate function instead of the column with the anomaly or to use this column. This is because the information stored in this type of columns is sometimes convenient to use as it is in the DB.

In queries involving repeated columns in two or more tables, columns containing information that is stored in another table were inserted, however, the values of these columns do not coincide with the information mentioned. Therefore, in the NLIDB without treatment of anomalies, it generated an incorrect SQL translation.

Consider for example, the query *Give me the cities in the state of Alaska*. In the table *city*, the *city.state_name* column was added; Additionally, in this table the cities of a state can be identified by its abbreviation (*city.state_abbreviation*). By adding a column with the name of the state (*city.state_name*), the NLIDB can use that column to relate it to the search value *Alaska*. However, this may result in erroneous information, since the column that should relate to *Alaska* is in the column *state.name*, and the column *city.state_name* might not have the correct name of the state.

5 Final Remarks

In this chapter a configuration module for treating design anomalies in databases for a NLIDB was presented. Specifically, this module can be used for configuring the Semantic Information Dictionary (SID) of the interface, which is one of the main components of the domain-independent NLIDB [12] used for this project. The SID stores DB metadata (DB schema) and relates it to the semantic content of the DB domain.

NLIDBs have not been designed for querying DBs with design anomalies, which affect the performance of the interfaces, causing them to function incorrectly. Therefore, many of the existing interfaces would not function correctly when consulting DBs that have such anomalies.

In [2], some design anomalies are described. In this chapter, four design anomalies were considered: absence of primary and foreign keys, use of surrogate keys, columns for storing aggregate function calculations, and columns repeated in two or more tables.

To address the aforementioned design anomalies, a configuration module is presented for the SID, which provides the NLIDB with a perception of the DB without design anomalies.

The configuration module does not make modifications to the DB schema, instead, it modifies the configuration of the SID in such a way that the design anomalies of the DB are not present in the configuration stored in the SID.

The configuration module guides the DBA to configure the SID. To this end, the module presents the administrator with an interface that shows the names of the anomalies separately. The module was implemented in this way so that the DBA could identify the anomalies and treat them one by one. In addition, this design prevents the administrator from editing sensitive SID information.

The tests carried out for the NLIDB with the module for the treatment of anomalies, show that, when querying a DB with design anomalies, the interface can answer correctly 100% of the queries. On the contrary, the performance of the previous version of the NLIDB without anomalies treatment is affected by the design anomalies; specifically, it can only answer correctly 25% of the queries.

As future work, an extension to the configuration module is planned. This extension will consist of an interface for configuring the SID for treating the nonconformity of a DB with the second and third normal forms.

References

1. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural language interface to database: an introduction. *J. Nat. Lang. Eng.* 29–81 (1995)
2. Pazos, R.A., Martínez, J.A., Aguirre, A.G., Aguirre, M.A.: Issues in querying databases with design anomalies using natural language interfaces, fuzzy logic augmentation of neural and

- optimization algorithms: theoretical aspects and real applications. *Stud. Comput. Intell.* **749** (2018)
3. O'Shea, J.D., Shabaz, K., Crockett, K.A.: Aneesah: a conversational natural language interface to databases. In: *Proceedings of World Congress on Engineering 2015*, vol. 1, London (2015)
 4. Amsterdamer, Y., Kukliansky, A., Milo, T.: A natural language interface for querying general and individual knowledge. In: *Proceedings of the VLDB Endowment*, vol. 8, no. 12 (2015)
 5. ElSayed, K.: An arabic natural language interface system for a database of the Holy Quran. *Int. J. Adv. Res. Artif. Intell. (IJARAI)* **4**(7) (2015)
 6. Kataria, A., Nath, R.: Natural language interface for databases in hindi based on karaka theory. *Int. J. Comput. Appl.* **122**(7) (2015) (India)
 7. Minock, M.: C-phrase: a system for building robust natural language interfaces to databases. *Data Knowl. Eng.* **69**, 290–302 (2010)
 8. Conlon, S., Conlon, J., James, T.: The economics of natural language interfaces: natural language processing technology as a scarce resource. *Decis. Support Syst.* **38**(1), 141–159 (2004)
 9. Giordani, A., Moschitti, A., Translating queries with generative parsers discriminatively reranked. In: *Proceedings of Computational Linguistics*, pp. 401–410. Mumbai (2012)
 10. Pedro de Jesus, M.L., Sousa, P.M.A.: Selection of reverse engineering methods for relational databases. In: *Proceedings of the European Conference on Software Maintenance and Reengineering* (1999)
 11. Mfourga, N.: Extracting entity-relationship schemas from relational databases: a form-driven approach. In: *Proceedings of the Fourth Working Conference on Reverse Engineering*. IEEE (1997)
 12. Pazos, R.A., González, J.J., Aguirre, M.A.: Semantic model for improving the performance of natural language interfaces to databases. In: *Advances in Artificial Intelligence. MICAI 2011. Lecture Notes in Computer Science*, vol. 7094 (2011)

Development of a Virtual View for Processing Complex Natural Language Queries



José A. Martínez F., Rodolfo A. Pazos R., Héctor Puga and Juana Gaspar H.

Abstract Natural language interfaces to databases (NLIDBs) allow inexperienced users to formulate questions in natural language (NL) to DB management systems. A survey of the literature on NLIDBs shows that the prevalence of the interfaces developed to date are limited by two aspects: it is difficult to port them from one database (DB) to another, and they fail on some questions, either because they misinterpret the questions or they are unable to respond even when the answer exists. Among the problems that occur in NL questions, there is a problem that may arise when querying complex DBs: *semantically implied entities*. This problem is related to the semantic meaning of a query, when upon referring to an entity (DB table), another entity (or entities) is semantically implied to which the first entity is related, and usually the user ignores the relationship between the two (or more) entities. This chapter describes a method for using a virtual view in a NLIDB, which allows processing queries that involve this problem more efficiently than a previous version of the interface that uses materialized views. Specifically, experimental results show that the new version reduces by 23% the time for translating questions from NL to SQL.

Keywords Natural language interface to databases · Complex queries · Semantic view · Virtual view

J. A. Martínez F. · R. A. Pazos R. (✉) · J. Gaspar H.
Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Madero, Madero, Mexico
e-mail: r_pazos_r@yahoo.com.mx

J. A. Martínez F.
e-mail: jose.mtz@gmail.com

J. Gaspar H.
e-mail: jgasparhdz@gmail.com

H. Puga
Tecnológico Nacional de México, Instituto Tecnológico de León, León, Mexico
e-mail: pugahector@yahoo.com

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_49

1 Introduction

Currently, society needs to access more information in a fast and reliable way, which requires new ways for retrieving information. Natural language interfaces to databases (NLIDBs) allow users to formulate queries in their own language, facilitating access to information without knowledge of a query language to databases or programming [1].

To date, there is a large number of NLIDBs, and new interfaces are still being developed recently. Some of the recently developed interfaces are: AskMe [2], SNL2SQL [3], NaLiR [4], Aneesah [5], NL2CM [6], NLI Arab Holy Quran [7], NLI Hindi [8]. Unfortunately, none of these interfaces reports results on querying complex databases (DBs) such as ATIS. It is for this reason that it is very difficult to determine if these interfaces can correctly answer complex queries, such as those of the ATIS corpus.

The NLIDB prototype [9] developed at the Instituto Tecnológico de Ciudad Madero (ITCM) has a mechanism to answer complex queries. This mechanism is called *semantic view*, which is an SQL view to which semantic information is added. Additionally, the definition of the semantic view is stored in the Semantic Information Dictionary (SID) [9]. Unfortunately, this interface uses materialized views, which consume many resources, making the interface inefficient when querying large DBs.

In order to improve the performance of the NLIDB, this chapter describes a method that allows using virtual views when processing queries involving semantic views. This method allows transforming a query that involves a view to a query that only involves base tables of the DB. This modification improves the response time of the interface by reducing the use of resources by the database management system (DBMS).

2 Complex Queries

There is a problem that may be present when querying complex DBs. This problem is related to the semantic meaning of a query, when referring to an entity (DB table), another entity (or entities) is semantically involved with the first entity and generally the user ignores the relationship between the two (or more) entities. In this chapter we will refer to this difficulty as the *problem of semantically implied entities*.

To illustrate this problem, consider two queries. The first one is:

When do the flights that leave from DFW arrive in LGA?

Where *LGA* and *DFW* are airport codes. The translation is as follows:

```
SELECT  flight.arrival_time
FROM    flight
WHERE   flight.from_airport LIKE 'DFW' AND
         flight.to_airport LIKE 'LGA'
```

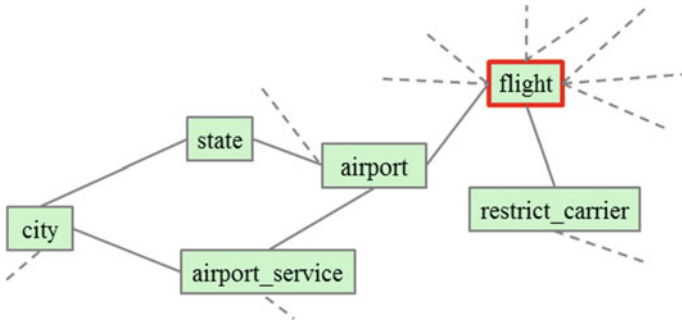


Fig. 1 Segment of the ATIS database

This statement is very simple, since table *flight* contains all the data required to answer the query (Fig. 1).

However, by slightly changing the previous query, it becomes very complex, as the following example shows:

When do the flights that leave from Fort Worth arrive in New York?

Where *New York* and *Fort Worth* are city names. The complexity of this lies in the fact that table *flight* does not contain city names; these are stored in table *city* (Fig. 1). In fact, flights do not leave from a city and arrive in another, but they leave and arrive to airports in or near cities.

Figure 2 shows that this situation, involving flights and cities, is modeled in the DB schema using two intermediate tables (*airport* and *airport_service*) that are related to tables *flight* and *city*. The SQL statement for answering this query is very complex and is expressed as follows:

```
SELECT flight.arrival_time
FROM city AS OC, airport_service AS OAS,
airport AS OA, flight AS FLIGHT,
airport AS DA, airport_service AS DAS,
```

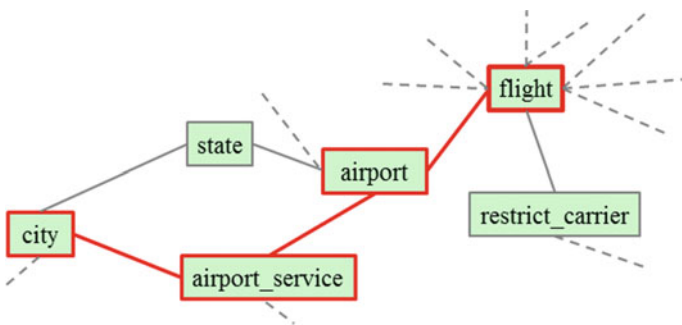


Fig. 2 Tables involved in a query that has a problem of *semantically implied entities*

city AS DC
WHERE *OC.city_name* LIKE 'Fort Worth' AND
DC.city_name LIKE 'New York' AND
OC.city_code = *OAS.city_code* AND
OAS.airport_code = *OA.airport_code* AND
OA.airport_code = *FLIGHT.from_airport* AND
FLIGHT.to_airport = *DA.airport_code* AND
DA.airport_code = *DAS.airport_code* AND
DAS.city_code = *DC.city_code*

3 Processing Complex Queries in the NLIDB

To answer complex queries like the one described, the NLIDB uses a semantic view to find the information of implicit relationships between entities. An example of a semantic view is shown in Fig. 3.

The process performed by the previous version of the NLIDB to translate a complex query to its equivalent in SQL is explained below.

First, the interface executes a lexical analysis to determine the syntactic category of the lexical units (tokens) of the NL query. With this information, it identifies the tables, columns and search values to which some lexical units refer. Subsequently, it associates each identified column with a clause (Select, From or Where).

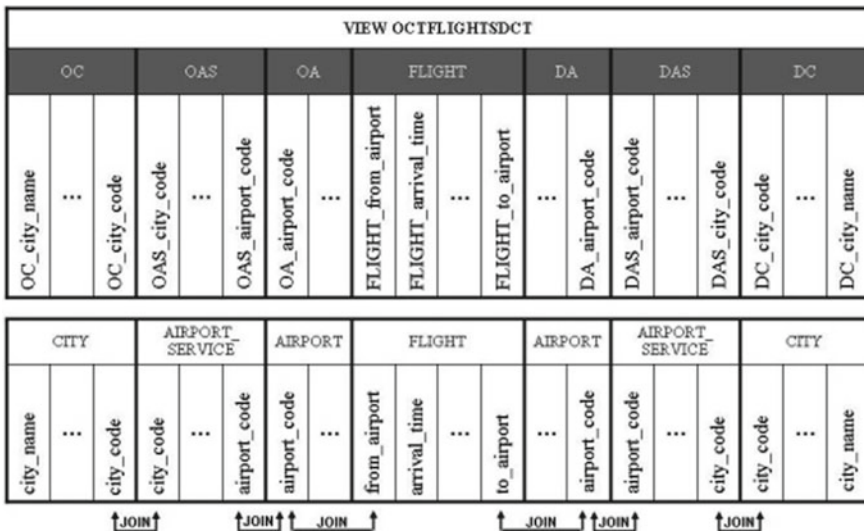


Fig. 3 Conceptual schema of a semantic view

At this point, the NLIDB determines whether the query requires the use of a semantic view for translating the query. To this end, it performs the comparison of each search value with a pattern associated to the semantic view. In case the view pattern and the search value coincide, the interface proceeds to generate a materialized view using the constituent tables and columns defined for it in the SID. An example of the SQL expression used to generate the aforementioned view is shown in Statement 1.

Statement 1

```

CREATE VIEW  octflightsdct AS
(SELECT      OC.city_code AS OC_city_code,
              OC.city_name AS OC_city_name,
              ...
              FLIGHT.flight_code AS FLIGHT_flight_code,
              FLIGHT.arrival_time AS FLIGHT_arrival_time,
              ...
FROM        city AS OC,
              airport_service AS OAS,
              airport AS OA,
              flight AS FLIGHT,
              airport AS DA,
              airport_service AS DAS,
              city AS DC
WHERE       OC.city_code = OAS.city_code AND
              OA.airport_code = OAS.airport_code AND
              FLIGHT.from_airport = OA.airport_code AND
              FLIGHT.to_airport = DA.airport_code AND
              DA.airport_code = DAS.airport_code AND
              DC.city_code = DAS.city_code)

```

Once the view is created, the NLIDB proceeds to generate a semantic graph to determine the implicit joins among the view and the tables involved in the query. To this end, the DB schema stored in the SID is considered and a semantic graph is generated with all the tables involved. However, when considering the view, the tables involved in it are removed from the semantic graph and the generated view is added, relating the view to the remaining base tables (Fig. 4).

After creating the semantic graph, the NLIDB generates an SQL query with the information obtained so far. This query involves the semantic view and, if necessary, some other base tables. For example, the SQL translation for the query *When do the flights that leave from Fort Worth arrive in New York?* is the one described in Statement 2.

Statement 2

```

SELECT  octflightsdct.FLIGHT_arrival_time
FROM    octflightsdct

```

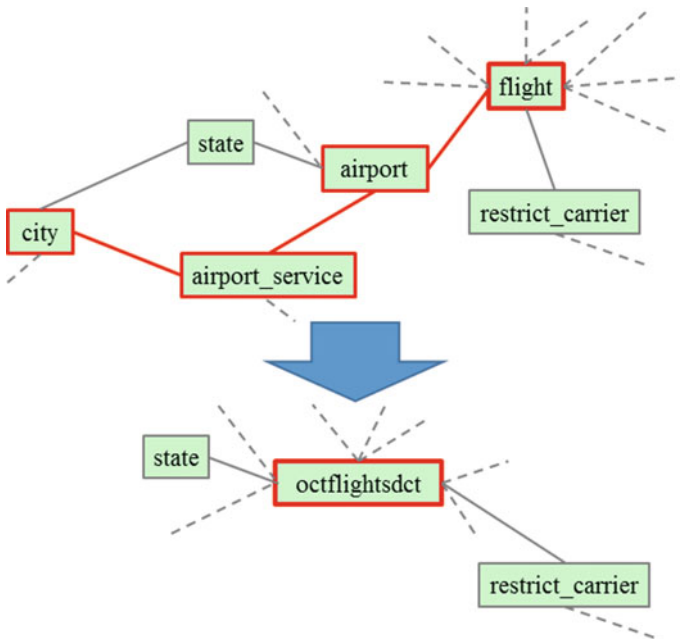


Fig. 4 Semantic graph involving a semantic view

WHERE *octflightsdct.OC_city_name* LIKE 'Fort Worth' AND
octflightsdct.DC_city_name LIKE 'New York';

The query described in Statement 2 uses the materialized view created by Statement 1. However, using a materialized view is very costly in resources for the DBMS. Therefore, in this project we opted for using a virtual view in order to improve the response time of the NLIDB for DBs with large volumes of information.

4 Implementation of a Virtual View

In this chapter we present a method for using a virtual view instead of a materialized view in a NLIDB, which allows avoiding the excessive use of memory resources and processing time due to the use of a materialized view.

It is important to mention that the view implemented in the previous version of the NLIDB, is a materialized view due to limitations of the DBMS that it uses. The materialized view is a table that is created by including all the columns of all the tables involved in the semantic view. Afterwards, all the rows corresponding to the information obtained from the joins between the tables are added to the view.

The implementation of the virtual view consists in transforming a query that involves a semantic view to a query that involves only base tables (Statement 3

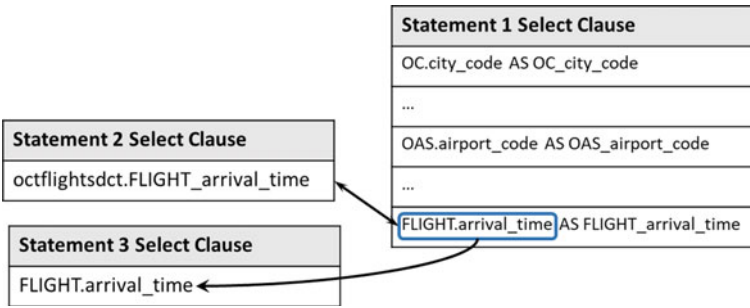


Fig. 5 Example of Select clause generation

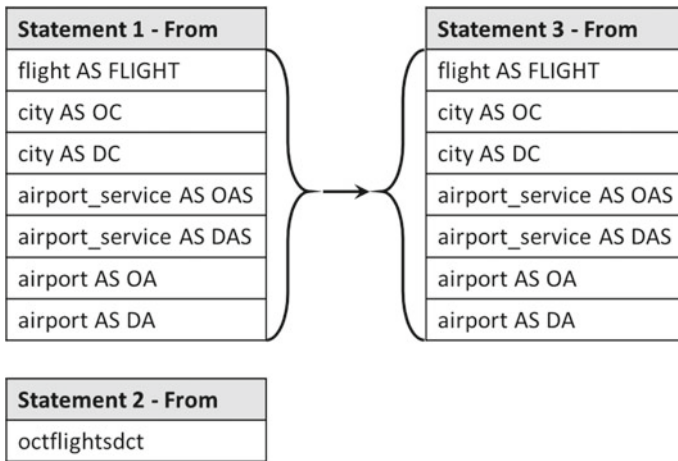


Fig. 6 Example of From clause generation

shown at the end of this section). In this way, the NLIDB will not have to generate the view in the DBMS, and therefore, it will not have to consume DBMS resources.

Before generating the query involving base tables, the NLIDB has already generated the query to create the materialized view (Statement 1) and the statement for answering the NL query (Statement 2).

The NLIDB first generates the Select clause of Statement 3. To this end, in the Select clause of Statement 2, the view columns are substituted by the corresponding columns of the base tables and the table aliases. This information is obtained from the semantic view stored in the Semantic Information Dictionary (see example in Fig. 5). Additionally, if the Select clause of Statement 2 includes columns of base tables, they remain unchanged.

For generating the From clause of Statement 3, in the From clause of Statement 2 the view name is substituted by the names of the base tables (including their aliases). This information is obtained from the From clause of Statement 1 (Fig. 6).

Additionally, if the From clause of Statement 2 includes base tables, they remain unchanged.

The Where clause of Statement 3 is composed of three elements: the existing joins within the semantic view, the existing joins between the semantic view and other base tables, and the search values of the query.

For generating the Where clause of Statement 3, in the Where clause of Statement 2, the view columns are substituted by the corresponding columns of the base tables and the table aliases. This information is obtained from the SID (Fig. 7). Additionally, in the Where clause of Statement 2, the Where clause of Statement 1 (statement for creating the view) is appended.

To exemplify the result of the process described, the following statement is the translation to SQL of the query *When do the flights that leave from Fort Worth arrive in New York?*

Statement 3

```

SELECT FLIGHT.arrival_time
FROM flight AS FLIGHT,
      city AS OC,
      city AS DC,
      airport_service AS OAS,
  
```

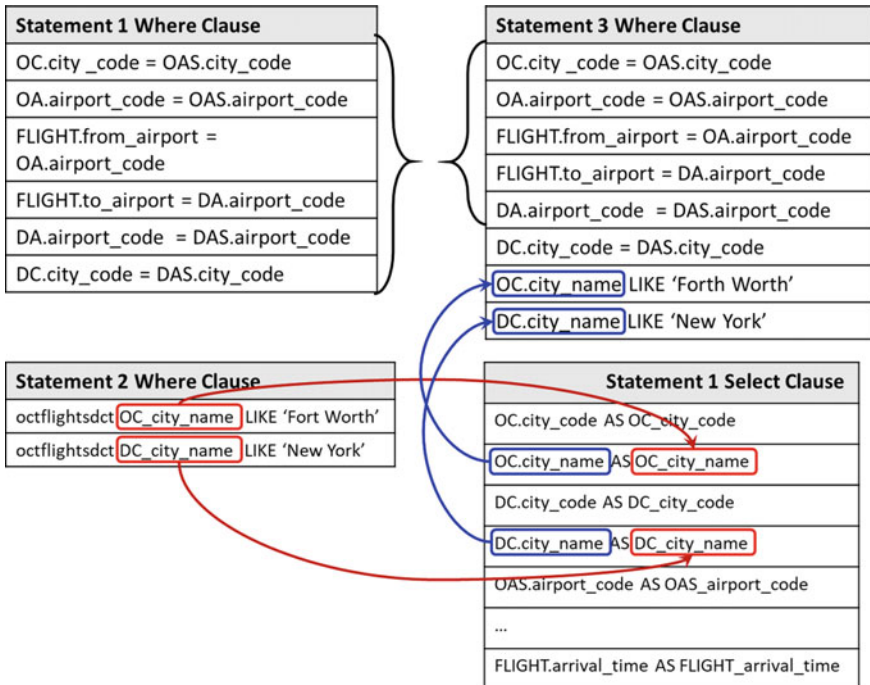


Fig. 7 Example of Where clause generation

```

        airport_service AS DAS,
        airport AS OA,
        airport AS DA
WHERE  OC.city_code = OAS.city_code AND
        OA.airport_code = OAS.airport_code AND
        FLIGHT.from_airport = OA.airport_code AND
        FLIGHT.to_airport = DA.airport_code AND
        DA.airport_code = DAS.airport_code AND
        DC.city_code = DAS.city_code
        OC.city_name LIKE 'Forth Worth' AND
        DC.city_name LIKE 'New York'
    
```

5 Experimental Results

In order to assess the improvement in response time of the NLIDB when using a virtual view instead of a materialized view, comparative tests of the interface with virtual view were performed versus the interface with materialized view.

The tests consisted of using a set of 41 queries from the ATIS corpus, which involve the need for using semantic views. For each query, the response times of the two versions of the NLIDB were recorded. The results obtained are shown in Table 1.

The first column of Table 1 indicates the version of the NLIDB, the second column indicates the total number of corpus queries, the third column shows that the answers of the new version coincide with those of the previous one, the fourth column indicates the processing time that the DBMS uses on average to process the SQL translation of a NL query formulated in the interface, the fifth column indicates the average time it takes the interface to perform the translation of the NL query to an SQL query.

As can be seen, the processing time of the DBMS in the NLIDB using a virtual view is shorter than that of the materialized view. This is explained because the DBMS processes more quickly a query that contains only base tables. Regarding the translation process of the NL query, the translation time is shorter using a virtual view, because it is not necessary to execute Statement 1 to create the materialized

Table 1 Experimental results

NLIDB	No. of queries	Answer coincidences	Proc. time DBMS (s)	Translation time (s)
With virtual view	41	41	0.00407	0.488
With materialized view	41	–	0.00946	0.628

Boldface for numbers 0.0047 and 0.488 is used for emphasizing improvement

view in the DB. This allows the DBMS to consume fewer memory resources and obtaining a lower execution time.

It is important to mention that this test was performed using the ATIS database, which on average contains around 500 to 1000 rows per table involved in the queries. It can be inferred that, when using a virtual view in a DB with a larger number of table rows, the processing of queries using the virtual view would be much more efficient than that of the NLIDB with a materialized view.

6 Conclusions

The NLIDB has a mechanism called semantic view, which is used to answer complex queries; i.e., queries that involve several tables that are not explicitly specified in the NL query. However, the implementation of the semantic view in the previous version of the interface used a materialized view, which behaves like a base table with data from all the tables involved in the semantic view. This makes the previous version inefficient, because the creation of a materialized view consumes a large amount of DBMS resources; specifically, time and memory.

To improve the performance of the NLIDB, a new method using a virtual view instead of a materialized view was designed. This method consists in generating the SQL translation of a NL query from the SQL statement that was used to create the materialized view (in the previous version) and the SQL statement that would be useful to retrieve data stored in the view.

When implementing the virtual view, the NLIDB does not need to execute the SQL statement to create the materialized view; therefore, it does not consume DBMS resources. Additionally, executing an SQL query using only base tables is faster than executing a query involving a materialized view and base tables.

The experimental results show an average improvement of 57% in DBMS processing time when processing complex queries with the new version with respect to the previous one. In addition, the processing time for query translation is 23% smaller in the version with virtual view.

References

1. Androutsopoulos, I., Ritchie, G., Thanisch, P.: Natural language interface to database, an introduction. *J. Nat. Lang. Eng.* 29–81 (1995)
2. Llopis, M., Ferrández, A.: How to make a natural language interface to query databases accessible to everyone: an example. *Comput. Stand. Interfaces Elsevier* **35**, 470–481 (2012)
3. Esquivel, I., Córdoba, R., González, D., Ogarita, E.: SNL2SQL: Conversión de consultas en SQL al idioma Español. In: Congreso Internacional de Investigación, vol. 5, no. 3, Chiapas, Mexico (2013). ISSN 1946–5351
4. Li, F., Jagadish, H.V.: Constructing an interactive natural language interface for relational databases. In: Proceedings of VLDB Endow, vol. 8 (2014)

5. J.D. O'Shea, K. Shabaz, K.A. Crockett, Aneesah: A conversational natural language interface to databases, in *Proceedings of World Congress on Engineering* 2015, vol. 1, London (2015)
6. Amsterdamer, Y., Kukliansky, A., Milo, T.: A natural language interface for querying general and individual knowledge. In: *Proceedings of the VLDB Endowment*, vol. 8, no. 12 (2015)
7. ElSayed, K.: An arabic natural language interface system for a database of the holy Quran. *Int. J. Adv. Res. Artif. Intell. (IJARAI)*, **4**(7) (2015)
8. Kataria, A., Nath, R.: Natural language interface for databases in hindi based on Karaka theory. *Int. J. Comput. Appl.* **122**(7) (2015) (India)
9. Pazos, R.A., González, J.J., Aguirre, M.A.: Semantic model for improving the performance of natural language interfaces to databases. In: *Advances in Artificial Intelligence, MICAI 2011, Lecture Notes in Computer Science*, vol. 7094 (2011)

Automated Ontology Extraction from Unstructured Texts using Deep Learning



Raúl Navarro-Almanza, Reyes Juárez-Ramírez, Guillermo Licea and Juan R. Castro

Abstract Ontologies are computational artifacts to represent knowledge through classes and relations between them. Those knowledge bases require a lot human effort to be constructed due to the need of domain experts and knowledge engineers. Ontology Learning aims to automatically build ontologies from data that can be from multimedia, web pages, databases, unstructured text, etc. In this work, we propose a methodology to automatically build an ontology to represent concepts map of subjects to be used in academic context. The main contribution of this methodology is that does not require handcrafted features by using Deep Learning techniques to identify taxonomic and semantic relations between concepts of some specific domain. Also, due the implementation of transfer learning is not needed of specific domain dataset, the relation classification model is trained with Wikipedia and WordNet by distant supervision technique and the knowledge is transferred to a specific domain by word embedding techniques. The results of this approach are promising considering the lack of human intervention and feature engineering.

Keywords Ontology learning · Deep learning · Recurrent neural network

R. Navarro-Almanza (✉) · R. Juárez-Ramírez · G. Licea · J. R. Castro
Calzada Universidad 14418, Universidad Autónoma de Baja California,
Tijuana, Baja California, Mexico
e-mail: rnavarro@uabc.edu.mx

R. Juárez-Ramírez
e-mail: reyesjua@uabc.edu.mx

G. Licea
e-mail: glicea@uabc.edu.mx

J. R. Castro
e-mail: jrcastor@uabc.edu.mx

© Springer Nature Switzerland AG 2020

O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_50

1 Introduction

The ontologies in computer science are knowledge bases that represent entities and relations between them. The manually ontology construction process requires a huge human effort and time, because it needs domain experts to model the knowledge base, that this is mainly constructed by knowledge engineers, and might be difficult due different vocabulary [13]. The ontologies are compounded by concepts, attributes, taxonomic and non-taxonomic relations. In this word we focused on identify concepts, taxonomic and part-of relations.

This is an approach to construct a domain ontology to be used in domain model in an Intelligent Tutoring System (ITS), those systems propose to emulate the human tutor (teacher) by recommending learning paths and learning objects to the students. ITS are based on the idea that the personalized instruction brings to the student a better way to learn, those ways can be adapted by learning styles, learning goals, motivation, studying time and so many other attributes of the student [10].

The construction of these knowledge models requires a lot of time and human effort, moreover, once there are building the application to another domain, in this case another subject, it is require a lot of time and effort to adapt it to the new domain. Even though, when some ontology is applied to the same domain that was constructed, some times those not share the objective and also some modifications are required [4].

Thus the required effort to construct each domain model of each subject, we need to use automatic ways to build domain models. This is our first approach to automatically build the model domain of this kind of systems. In this step we are focusing in extract those relations to create the structure of concepts in some subject, the relations that this approach handles are taxonomic (is-a) and parf-of relations, we consider that these two relations are enough to bring to ITS the capability to construct personalized teaching paths based on student actual knowledge.

The relation detection among entities is an important task in automatic Knowledge Base Construction (KBC), brings capability of reasoning and discover new knowledge from existing one.

There is special attention to relation detection without handcrafted features, overcoming the dependency and limited application domain that implies those approaches. Handcrafted resources such as WordNet¹ are used on generic domains [3, 5, 24, 25], however, on more specialized domains are more difficult to find structured data to improve KBC. Through more linguistic independent and automatic methods to capture semantics improve automatic KBC processes on specialized domains, especially when knowledge bases are constructed from unstructured data.

One of the novel word representation models is neural word embeddings, which is the basis of our proposal. Embedding word representation aims to reduce a word vector representation into lower dimensionality and continuous vector space. Neural network language models have been used on word embeddings [19, 27]. In [19], it is

¹<https://wordnet.princeton.edu/wordnet/>.

demonstrated that many syntactic and semantic regularities can be captured in those embedding representations.

The available datasets for relation classification such as SemEval 2010 Task 8 [11] promote comparison between state of the art results and naturally thrust to make more prominent models [7, 8, 15, 22]. In this dataset there is various kind of relations, however, in this work we are interested in identifying relations such as *hypernym-hyponym* and *holonym-meronym*, because of that, we use distant supervision to create a dataset from WordNet and Wikipedia.²

The aim of our proposal is to build an automated system to build and knowledge base of subjects to be used in ITS. The domain of the subject is an specialized domain, that fact difficult to get enough data to train machine learning models. Because of the lack of training data, we apply a technique called distant supervision to take advantage of the existing knowledge base and resources. The distant supervision technique is based on the idea that if there are some relations between concepts in a knowledge base, that relation might be reflected in a corpus that contains both concepts in a phrase.

Due the need of training generic domain models to generalize specialized domains, we use WordNet knowledge base that contains human labeled concepts and relationships between them, for the generic domain corpus, we use Wikipedia (dump 2016).

This study is focused to answer these research questions:

- (RQ1) How well Deep Learning techniques can learn semantic relation between terms?
- (RQ2) Are distant supervision technique enough to generalize relations between concepts?
- (RQ3) Is it feasible apply transfer learning in SRC when the model is used with other data distributions?

Remaining paper is structured as follows: Sect. 2 defines general concepts of problem domain, Sect. 3 consists of related work on Ontology Learning methods, Sect. 4 describes the proposed methodology for Automatic Ontology construction through Semantic Relation Classification, Sects. 5 and 6 present the experiment and results of our proposal; Sect. 7 consists of some conclusions and future work.

2 Background

2.1 Word Representation

Word embedding consists on representing each word in a real value vector, there are many ways to do it, the method that we used in this work is skip-gram model proposed by Mikolov et al. [19] trained on Wikipedia articles (dump 2016). Skip-gram model

²<https://www.wikipedia.org/>.

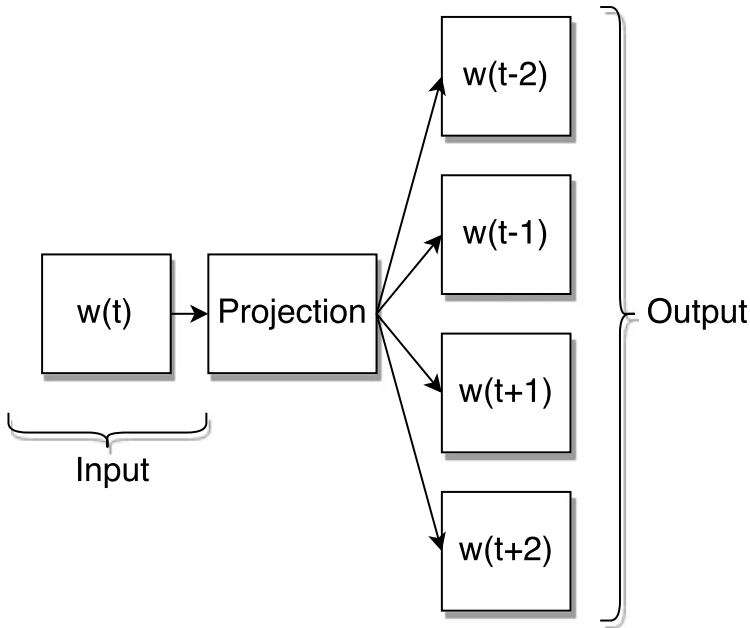


Fig. 1 Skip-gram model architecture [19]

aims to predict context words $w_{context} = (w_{t-i}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+i})$ where $i \geq 1$ given a word $w(t)$ as input (Fig. 1). The projection is a low-dimensionality and continuous vector space representation of the word $w(t)$.

2.1.1 Distant Supervision

The traditional way to train a learning system is by examples in supervised learning, the datasets that contains contains those samples are mostly created and labeled by humans. This approach is expensive and also could have noisy data due error in labeling process, moreover, the dataset are focused in one objective and it is difficult to reuse in another task. In text analysis there are a technique called distant supervision, the objective of this technique is to automatically generate training data using existing resources (mostly manually created) to extract relations and extract sentences from corpus that contained them. This technique requires at least two resources, one that must contains relations between entities (commonly a database), and a corpus in which might appear those entities.

The database used in this work due the generality of its domain was WordNet, specifically we extracted the taxonomic and part-of relations. In the other hand, we select Wikipedia as the corpus by the same reason of WordNet, the generic domain comprising.

WordNet is used as knowledge base to extract semantic relations among words. Randomly 9,000 relations are selected for each class (*hypernym*, *hyponym*, *holonym*, *meronym*) to maintain balanced number of instances (36,000 instances in entire dataset).

2.2 Deep Learning

Back in days machine learning techniques required a lot of human effort to carefully design features from raw data. Deep Learning methods are capable to deal with raw data (such as text, pixel data from an image) through multiple levels of representation where each level transform the previous data by non-linear modules, the most important thing about DL is that each representation of the data is not modeled by humans, actually are discovered automatically by the models [17].

2.2.1 Recurrent Neural Networks

The Recurrent Neural Network, specifically the type Long Short Term Memory (LSTM) proposed by Hochreiter and Schmidhuber [12] to overcome gradient vanishing problem. This model incorporate a mechanism that permit memorize the current feature data to be used in future inputs. The components that composite those models are the input gate (i_t), forget gate (f_t), output gate (o_t) and cell state (c_t).

$$\begin{aligned}
 i_t &= \sigma(W_{x_i}x_t + W_{h_i}h_{t-1} + W_{c_i}c_{t-1} + b_i) \\
 f_t &= \sigma(W_{x_f}x_t + W_{h_f}h_{t-1} + W_{c_f}c_{t-1} + b_f) \\
 g_t &= \tanh(W_{x_g}x_t + W_{h_g}h_{t-1} + W_{c_g}c_{t-1} + b_g) \\
 c_t &= i_t g_t + f_t c_{t-1} \\
 o_t &= \tanh(W_{x_o}x_t + W_{h_o}h_{t-1} + W_{c_o}c_{t-1} + b_o) \\
 h_t &= o_t \tanh(c_t)
 \end{aligned} \tag{1}$$

3 Related Work

3.1 Semantic Relation Classification

Deep learning techniques have been state-of-art in semantic relation extraction [18, 26, 28], most of the approaches are based on word embedding representations. Those representations have been show to capture semantic regularities between words [19]. Most of the approaches relies on Deep Learning models such as Convolutional Neural Networks and Recurrent Neural Networks.

In Semantic Relation Classification we need to learn long-range dependencies between the terms. To capture the semantic relation across all the sentence, we use the model proposed that is the state-of-art in semantic relation extraction by Zhou et al. [28] a Bidirectional Long Short Term Memory. This configuration of the LSTM contains two sub-networks for the left and right sequence context, which are forward and backward pass respectively, the output of the model are shown in the Eq. 2.

$$h_i = \left[\vec{h}_i \oplus \overleftarrow{h}_i \right] \quad (2)$$

3.2 Automated Ontology Construction

There are some approaches that aim to automatically build an ontology to represent the concept map of some subject. This kind of knowledge representations have been used on Intelligent Tutoring Systems which try to improve learning process of the students. The main idea of represent the domain model of some subject is to automatically find the best learning path to the students based on their characteristics such as learning style, actual knowledge, studying time, etc.

However, this kind of knowledge bases require a lot of human effort to construct them, because of that, there are some proposals to construct those models from academic material such as textbooks, power point slides and lecture notes [1, 6, 9].

In the work [1] aims to automatically generate a knowledge base for domain representation from lecture notes. Their methodology relies in Natural Language Processing (NLP) techniques such as part-of-speech tagging and weighting model to extract key terms. For concept hierarchy extraction, their proposed to use PowerPoint document layout structure identifying topic, title, bullet and sub bullet, after that, they apply a link-distance algorithm to determine which extracted concept are more general based on their number of links to form the hierarchy structure. The reported results in this work are precision: 35%, recall: 60% and f-measure: 42%.

Another related work that have the same objective, automatic generation of knowledge base to an ITS, analyzing textbooks structure to extract key terms and their relationship [6]. They proposed a methodology that relies in NLP techniques and heuristics, based on how the sentence are structured, to find by patterns the relation (is-a and part-of relations) between concepts. Also, they propose analyze the outline to find relevant concepts and another relationships. The results reported in this experiment (specifically on part-of and taxonomic relationships) are recall of 55.81% for part-of relationships and recall of 30.30% and precision of 56.10% for is-a relationships.

This work have a similar objective than both works, automatically build a knowledge base to represent a domain model to be use in educational systems. Those works require structured data as input to be analyzed and extract concepts and relationships, in some way those method still need some processing to prepare the data, moreover, they need NLP techniques process the corpus. The main difference to this work is

that the input of the proposed methodology support unstructured data, although, in the experiments we use textbooks as input, they are passed as a single text document without any other manually process. The aim of this work is to analyze how Deep Learning techniques for text processing can support OL to make their tasks easier to apply and without human effort.

4 Ontology Learning Methodology using Deep Learning Techniques

The proposed methodology to automatically build an ontology from raw text through deep learning techniques. This methodology is composed by twelve tasks, the first steps are the selection of the generic corpus and the generic database, that must contains entity pairs relations; and ends with the ontology construction. The proposed methodology can be grouped in three main modules, which are: (i) Model training for semantic relations detection in a generic model; (ii) Domain adaptation through transfer learning techniques; and (iii) Inference of semantic relations in the specific domain application.

The involved general tasks in the methodology are presented in the following list, then it will presented in detailed form how these tasks must be applied and the data structure definition for the inputs and outputs.

1. Model training for semantic relations detection in a generic domain
 - a. Selection of generic corpus.
 - b. Selection of the database for extract relationships between the entities.
 - c. Automatic database construction of semantic relations in the generic domain by distant supervision technique.
 - d. Generation of word embedding space of the generic corpus.
 - e. Construction of the dataset conformed of semantic relations of the generic domain.
 - f. Training the machine learning model for classification of selected semantic relations.
2. Domain adaptation
 - a. Retraining of word embedding space with the words in the specialized corpus.
3. Inference of semantic relations in the specific domain application
 - a. Automatic preparation of the specialized domain dataset.
 - b. Prediction of the semantic relations in the specialized domain.
 - c. Construction of the ontology based in the semantic relations detected previously.
 - d. Evaluation of the extracted terms in the resulted ontology.
 - e. Evaluation of the extracted semantic relations in the resulted ontology (Fig. 2).

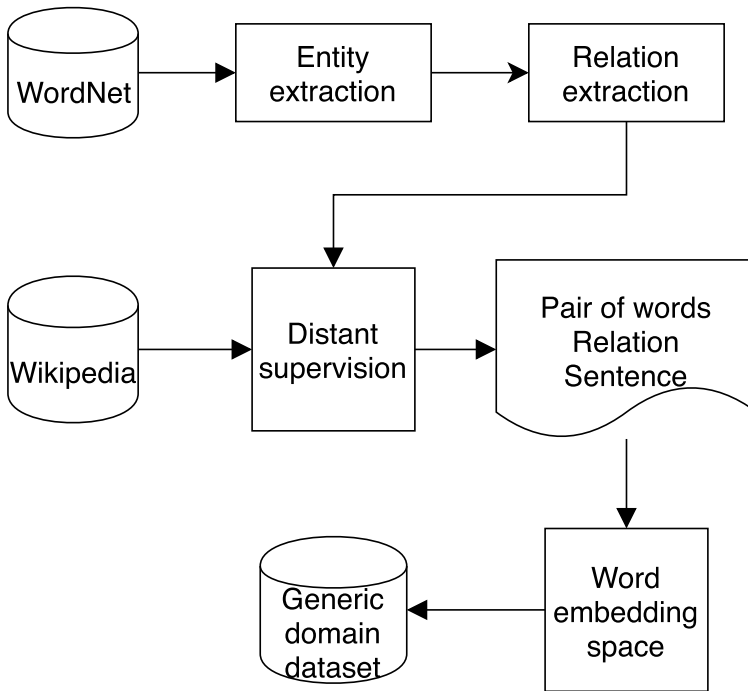


Fig. 2 Generic domain dataset construction using distant supervision

4.1 Model Training for Semantic Relations Detection in a Generic Model

This phase consists in the creation of a learning model for the classification of semantic relations given a pair of entities and the shared context. At this point, the quantity and quality of information available should be considered, since the proposed techniques for the classification of semantic relationships are not viable when there is little data, in addition, in the case of text processing, since they contain a large number of variables (each word) is also required a large number of instances (sentences) to detect patterns between words.

The semantic regularities (relations) can be generalized in the space of embedded words, however, by not considering the shared context between the entities, the inference of relations in a specialized domain is not feasible due, mainly, to the available quantity of information of the domain of interest and the different distribution of data. Therefore, in this phase the context between the entities is considered when training the learning model, which is based on the state of the art in the task of classification of semantic relation [28].

In the rest of the section, each of the steps involved in this phase is described, beginning with the selection of the corpus and the knowledge base, and ending with the trained learning model for the detection of semantic relationships between entities.

4.1.1 Generic Corpus and Database Selection

The selection of the corpus is a crucial step for the training of the learning model for the detection of semantic relations, since the quality of the model depends on the quality of information with which it has been trained. There are several factors to consider at this point, such as the size of the corpus, this is due to the fact that the textual data are complex for processing since the meaning and intention of the words depend on the context in which they are, therefore, the more contexts are considered, the better representation of the terms considered is presumed and possibly a better generalization of the problem. Taking into account the quality of the information in the corpus, it is also important to consider the process with which it was created, for example if it was built automatically through extraction of websites or some other source that may contain information with noise, or, if it was built manually.

By generic domain corpus, it is referred to a corpus that does not have a particular topic, that is, that covers as many topics as possible with the idea that at some point it is considered in the training of the model a data distribution similar to which will be the objective domain (or specialized domain) that is desired to generate an ontology for its representation.

On the other hand, with regard to the selection of the knowledge base that explicitly contains the relationships between term or entities, factors similar to those described in the selection of the corpus should be considered, as are the number of relationships contained, the number of terms or entities, the form with which it was built and finally the relationships it contains, since the learning model will be trained to detect the relationships contained in the knowledge base, therefore, it is essential to consider this factor.

For the selection of both sources of information, the intersection of entities or terms between them should be maximized, in such a way that when generating the training data set, the greatest amount of information available is obtained to increase the quality of the resulting model. Therefore, the domain of the corpus and the domain of the knowledge base must be the most similar, this with the objective of maximizing the probability of the same vocabulary in ideally identical or as similar contexts as possible, this with the objective of reduce the ambiguity of the terms.

4.1.2 Automatic Generic Domain Semantic Relation Database Construction, Using Distant Supervision Technique

At this point, the technique of distant supervision is used, consisting of the construction of a set of training data, through the integration of two resources, such as a corpus

and a knowledge base where some type of relationship is explicitly represented between terms or entities. The general idea of this technique is that the sentence that contains both terms, therefore, the learning model could find a pattern in that context (surrounding words) and generalize it for another pair of terms.

One of the objectives of this technique is to create datasets automatically without human intervention, which implies a saving of effort and time in the process of building a knowledge base. Specifically, the technique of distant supervision is defined in this methodology with the following steps:

1. Extraction of the terms or entities of the knowledge base, represented by the set $\{\tau_1, \tau_2, \dots, \tau_{|\Gamma|}\} \in \Gamma$.
2. Extraction of instances of semantic relations contained in the knowledge base, in this point the knowledge base is considered as a directed graph, in which each relation must be formed by an edge that goes from a source vertex to a destination vertex, represented by the tuple $\langle \tau_i, \tau_j, r_k \rangle$ where τ_i is the origin term and τ_j the destination term, linked by the semantic relation $r_k \in \mathcal{R}$, where \mathcal{R} is the set of semantic relationships of interest contained in the knowledge base (for example, synonymy, hyperemia, meronymy, among others).
3. For each instance extracted in the previous step, consider the terms or entities τ_i y τ_j for their search in the selected corpus, where both terms must be contained in the same sentence (phrase), it is say $\{\tau_i, \tau_j\} \in \rho_{i,j}$ where $\rho_{i,j}$ is the extracted phrase. If possible, it is advisable to extract several sentences where both terms are contained, this in order to consider as many contexts as possible.
4. Finally, the training data set is formed where the terms and the words in it are in their natural state, represented by characters of the language in question. The training data set will be defined by the tuple $\langle \mathcal{X}_S^*, \mathcal{Y}_S \rangle = \{\langle \tau_i, \tau_j, \rho_{i,j}^m \rangle, \langle r_k^m \rangle\}^M$, where $m = 1, 2, \dots, M$ represents the total number of instances built, with the restriction that the sentences $\rho_{i,j}$ should not be repeated given a pair of terms τ_i and τ_j ; the set \mathcal{X}_S^* is made up of natural language words, the set $\mathcal{Y}_S \in \mathcal{R}$ being the desired output of the training model.

The distant supervision technique requires search each term pair in the corpus to extract the sentence which appears both terms. In this step we used Wikipedia API to search the concepts in Wikipedia corpus. We select ten sentences at maximum for each pair terms.

4.1.3 Create Word Embedding for Generic Domain Corpus

Word embedding representations have been used to represent words in a low dimensional space, which allows more efficient computation, also those low-dimensional representation captures semantic relation between the words, for example, as can be seen in the well-known expression $v(\text{king}) - v(\text{queen}) \equiv v(\text{man}) - v(\text{woman})$. These new word representations allow work with models without handcrafted feature in learning models (Fig. 3; Table 1).

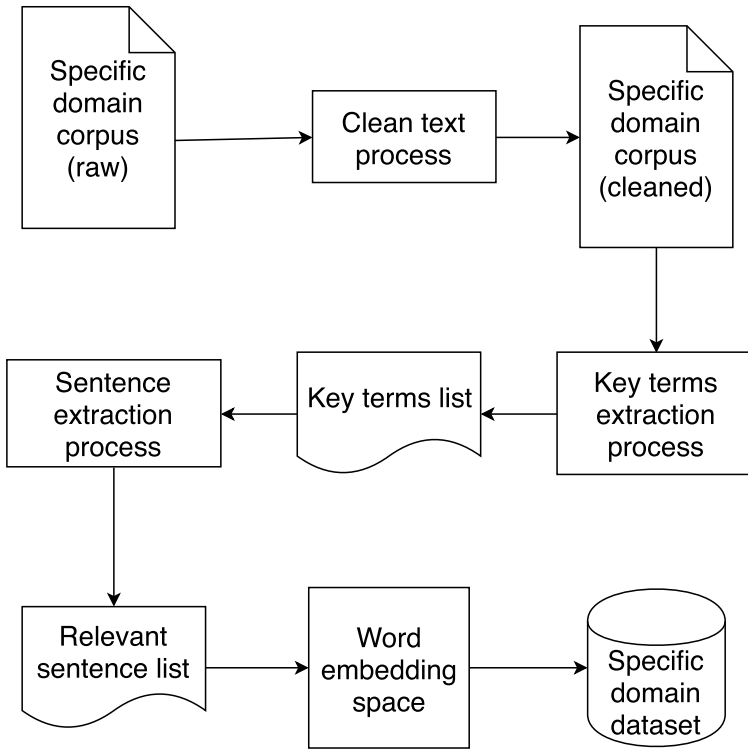


Fig. 3 Specific domain dataset construction process

Table 1 Semantic relations selected in this study case

Relation	Representation	Inverse relation
Hypernym	$X \overrightarrow{\text{hypernym}} Y$	$Y \overrightarrow{\text{hyponym}} X$
Hyponym	$X \overrightarrow{\text{hyponym}} Y$	$Y \overrightarrow{\text{hypernym}} X$
Holonym	$X \overrightarrow{\text{holonym}} Y$	$Y \overrightarrow{\text{meronym}} X$
Meronym	$X \overrightarrow{\text{meronym}} Y$	$Y \overrightarrow{\text{holonym}} X$

There are several methods to create this representation of the words, however, given the results obtained in the previous study [20] we will use *skip-gram* and *CBOV* [19] models, whose word embedding representation is generated through an ANN model, where the process for the creation of this new representation is conceived from a perspective of a classification problem, however, the preparation of the entries is automatic, so from the perspective of the analyst of data this method can be considered as unsupervised.

In the case of *skip-gram* model, given a word as a input (w_t) in the Neural Network model, we seek to obtain a set of words with the highest probability of appearing in the context where the input word is found $w_{context} = (w_{t-i}, \dots, w_{t-1}, w_{t+1}, \dots, w_{t+i})$ where $i \geq 1$ and is a parameter to consider called *window*, which refers to the number of words to be considered as context in a sentence. Where each word w_t can be represented in *one-hot-encoding* form, such that $w(t) \in \{0, 1\}^{|V|}$ where $|V|$ is the size of the vocabulary to consider. In this step, the training of the linguistic neuronal model of the generic domain corpus (or source domain), defined as $C_S \in V$, where V is the vocabulary handled in this domain. Once the model is trained, a hypothesis $h : w_t \rightarrow w_{context}$ is generated in such a way that the representation of the word is defined by the output of the neurons in the hidden layer, whose space is represented by $\tilde{h}(w_t) \in \mathbb{R}^D$ where D is the number of neurons in the hidden layer of the Neural Network, usually $300 \leq D \leq 1000$. This new representation of words is of relatively low dimensionality when compared to the initial dimension $|V|$. This representation of the words allows a more efficient handling of the data, it has been demonstrated that in the new space of low dimensionality it is possible to capture semantic relations between the words, for example in the well-known expression $v(\text{king}) - v(\text{queen}) \equiv v(\text{man}) - v(\text{woman})$. The representation of words in this space allows you to make use of learning models without the need to create the attributes (characteristics or variables) manually.

This step in the methodology, ends with a dictionary of words and their representation in the projection space (or space of embedded words), represented by the following set $\{(w_i, \tilde{h}(w_i))\}_{i=1}^{|V|}$.

4.1.4 Generic Domain Semantic Relation Dataset Building

This task consists of preparing the training data set to be fed into the learning model. The processing data treatment $\langle \tau_i, \tau_j, \rho_{i,j}^m \rangle^M$ defined in previous steps, consists in the transformation of the words, which up to this point are in their state natural, the continuous space and low dimensionality generated in the previous step. Since $\Gamma \subset V$, each term is transformed by its representation in the embedding space \tilde{h} , in addition, each of the words contained in the sentences ($\rho_{i,j}^m$) where each pair of terms appears (τ_i, τ_j) are also transformed into the space of embedded words, through the function $f_{\tilde{h}} : V \rightarrow \mathbb{R}^D$ where $f_{\tilde{h}}(\rho) = \{\tilde{h}(w) | \forall w \in \rho\}$. As a result of the transformations of the generic domain training data set, the following sets are obtained: $\langle \mathcal{X}_S, \mathcal{Y}_S \rangle = \{(f_{\tilde{h}}(\tau_i), f_{\tilde{h}}(\tau_j), f_{\tilde{h}}(\rho_{i,j}^m)), \langle r_k^m \rangle_{m=1}^M\}$, where each of the words is replaced by the representation in the embedded space.

The dataset structure that we construct are based on the dataset SemEval task 8 that contains the pairs of words, the relation and the sentences in natural language representation. After build this dataset representations, all words in the sentence are transformed in a word embedding space. The limit size of each sentence was set to 60 words, therefore, each instance are transform to a matrix with dimensions 60×300 , where 300 are the dimension used in the word embedding space.

4.2 Training the Semantic Relation Classification Model

In this last step of the first phase consists in the training of the learning model for the detection of semantic relations between two terms or entities, through the context between both.

The input of the model is represented by the set $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_M\}$, where each instance $\mathcal{X}_m \in \mathbb{R}^{\ell+2 \times D}$, ℓ is the maximum number of words contained in each of the extracted sentences (ρ) and D is the dimension of the embedded space; each instance in the input data set is composed of the concatenation of the matrices of the representation in the embedded space of the terms and the sentence (Eq. 3).

$$\mathcal{X}_m = [f_h(\tau_i)_{1 \times D} f_h(\tau_j)_{1 \times D} f_h(\rho_{i,j}^m)_{\ell \times D}]_{\ell+2 \times D} \quad (3)$$

The desired outputs is represented by \mathcal{Y} , where for each entry \mathcal{X}_m corresponds an output $\mathcal{Y}_m \in \{1, 2, \dots, K\}$, where K is the number of classes corresponding to the semantic relations (\mathcal{R}). So that each tuple contained in the training data set is represented as the following set $\{(\mathcal{X}_m, \mathcal{Y}_m)\}_{m=1}^M$ to be used in the learning model $\mathcal{M}: (\theta, \mathcal{X}) \rightarrow \mathcal{Y}$, where $\theta = \{\theta_1, \theta_2, \dots, \theta_{|\theta|}\}$ are the parameters of the model \mathcal{M} to adjust the input data \mathcal{X}_m with those of the desired output \mathcal{Y}_m . In the training stage of the model the parameters θ are updated by means of a learning algorithm (for example, *gradient descend*), until arriving at a desirable error $Error(\mathcal{Y}_m, \hat{\mathcal{Y}}_m) \leq \epsilon$, where $\hat{\mathcal{Y}}_m$ is the actual output of the model \mathcal{M} and ϵ is the error threshold of tolerance of the model.

The resulting in this step is the \mathcal{M} trained model, with the pseudo-optimal parameters θ for the detection of semantic relationships between terms.

The selected model was based on the state of art in semantic classification model using SemEval task 8 dataset. The state of art model is a Bidirectional Gate Recurrent Unit Neural Network with attention technique, those types of Neural Networks have been used in sequential data such as video, speech and text processing. The parameters used to train this model was based on [28].

4.3 Domain Adaptation

In the process of supervised learning, learning models are trained through a set of training data, which belong to a certain domain. After the learning process, the models when receiving new entries have as output a response belonging to the solution space, as long as the entries belong to the distribution of the training data set. However, sometimes the evaluation of the models is done with different data distribution, for example if a model for the recognition of faces with some images of light intensity was trained, when evaluating the system with images with a lower intensity (for example, captured at night), it is likely that the model fails to place the response in the solution space correctly; in the case of handling a set of text (corpus) of a certain

domain (topic) \mathcal{D}_S , when training a learning model it is also likely that it will not be able to grant a correct output if it is applied in another domain \mathcal{D}_T , consider also the use of words that do not exist in the domain \mathcal{D}_S in the domain \mathcal{D}_T .

When these types of situations are presented in which the distribution between the training and evaluation data are different, the distribution in which the trained model will be evaluated should be re-labeled. There are several possible scenarios that cause the learning models to malfunction, such as different attributes of the data between the domain \mathcal{D}_S and \mathcal{D}_T ($\mathcal{F}_S \neq \mathcal{F}_T$), different distribution of marginal probabilities between both domains ($P(X_S) \neq P(X_T)$), same space of characteristics but different desired outputs ($Y_S \neq Y_T$), different conditional probability between the domains ($P(Y_S|X_S) \neq P(Y_T|X_T)$).

In text mining scope, it is common to apply learning transfer techniques since the training corpus does not generally belong to the same marginal distribution of the evaluation dataset [21]. One of the most common processes within the transfer of learning applied in models such as Neural Networks is to re-train the data available in the domain \mathcal{D}_T with the weights calculated in the training with the set of domain data \mathcal{D}_S .

In this methodology we propose to apply learning transfer by domain adaptation, which is done when both domains \mathcal{D}_S and \mathcal{D}_T have different distribution of marginal probabilities $P(X_S) \neq P(X_T)$, however, have in common the task to be performed. When making a domain adaptation, so that the model trained in the domain \mathcal{D}_S can be applied in the domain \mathcal{D}_T , the type of learning transfer that is used is transductive.

The input of the learning model is defined by $\mathcal{X} \in \mathbb{R}^D$ where D is the representation dimension of the words in the embedded vector space, with the system output defined for $\mathcal{Y} \in \{1, \dots, K\}$ where K is the number of classes. The tuple $\langle \mathcal{D}, \mathcal{S}, \mathcal{T} \rangle$ defined for a classification problem where \mathcal{D} is the distribution $\mathcal{X} \times \mathcal{Y}$, \mathcal{S} is a set of training data, of m samples of observed phenomenon, represented as $\{(x_i, y_i)\}^m$ where $i = 1, 2, \dots, m$; and \mathcal{T} is the data set used for the evaluation such that $\mathcal{S} \cap \mathcal{T} \equiv \emptyset$. The learning process consists of discovering a mathematical model $h: \mathcal{X} \rightarrow \mathcal{Y}$ to represent the behavior of a given phenomenon through samples (\mathcal{S}). In this step of the methodology we propose the adaptation of the domain of the data distribution, unifying the distribution of the data, that is, a common representation for both domains that is achieved by embedding the data \mathcal{X}_T in the space representation \mathcal{X}_S , procedure described below.

4.3.1 Retrain Word Embedding Vector Representation with Specific Domain Corpus

For the adaptation of the domain between the data set of the source domain \mathcal{X}_S to the target domain \mathcal{X}_T , it is proposed to do it through the embedding of the corpus of the specific domain in the space of Embedded words created from the corpus of the generic domain.

The specialized domain corpus is defined as $\mathcal{C}_T \in V_T$ where $V_T = \{w_1, w_2, \dots, w_{|V_T|}\}$ is the vocabulary of the specialized domain, due to the difference of domain

between the corpus, the vocabulary used is not the same $V \neq V_T$. So a function is defined for the transformation of the vocabulary of the specialized domain to the domain of the embedded words of the generic domain $f_{\tilde{h}} : V_T \rightarrow \mathbb{R}^D$.

The specific domain corpus extracted from textbooks are pre-processed to decrease the amount of noise in the embedded word model. The process of cleaning the document of the specific domain consists of the following processes:

1. Empty lines.
2. Punctuation marks.
3. Elimination of non-alphanumeric characters.
4. Elimination of sentences with less than N words.
5. Transformation of lowercase words.

Once the corpus of the specific domain is applied the aforementioned cleaning procedures are denoted as \tilde{C}_T , and the re-training of the embedded word model is followed, adding the domain vocabulary specific to the existing vocabulary. In such a way that a set is created that contains the new terms of the specific domain transformed into the embedded space \tilde{h} . It is important to note that the training of the linguistic neuronal model was based on the weights generated in the construction process of the space embedded with the corpus of the generic domain. The resulting set in this step is composed of the dictionary of the terms that appear in the corpus of the specific domain ($\tilde{C}_T \in V_T$), defined as $\{(w_i, \tilde{h}(w_i))\}_{i=1}^{|V_T|}$.

4.4 Inference of Semantic Relations in the Specific Domain Application

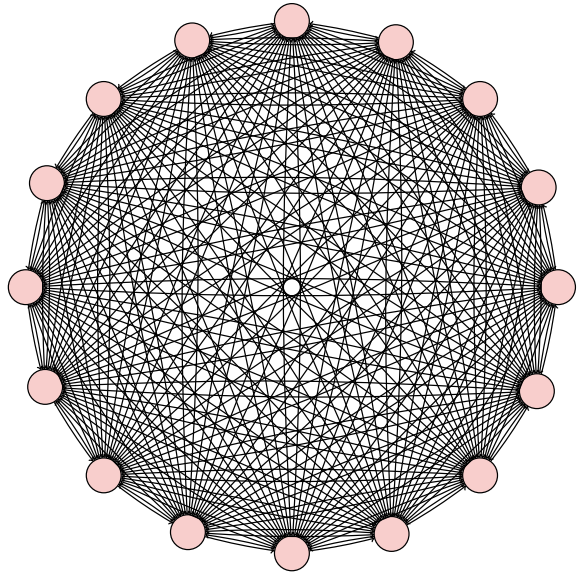
In this phase the inference of existing semantic relationships between the relevant terms of the specific domain is made. The inference of the relationships is done through the evaluation of the model \mathcal{M} trained with the corpus of the generic domain C_S , however, the evaluation is done through the data set input \mathcal{X}_T , whose words are represented in the modified embedded space for the specialized domain (\mathcal{D}_T).

The steps that constitute this phase consist in the preparation of the data with data held in the corpus of the specific domain up to the inference of inferred relationships among the relevant terms of the domain. In this phase, the corpus of the specific pre-processed domain defined in previous steps is used, such as $\tilde{C}_T \in V_T$.

4.4.1 Automatic Specific Domain Dataset Generation

The tasks to be carried out in this step consist in the identification of the potential relevant terms of the specific domain (\mathcal{D}_T), which can be granted directly or be discovered through techniques such as (Latent Dirichlet Allocation) LDA [2]; Latent Semantic Analysis (LSA) [16]; TextRank [9]; among others. Due to the approach of this proposal, the extraction of semantic relationships, is not limited to the use of

Fig. 4 Illustration of complete graph $\mathbb{K}_{|\tilde{\Gamma}|}^*$ considering 16 relevant terms of the specific domain



a particular technique to perform this task. However, the use of some technique in which the intervention of a data scientist is not required, this is because of the main idea of this methodology, which is the complete ontology automation of a specific domain.

Once the set of relevant terms of the specialized domain are identified, defined by $\tilde{\Gamma}$, the procedure for the construction of the data set through the remote monitoring technique is used.

In this step you can represent the data as the complete digraph $\mathbb{K}^* = \langle \tilde{\Gamma}, E \rangle$, that is, each pair of terms is connected by an edge where E is the ordered set of edges that connect to the vertices (relevant terms of the domain $\tilde{\tau} \in \tilde{\Gamma}$), which is a complete digraph containing $|E| = |\tilde{\Gamma}|(|\tilde{\Gamma}| - 1)$ edges, illustrated in Fig. 4.

Each pair of terms are searched in the pre-processed corpus \tilde{C}_T for the extraction of sentences where the terms are contained.

For each pair of terms contained in the set $\tilde{\Gamma}$ a search is made in the corpus of the specific domain (\tilde{C}_T) where they appear in the same sentence, represented for the set $\tilde{\rho} = \{\tilde{\rho}_{i,j} | \forall \{\tilde{\tau}_i, \tilde{\tau}_j\} \in \tilde{\Gamma}, \forall (\tilde{\tau}_i, \tilde{\tau}_j) \in \tilde{\sigma}, \tilde{\tau}_i \neq \tilde{\tau}_j\}$, where $\tilde{\sigma} \in \tilde{C}_T$ are the sentences contained in the corpus of the specific domain. Since $\tilde{\rho} \subset \tilde{\sigma}$, the structure of the resulting data becomes the directed graph $\mathcal{G}^* = \langle \tilde{\Gamma}, \tilde{E} \rangle$, where $\tilde{E} = \{\forall (\tilde{\tau}_i, \tilde{\tau}_j) \in \tilde{\rho}\}$, which is a directed graph not complete.

The resulting set of this process is defined as $\mathcal{X}_T^* = \{(\tilde{\tau}_i, \tilde{\tau}_j, \tilde{\rho}_{i,j}^n)\}^N$, where $n = 1, 2, \dots, N$; and represents the input data set to the model \mathcal{M} , in natural language.

4.5 Specialized Domain Semantic Relation Dataset Building

The procedure for the construction of the input data set for the inference of relations in a specialized domain is similar to that of the generic domain, except for the embedded space in which the transformation of the words contained in the resulting set will be performed.

The treatment performed in each instance of the set $\mathcal{X}_T^* = \{(\tilde{\tau}_i, \tilde{\tau}_j, \tilde{\rho}_{i,j}^n)\}_{n=1}^N$ consists of the transformation of each of the words contained in the set to the embedded space created with the corpus of the specific domain, represented by the function $f_{\tilde{h}} : V_T \rightarrow \mathbb{R}^D$ where $f_{\tilde{h}}(\tilde{\rho}) = \{\tilde{h}(w) | \forall w \in \tilde{\rho}\}$, since $\tilde{\Gamma} \in V_T$, each of the words identified as relevant terms in the domain \mathcal{T} are defined in the modified embedded space \tilde{h} . In such a way that the resulting set is defined as $\mathcal{X}_T = \{(f_{\tilde{h}}(\tilde{\tau}_i), f_{\tilde{h}}(\tilde{\tau}_j), f_{\tilde{h}}(\tilde{\rho}_{i,j}^n))\}_{n=1}^N$. As you can see, the dataset \mathcal{X}_S is similar to the set \mathcal{X}_T , except that in the latter the words contained in the set are established by their representation in the vector modified embedding space \tilde{h} .

4.5.1 Semantic Relation Prediction in the Specific Domain

In this step of the phase of the inference of semantic relations in the specific domain, the evaluation of the \mathcal{M} model is performed with the input data set \mathcal{X}_T . You can represent, in this step, the model \mathcal{M} as a function $h_{\mathcal{M}} : \{\theta, \mathcal{X}_T\} \rightarrow Y_T$ that has as input the set in the domain of the characteristics of the specialized domain and the pseudo-optimal parameters obtained in the training stage of the model; and therefore, as output, the semantic relation type $r_k \in \mathcal{R}$ is obtained, which corresponds to the pair of terms $\{\tilde{\tau}_i, \tilde{\tau}_j\} \in \mathcal{X}_{T_m}$.

Analogously in the training process of the \mathcal{M} model, the appropriate processing of the input data set is carried out, in such a way that it is represented as a matrix of dimensions $\ell + 2 \times D$ same defined training process, ℓ represents maximum amount of words contained in each sentence (ρ), and D is the dimension of the embedded space \tilde{h} and h .

Each instance in the input data set $\{\mathcal{X}_{T_1}, \mathcal{X}_{T_2}, \dots, \mathcal{X}_{T_m}\} \in \mathcal{X}_T$ is composed of the concatenation of the matrices of the representation in the embedded space of the relevant terms $\{\tilde{\tau}_i, \tilde{\tau}_j\} \in \tilde{\Gamma}$ and the sentence $\tilde{\rho}_{i,j}^n$ (Eq. 4).

$$\mathcal{X}_{T_m} = [f_{\tilde{h}}(\tilde{\tau}_i)_{1 \times D} f_{\tilde{h}}(\tau_j)_{1 \times D} f_{\tilde{h}}(\tilde{\rho}_{i,j}^n)_{\ell \times D}]_{\ell+2 \times D} \quad (4)$$

When the model \mathcal{M} is evaluated through the function $h_{\mathcal{M}}$, the prediction of the semantic relations \mathcal{R} represented as $\hat{\mathcal{Y}}_T \in \mathbb{R}^{|\mathcal{R}|}$, that to get the relation $r_k \in \mathcal{R}$, you get the maximum argument in $\hat{\mathcal{Y}}_{T_n} \therefore \mathcal{Y}_{T_m} = \max_{r_k} \hat{\mathcal{Y}}_{T_n}$.

Specifically, once the model is evaluated in the target domain, given an input set $\mathcal{X}_T = \{(f_{\tilde{h}}(\tilde{\tau}_i), f_{\tilde{h}}(\tilde{\tau}_j), f_{\tilde{h}}(\tilde{\rho}_{i,j}^n))\}_{n=1}^N$, we obtain a set with the predictions of the model $\mathcal{Y}_T = \{\hat{\mathcal{Y}}_{T_n}\}_{n=1}^N$. The interpretation of the model can be represented by the

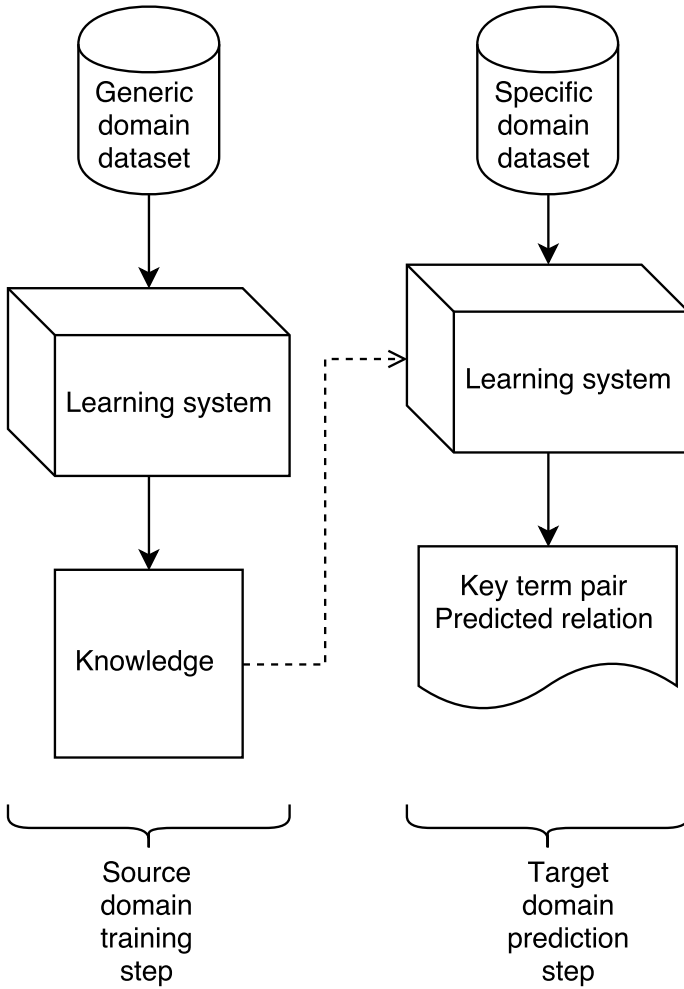


Fig. 5 Ontology learning using transfer learning

set $\mathcal{Z} = \{(\tilde{\tau}_i^n, \tilde{\tau}_j^n, r_k^n)\}_{n=1}^N$, which contains the pair of relevant terms and the relation obtained as a result of the evaluation of the model.

4.5.2 Ontology Building Step

With the resulting set \mathcal{Z} , a directed graph is formed where each term corresponds to a node (vertex) connected to another node through an edge that represents a semantic relation between both terms, so each tuple in the set \mathcal{Z} is represented graphically in the following way $\tilde{\tau}_i^n \xrightarrow{r_k^n} \tilde{\tau}_j^n$. For each instance in the resulting set each pair of

nodes with its respective edge is added to the directed graph, if a node is defined, then only the edge is added to (or from) the other node (which is added to the graph if it does not exist).

In such a way that a graph is defined $\mathcal{G} = \langle \tilde{\Gamma}, \mathcal{R} \rangle$ formed of the terms of the domain and the semantic relationships between them that will later be represented as ontology.

5 Study Case: Ontology of the Object-Oriented Paradigm

The case study that is addressed in this paper is the automatic generation of an ontology on the domain of the object-oriented paradigm. The general characteristics of the ontology to generate are the extraction of relevant terms of the domain and the semantic relationships between them, in terms of semantic relationships, are intended to cover only those referring to hyperonymy, hyponymy, holonymy, meronymy; which are basic relationships that are generally found in ontologies.

The case study that is addressed in this paper is the automatic generation of an ontology on the domain of the object-oriented paradigm. The general characteristics of the ontology to generate are the extraction of relevant terms of the domain and the semantic relationships between them, in terms of semantic relationships, are intended to cover only those referring to hyperonymy, hyponymy, holonymy, meronymy; which are basic relationships that are generally found in ontologies.

The purpose of this case study is to build the domain model of an Intelligent Tutor System, which will represent the general topics of object-oriented programming, interrelated among themselves through the aforementioned semantic relationships. The conditions established for each of the steps mentioned in the proposed methodology will be described later in this section. Once generated the ontology will be evaluated against one carried out manually through a consensus among experts of the domain of the object-oriented paradigm [23], the aspects to be evaluated are: extraction of relevant terms and extraction of relationships between the relevant terms. However, it is important to mention that the proposal is focused on the extraction of semantic relations, the process of extracting relevant terms from the domain is contemplated in the methodology to cover all phases of the ontology construction but is not limited to the use of some technique in particular, but as already mentioned, it is recommended the application of a technique of extraction of relevant terms not supervised, for reasons of the ease of fully automating the implementation of the methodology (Fig. 5).

The ontology of the object-oriented paradigm with which our proposal is to be compared is in the English language, it contains 48 classes (terms or entities), 51 relationships and a maximum depth of 6 levels. The terms that are represented in the ontology are the following: *Object orientation, Abstraction, Concurrency, Modularity, Persistence, Hierarchy, Typing, Encapsulation, Abstract class, Class, Interface, Class, Package, Aggregation, Composition, Inheritance, Scope, Modifiers, Hiding, Encapsulation, Attribute, Class modifiers, Constructor, Method, Object, Single*

inheritance, Multiple inheritance, Subclass, Superclass, Setter, Getter, Abstract, Final, Public, Overloading, Overriding, Protocol, Behavior, Identity, Instance, State, Method modifiers, Parameters, Return, Abstract, Final, Native, Private, Protected, Public, Static, Synchronized, data obtained from the article [23].

The main object of study is the recognition of semantic relationships in the domain of the object-oriented paradigm, through an unstructured corpus. As proposed in the methodology, this process is divided into three different phases, which are: *Training of the model for detection of semantic relations in a generic domain, Adaptation of the domain* and *Inference of semantic relationships in a specialized domain*; The first phase consists of obtaining the pseudo-optimal parameters of the learning model in the generic domain (or source domain), followed by the adaptation of the domain consisting of preparing the data with which the model will be evaluated (objective domain), and in the last phase the inference of semantic relations between the relevant terms of the domain is made.

In this section we describe in detail the characteristics of the data to be used in each of the steps of the proposed methodology, whose sections will be called the same as the phases and steps described in the methodology.

The present experiment is carried out in the academic field, this due to the objective and main motivation of this project, so that the theme as already mentioned will be the theme of the object-oriented paradigm; the application of the resulting knowledge base will be implemented (in later works) in an Intelligent Tutor System, for which basic semantic relationships have been defined, necessary for this system to make inferences about the learning order of each topic of a given subject, such as relationships of hyperonymy, hyponymy, holonymy, meronymy, which are commonly known, the first two as taxonomic relationships or relationships *textit is-a*, the last relations being of the type *textit part-of*. It is important to emphasize that this type of semantic relationships are relations directed by what the order of the elements connected by this edge reflects a hierarchy between the data.

In this experiment, resources were selected in all their phases that were affordable in any domain, within the context of the application of this experiment. So probably, the resources described below could be changed by some with some type of structure or pre-processed and improve the results. However, the focus is on the reuse of this methodology in various domains in the context of automatic curricular maps for computer-aided teaching systems.

Model Training for Semantic Relations Detection in a Generic Domain

In this first phase of the experiment the training of a learning model for the detection of semantic relationships is performed, this in a generic domain, which will indistinctly be called source domain throughout the document. This source domain is constituted by a set of unstructured text documents corresponding not only to a topic or domain, but to several of these, with the intention of generalizing its application in specific domains.

Selection of Generic Corpus and the Database for Extract Relationships Between the Entities

Given the complexity of the type of information inherent in the texts, a large amount of information is required to properly generalize the solution area of the detection of semantic relationships, so the size of the corpus in this step has a very important role, coupled To the foregoing, we must consider a corpus that is generic, that is, that is not focused on a particular domain but encompasses several domains.

The selection of the corpus was considered taking into account the most extensive corpus of a generic domain, that is, that the main topic is not focused on a specific domain. Therefore, the Wikipedia open-access online encyclopedia was selected as a generic domain corpus, due to the ease of access and the amount of information available, in addition to being continuously updated, adding new information.

Once the corpus of the generic domain has been selected, the selection of the database or knowledge base that explicitly contains relations between classes, relevant terms or entities, which must appear in the corpus selected in the previous step, is continued. The domain that covers this information base should be in accordance with the corpus to maximize the set of entities that are in both.

For the selection of the database containing the set of relationships of interest for this project (hyperonym, hyponym, holonym, and meronym), the WordNet knowledge base was selected, since it is constructed manually, which would reduce the inclusion of noise in the learning process of the classification model. In addition, it is not focused on a particular domain, so it makes it a good complement to the selected corpus.

Automatic Database Construction of Semantic Relations in the Generic Domain by Distant Supervision Technique

The main idea in this process is to generate enough training data to represent a generic domain through relationships between entities (concepts). The semantic relationships extracted from the WordNet database are represented by the set $\mathcal{S} = \{hyperonym, hyponym, holonym, meronym\}$.

The technique of distant supervision requires the search of each pair of terms in the corpus to extract sentences in which both terms appear. In this process, the Wikipedia encyclopedia was searched through the Wikipedia API. For each pair of terms, up to 10 different sentences were taken where both terms appeared.

Generation of Word Embedding Space of the Generic Corpus

The representation of embedded words have been used to represent words in a space of low dimensionality. This representation of the words allows a more efficient handling of the data, in addition it has been demonstrated that in the new space of low dimensionality it is possible to capture semantic relations between the words. The representation of words in this space allows to make use of learning models without the need to create the attributes manually.

In this experiment word2vec with skipgram model was use to represent in word embedding space to automatically extract semantic and non semantic relations between the words. In Table 2 is shown the the configuration used in this experiment.

Table 2 Word embedding model parameters

Parameter	Value
Word dimension	300
Window size	10
Model	Skipgram

Construction of the Dataset Conformed of Semantic Relations of the Generic Domain

Once extracted the terms or entities contained in the knowledge base and have performed the search of these in the corpus of the generic domain, represented in the methodology as \mathcal{X}_S^* , the data must be prepared to a numerical domain to be able to be granted to a learning model.

In this step the transformation of the data generated in the previous step was carried out, in such a way that the relevant terms of the specific domain and the words found in the sentence where this pair of terms appear are converted to their representation in the continuous space and of low dimension \tilde{h} . The dimension established in the generation of the embedded space was 300, so each instance has a dimension of $\ell + 2 \times D$, where $\ell = 60$ and $D = 300$.

Training the Machine Learning Model for Classification of Selected Semantic Relations

In this step, a preparation of the data set that represents the desired input and output of each instance is performed.

As described in the methodology, the input data set \mathcal{X} is processed by concatenating the representation of both terms and the statement where these are found (Eq. 3). While the processing of the desired outputs involves the representation in \mathbb{Z} of the types of relations, which refer to 0: Does not exist, 1: Hyperonym, 2: Hyponym, 3: Holonym, 4: Meronym.

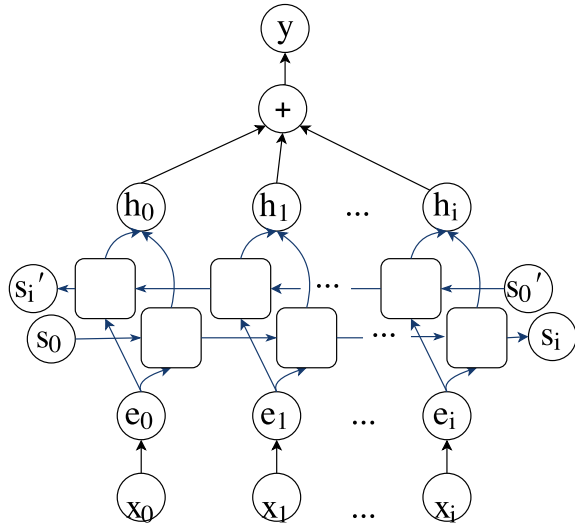
The selected model was used based on the state of the art in the classification of semantic relations used in the data set *SemEval task 8*. The model in the state of the art is a Neural Network with a Bidirectional Recurrent Gate Unit with the attention technique. This type of models have been used in sequential data such as videos, audio and text processing. The parameters used for the training were based on the proposal of [28], which consists of an embedded layer (e) and a layer of 230 neurons (h). The model was trained up to 10,000 iterations using the learning algorithm Adam [14] with initial value $\eta = 1 \times 10^{-3}$ with mini-lots of 50 instances (Fig. 6).

5.1 Domain Adaptation

Retraining of Word Embedding Space with the Words in the Specialized Corpus

In this step we retrain the neural model for word representation (*word2vec*), however, the established weights of the source domain (\mathcal{D}_S) are maintained and the new

Fig. 6 Architecture of a BGRU with attention proposed by Zhou [28]



vocabulary of the specific domain to generate the representation of the words added to the model. The main idea of performing this procedure is to reduce the difference in the distribution of data from the source domain and the target domain.

The specific domain corpus, consisting of textbooks on the domain of the object-oriented paradigm, are pre-processed to avoid introducing noise into the embedded word model. The text cleaning process was based on removing the following:

1. Empty lines.
2. Punctuation marks.
3. Elimination of non-alphanumeric characters.
4. Elimination of sentences with less than 3 words.
5. Transformation of lowercase words.

5.2 Inference of Semantic Relations in the Specific Domain Application

Automatic Preparation of the Specialized Domain Dataset

Once the representation of the embedded words is modified, now in the specific domain, it is required to create the data set with this new representation of the words. The data set used to use the learned model is made up of pairs of words, previously extracted those most likely to be representative of the new specific domain, and a sentence where the pair of words appear in the specific domain corpus.

The evaluation of the proposed methodology, the terms defined in the ontology that were considered as a model were directly introduced, which are listed below:

Scope modularity composition subclass persistence protected final interface parameters constructor superclass setter overriding getter static class hiding hierarchy return inheritance method behavior abstraction abstract state single protocol aggregation orientation modifiers private encapsulation package native typing identity synchronized overloading multiple object instance public methods attribute.

Construction of Data Set of Semantic Relationships of the Specialized Domain

After the data set of the specific domain is generated, each word that appears in the sentences, where both terms are extracted as relevant terms, are transformed into the embedded space. In this experiment the dimension of embedded space was set at 300. The maximum size of each statement was set at 60, so each instance has a dimension of $60 + 2 \times 300$.

Prediction of Semantic Relationships in the Specific Domain

Once the data set of the target domain was built, the trained model is ready to be fed. The input of the model is the representation of vectors of embedded words of each pair of key terms and the sentence that contain them. The output of the model is one of the classes with which they trained, which are hyperonym, hyponym, holonym, and meronym.

Construction of the Ontology Based in the Semantic Relations Detected Previously

The list of concepts and the relationship obtained by the model are used for the construction of the graph, which consists of the basic relationships of an ontology such as relations is-a and part-of and their inverse relations. In this first proposal, these relationships are sufficient to represent in a basic way the curricular structure of a subject in the academic context.

5.3 Evaluation of Extracted Terms and Relations

There are two task that can be evaluated, the first is the key term identification after be evaluated with the model, and the second one is to measure how many relations have been identified by the model and their relation type.

To measure how well the model learn to semantic relations, it was evaluated with precision (Eq. 5), recall (Eq. 6) and f-measure (Eq. 7) metrics.

Precision

$$Precision = \frac{tp}{tp + fp} \quad (5)$$

where, tp are the true positives and fp false positives.

Recall

$$Recall = \frac{tp}{tp + fn} \quad (6)$$

where, tp are the true positives and fn false negatives.

F-measure

$$Fmeasure = \frac{2 * precision * recall}{precision + recall} \quad (7)$$

6 Results and Discussion

The methodology was implemented, with various corpora, created from four textbooks belonging to the topic of object-oriented programming, which were automatically transformed into plain text, without applying any type of correction except those that are explicitly mentioned in the text. methodology. In addition to the four textbooks, a corpus was created through the concatenation of these documents, obtaining a corpus richer in terms of information.

In the application of the metrics for the evaluation of the results in the extraction of relations of the specific domain, only the relationships that contain relevant terms that appear in the corpus were considered, because the learning model can not determine semantic relations between concepts without the sentence that contains such relevant terms. So in the exhaustivity metric the number of relevant relationships (in the context of information retrieval) is considered as the number of elements that contain the intersection of the sets of terms identified as relevant to the target domain and the words that are contained in the corpus or text document.

Table 3 shows the results obtained when evaluating the resulting graph when applying the methodology proposed in the four textbooks and in the corpus formed of

Table 3 Results of the experiment

Document	Precision	Recall	F1
1	0.695	0.826	0.754
2	0.625	0.812	0.706
3	0.761	0.95	0.845
4	0.521	0.84	0.643
All	0.545	0.59	0.566
Mean	0.629	0.803	0.702
σ	0.1	0.131	0.106

all of them. The evaluations shown refer to the task of extracting semantic relationships only, given that the step of extracting relevant terms was made in a static way since the proposal deals with the extraction of semantic relations, however, in the previous chapter mention of some unsupervised techniques that could be applied.

In the relationship extraction task, we obtained an average precision of 62.9%, a completeness of 80.3% and a harmonic average of 70.2%, with a standard deviation of 0.1, 0.131, 0.106 respectively. A relatively wide standard deviation can be observed, however, these are presumably due to the length difference of the corpus.

7 Conclusions and Future Work

The proposed methodology shows prominent results considering that the process is fully automated and human intervention can be dispensed with. This is because once the model is trained in the source domain, it is only required to train the embedded words of the new corpus of the specific domain, identify the relevant words for the domain, and finally, construct the data set for be evaluated by the model. The necessary steps to change specific domain are automatic, which avoids a bottleneck in the process of generation of ontologies.

Each of the proposed processes can be easily improved due to its flexibility, since each phase in this methodology does not depend on the previous process but only on the structure of the required information. Therefore, as long as the data is structured correctly there would be no major complications.

This study focused on answering the following research questions:

- How well can models used in Deep Learning learn terms relationships without using NLP techniques?
- When there is a lack of training data set for each specific domain, will it be useful to apply the distant supervision technique for the representation of the relationships between domain terms?
- How feasible is it to apply Transfer of Learning in the problem of the classification of semantic relationships when the model was trained with another corpus (different data distribution)?

Due to lack of structured information of the different specific domains, it is difficult to train the models to represent that knowledge. In the domain of Ontology Learning one of the biggest problems is the effort required by people to build ontologies to represent a certain domain. Due to the low level of human intervention and that the process of manual construction of characteristics can be ignored, this methodology can be valuable as a support tool in the construction of ontologies.

The results of these experiments show that the techniques of learning transfer and distant supervision can be applied in the Ontology Learning domain. In the case of the application of Deep Learning techniques for the representation of text and classification of semantic relations, an accuracy of 62.9%, an exhaustivity of 80.3% and a measure of $F1$ of 70%, was obtained when comparing these results with the

semi-automated methods for the construction of ontologies using NLP techniques, such as in the work of [1], in whose results the following values were reported, in precision a 35%, exhaustivity was obtained 60% and F1 was 42%; it is appreciated that the results obtained in this work exceed in terms of precision and even in the harmonic average of precision and completeness, given by the metric F1, which in our case was 70%, which means an improvement of 28%.

Comparing this proposal with the proposal of Larrañaga [6], in which it has the same objective as the previous work, semi-automate the process of building the domain model of an Intelligent Tutoring System. The main idea of his proposal is based on the extraction of relationships through patterns between words, making him dependent on language and possibly on domain. In this work we take advantage of the structure contained in the table of contents of textbooks, in which it is not specified if the identification and extraction of this is automatically or was built by a human. The author reports a completeness in the relations *part-of* (holonyms and meronyms) of 55.81%, in the taxonomic relationships (hyperonyms and hyponyms) a value of 56.1% and 30.30% is reported for the precision and recall respectively.

The proposed methodology, is based entirely on unsupervised methods where they have proven to be non-language dependent, as are methods for transforming words into the embedded space *word2vec* [19]. Although the proposed methodologies extract diverse relationships, in addition to those contemplated in this paper, it is possible to equate and in some cases overcome their results, which is notable because the methodology proposed does not use language-dependent NLP techniques. word tagging, identification of word patterns mostly used in certain relationships; even so, results were obtained according to the state of the art. In addition to the lack of use of NLP techniques, the structure of the corpus was also not taken into account, that is to say, sections frequently found in textbooks, such as the title, were not contemplated.

The main contributions of this work are the following:

- Methodology to automate the process of building ontologies of a specific domain through non-structured corpora.
- Creation of data set of the generic domain through the technique of distant supervision between the corpus of Wikipedia and WordNet.
- Ontology Learning Technique by means of classification of relations between domain terms in the space of embedded words.

Acknowledgements This research was supported/partially supported by MyDCI (Maestría y Doctorado en Ciencias e Ingeniería).

References

1. Atapattu, T., Falkner, K., Falkner, N.: Automated extraction of semantic concepts from semi-structured data: supporting computer-based education through the analysis of lecture notes. In: International Conference on Database and Expert Systems Applications, pp. 161–175 (2012)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *J. Mach. Learn. Res.* **3**, 993–1022 (2003)

3. Carballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL'99, pp. 120–126. Association for Computational Linguistics, Stroudsburg, PA, USA (1999)
4. Chakraborty, S., Roy, D., Basu, A.: TMRF e-book development of knowledge based intelligent tutoring system. *Sajja & Akerkar* **1**, 74–100 (2010)
5. Colace, F., De Santo, M., Greco, L., Amato, F., Moscato, V., Picariello, A.: Terminological ontology learning and population using latent Dirichlet allocation. *J. Vis. Lang. Comput.* **25**(6), 818–826 (2014)
6. Conde, A., Larrañaga, M., Calvo, I., Elorriaga, J., Arruarte, A.: Automatic generation of the domain module from electronic textbooks: method and validation. *IEEE Trans. Knowl. Data Eng.* **26**(1), 69–82 (2014)
7. dos Santos, C.N., Xiang, B., Zhou, B.: Classifying relations by ranking with convolutional neural networks. *Acl-2015* **3**, 626–634 (2015)
8. Fan, M., Zhou, Q., Abel, A., Zheng, T.F., Grishman, R.: Probabilistic belief embedding for large-scale knowledge population. *Cogn. Comput.* **8**(6), 1087–1102 (2016)
9. Gantayat, N., Iyer, S.: Automated building of domain ontologies from lecture notes in courseware. In: Proceedings—IEEE International Conference on Technology for Education, T4E 2011, pp. 89–95 (2011)
10. Gliğora, M., Jakupovic, A.: A prevalence trend of characteristics of intelligent and adaptive hypermedia e-learning systems (2015)
11. Hendrickx, I., Kim, S.N., Kozareva, Z., Nakov, P., Séaghdha, D., Padó, S., Pennacchiotti, M., Romano, L., Szpakowicz, S.: SemEval-2010 Task 8: multi-way classification of semantic relations between pairs of nominals. In: Computational Linguistics, Number June 2009 in DEW'09, pp. 94–99. Association for Computational Linguistics, Stroudsburg, PA, USA (2010)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
13. Isir, R.M., Canada, G.H.: Using ontological engineering to overcome common AI-ED problems. *J. Artif. Intell. Educ.* **11**, 107–121 (2000)
14. Kingma, D.P., Adam, J.B.: Method for stochastic optimization. *CoRR* (2014). abs/1412.6
15. Komninos, A.: Dependency based embeddings for sentence classification tasks. In: *Naacl 2016*, pp. 1490–1500 (2016)
16. Landauer, T.K., Dumais, S.T.: A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychol. Rev.* **104**(2), 211–240 (1997)
17. Lecun, Y., Bengio, Y., Hinton, G.: Deep learning (2015)
18. Lin, Y., Shen, S., Liu, Z., Luan, H., Sun, M.: Neural relation extraction with selective attention over instances. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, vol. 1: Long Papers, pp. 2124–2133 (2016)
19. Mikolov, T., Corrado, G., Chen, K., Dean, J.: Efficient estimation of word representations in vector space. In: Proceedings of the International Conference on Learning Representations (ICLR 2013), pp. 1–12 (2013)
20. Navarro-Almanza, R., Licea, G., Juárez-Ramírez, R., Mendoza, O.: Semantic Capture analysis in word embedding vectors using convolutional neural network. In: Rocha, A., Correia, A.M., Adeli, H., Reis, L.P., Costanzo, S. (eds.) *Recent Advances in Information Systems and Technologies*, vol. 1, pp. 106–114. Springer, Cham (2017)
21. Pan, W., Zhong, F., Yang, Q.: Transfer learning for text mining. In: Aggarwal, C.C., Zhai, C. (eds.) *Mining Text Data*, pp. 223–257. Springer, Boston, MA, USA (2012)
22. Qin, P., Weiran, X., Guo, J.: An empirical convolutional neural network approach for semantic relation classification. *Neurocomputing* **190**, 1–9 (2016)
23. Ramírez-Noriega, A., Juárez-Ramírez, R., Jiménez, S., Inzunza, S., Navarro, R., López-Martínez, J.: An ontology of the object orientation for intelligent tutoring systems. In: *2017 5th International Conference in Software Engineering Research and Innovation*, pp. 163–170 (2017)

24. Rios-Alvarado, A.B., Lopez-Arevalo, I., Sosa-Sosa, V.J.: Learning concept hierarchies from textual resources for ontologies construction. *Expert Syst. Appl.* **40**(15), 5907–5915 (2013)
25. Xiong, S., Ji, D.: Exploiting flexible-constrained K-means clustering with word embedding for aspect-pharse grouping. *Inf. Sci.* **367–368**, 689–699 (2016)
26. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1753–1762 (2015)
27. Zhao, H., Lu, Z., Poupart, P.: Self-adaptive hierarchical sentence model. In: *IJCAI International Joint Conference on Artificial Intelligence*, January 2015, pp. 4069–4076 (2015)
28. Zhou, P., Shi, W., Tian, J., Qi, Z., Li, B., Hao, H., Xu, B.: Attention-based bidirectional long short-term memory networks for relation classification. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 2, Short Papers, pp. 207–212 (2016)

Implementation of a Multicriteria Analysis Model to Determine Anthropometric Characteristics of an Optimal Helmet of an Italian Scooter



Josué Cuevas, Alberto Ochoa, Juan Luis Hernandez, José Mejia, Liliana Avelar and Boris Mederos

Abstract At the beginning of the sixties, the Italian scooters became popular in Italy, an age that was considered unimportant the safety of people. However, safety standards are now stronger, and helmet use is mandatory in most countries. That is why this paper try to analyze in greater detail the characteristics associated with the anthropometric measurements of a representative sample of a broader society for determining the ideal characteristics of a helmet associated with this type of vehicle. For this reason and considering the relevance of our study to a Smart City, we will use a multicriteria analysis and an intelligent data analysis in order to understand much better the ideal and suitable measurements to be able to design a helmet for an Italian scooter.

Keywords Italian scooter · Optimal and ideal helmet · Smart city

1 Introduction

An accepted definition for an Italian scooter is: “(1) has a platform for the operator’s feet or has integrated footrests, and (2) has a step through architecture, meaning that the part of the vehicle forward of the operator’s seat and between the legs of an operator seated in the riding position, is lower in height than the operator’s seat” [1]. Nevertheless, because of the similarly to motorcycles, regulations in some countries do not differentiate between motorcycles and scooters [2]. As the time pass through, terms have changed, but in the first apparitions of scooters in the market, “the gendered connotations of the term ‘scooter’ were deliberately accentuated in the design and marketing of the original Vespa and Lambretta scooters in Italy. The design made concessions to the rider’s comfort, convenience and vanity (the enveloping of machine parts meant that the scooterist was not obliged to wear specialist protective clothing)” [3]. But there can exist a lot of accidents that can occur while driving a

J. Cuevas · A. Ochoa (✉) · J. L. Hernandez · J. Mejia · L. Avelar · B. Mederos
Juarez City University, Ciudad Juárez, Mexico
e-mail: alberto.ochoa@uacj.mx

© Springer Nature Switzerland AG 2020

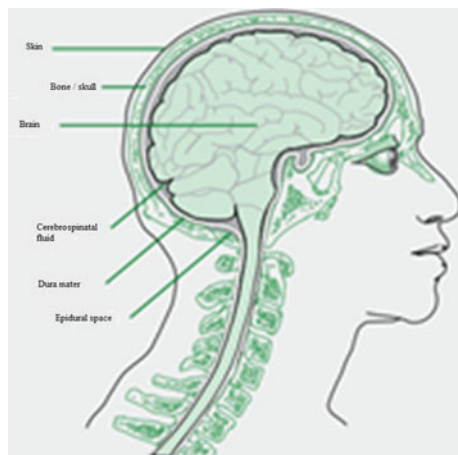
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_51

757

scooter as with any other PTW; “top of the clothing shopping list for any aspiring scooterist is a decent and appropriate helmet. As well as being a legal requirement in most countries, they make sound safety sense” [4]. But, anyway, every single scooter driver must have in mind that if a helmet it not adequately secured to the head it could not provide the necessary protection in case of accident [5].

That is why it is necessary to adapt the helmets as much as possible to the morphology of the human head. Therefore, it’s important to recognize some basic anatomical information about the head (Fig. 1). The brain is contained within the skull and sits on its base from where the spinal cord is distributed to the rest of the body. There exist a tough tissue called dura that separate the skull from the brain, and between brain and dura is a cerebrospinal fluid that protect the brain from mechanical shocks. The brain is suspended in within this fluid and only can move about 1 mm in any direction (World Health Organization, n.d.). The cerebrospinal fluid cannot avoid a severe accident because since the brain doesn’t fill the entire skull cavity, a person that falls from a fast-moving vehicle, such as a italian scooter, and hitting a hard surface can cause severe bruising and swelling of the delicate brain tissues. Also skull fractures and bleeding could result from sheared vessels around and inside the brain. Recent research suggests that safety helmets, under certain conditions, could reduce the risk of head injury by up to 90% [6]. That takes us to the fact that “a helmet is the most important protection against severe injuries and deaths. It dramatically reduces the risk of being killed or severely injured and should be worn by all riders and passengers of motorcycles and mopeds” [7]. In this study, we develop a multicriteria analysis model to determine anthropometric characteristics of an optimal helmet of an Italian Scooter, for this end, we have used a database with 15 cranial measures with the intention to find out the most information as it is possible from 130 instances. Those instances correspond to potential Italian scooters drivers. The purpose of this research is to analyze the general morphological characteristics that describe this sample.

Fig. 1 Head structure



2 Methodology Applied

For this research the WEKA software developed by the Waikato New Zealand University was used, since it contains different machine learning algorithms that took an important role for patterns discovery. In order to do that, we had to follow a data mining sequence. In first place, we clean out the database to create a minable set. This was crucial due to the needing to avoid an overfitting analysis, as well to help discover the most meaningful attributes. There are a lot of ways to accomplish this. But, due to the need to make a unisex helmet, we'll look for the attributes that are capable to distinguish the gender of a person and watch their behavior to get an analysis. Before getting a database more significantly between genders, a Principal Components Analysis (PCA) was made for gain insight about the correlation between every single cranium measurement and be able to absolutely discard useless attributes for further researches. After that, a classification experiment took place in the Weka Experimenter tool. The running settings for the experiment were configured for a ten repetitions train/test percentage split (data randomized) with 70% of the data used for training. This was implemented to observe the behavior of seven different algorithms working on the dataset and then, use the most efficient algorithm to get running the Weka's Attribute Evaluator to discard attributes that are relatively unable to differentiate between men and women. Then, the remaining attributes were used to make a clustering analysis. First, the clustering algorithm was run over the whole database, thus the general description about the sample could be found. Finally, a more detailed clustering was made over the man instances and the women instances with the intention to observe the most distinctive features.

3 Results of Our Multivariable Analysis

The initial data attributes are the following (the number at the left of each attribute will be used to represent that attribute throughout this research):

1. Age
2. Weight
3. Height
4. Head width (Eu-Eu)
5. Head length (G-Op)
6. Nasal width (Al-Al)
7. Nasal height (N-Ns)
8. Face height (N-Gn)
9. Forehead width (Ft-Ft)
10. Biocular distance (Ex-Ex)
11. Inter-canthal distance (En-En)
12. Face width (Zy-Zy)
13. Head circumference (G-Op)

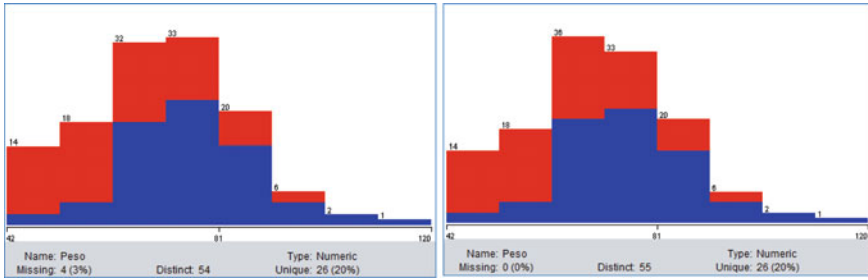


Fig. 2 Use of replace missing values filter on Weka

- 14. G-V-Op length
- 15. Eu-V-Eu length
- 16. V-Gn length
- 17. V-Po length
- 18. Head height(V-N)
- 19. Gender (Database class).

To obtain a minable set, there were some actions to make in order to start the beginning of the data mining sequence. There were some missing values that has been replaced by the Weka filter *ReplaceMissingValues* (Fig. 2). It consists in impute new values that does not affect statistically the database.

Additionally, the “Place” attribute was thrower away, since being highly unbalanced it does not add any practical information to the final research. It will be assumed that the sample represent Ciudad Juárez citizens. Consequently, another cleaning filter was carried out. Instances that were found with extreme or outlier values were eliminated with the firm intention of reducing the noise of the data that looked severely as erroneous data captures. After getting a cleaner minable sight, the Principal Component Analysis (PCA) took place as another kind of filter. As the data does is not homogenous and it has different units of measurement (meters, centimeters, years and kilograms) and even the attributes that has the same units are not in the same scale, e.g., the biocular distance and the head length are in centimeter, but the range of the last one is relatively higher than the biocular distance, the PCA was configured to standardize the variables. When executing the analysis, thanks to the output information provided by Weka it was observed with the correlation matrix that the age was not correlated with any other attribute so, this attribute was eliminated and a PCA was performed again. The correlation matrix is shown in the Fig. 3. The light green cells denote a low correlation coefficient, while the darker ones have a strong correlation index.

The results are shown in Table 1. The magnitudes of the coefficients of each eigenvector give the contributions of each attribute to that component. The number next to each weight represent the number of the attribute accordingly to the list showed previously. The number of attributes selected to be displayed in the table

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1		0.2	0.1	0.0	0.1	0.0	0.0	0.1	0.0	0.2	0.0	0.0	0.1	0.1	0.0	0.1	0.1	0.0
2	0.2		0.6	0.5	0.4	0.4	0.1	0.5	0.2	0.4	0.3	0.4	0.5	0.3	0.3	0.6	0.3	0.3
3	0.1	0.6		0.5	0.5	0.2	0.4	0.6	0.3	0.4	0.3	0.4	0.5	0.5	0.4	0.6	0.3	0.4
4	0.0	0.5	0.5		0.3	0.4	0.1	0.4	0.4	0.5	0.3	0.7	0.6	0.4	0.5	0.5	0.4	0.4
5	0.1	0.4	0.5	0.3		0.3	0.4	0.5	0.3	0.5	0.3	0.4	0.8	0.6	0.5	0.6	0.2	0.4
6	0.0	0.4	0.2	0.4	0.3		0.3	0.4	0.3	0.5	0.3	0.4	0.4	0.1	0.2	0.5	0.3	0.3
7	0.0	0.1	0.4	0.1	0.4	0.3		0.7	0.3	0.3	0.3	0.3	0.3	0.2	0.2	0.5	0.1	0.2
8	0.1	0.5	0.6	0.4	0.5	0.4	0.7		0.4	0.5	0.3	0.5	0.5	0.4	0.4	0.7	0.3	0.3
9	0.0	0.2	0.3	0.4	0.3	0.3	0.3	0.4		0.4	0.3	0.5	0.5	0.4	0.5	0.4	0.0	0.4
10	0.2	0.4	0.4	0.5	0.5	0.5	0.3	0.5	0.4		0.5	0.7	0.6	0.4	0.4	0.5	0.3	0.3
11	0.0	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.3	0.5		0.4	0.4	0.2	0.2	0.3	0.1	0.2
12	0.0	0.4	0.4	0.7	0.4	0.4	0.3	0.5	0.5	0.7	0.4		0.6	0.4	0.5	0.5	0.2	0.4
13	0.1	0.5	0.5	0.6	0.8	0.4	0.3	0.5	0.5	0.6	0.4	0.6		0.7	0.7	0.6	0.3	0.5
14	0.1	0.3	0.5	0.4	0.6	0.1	0.2	0.4	0.4	0.4	0.2	0.4	0.7		0.8	0.5	0.3	0.5
15	0.0	0.3	0.4	0.5	0.5	0.2	0.2	0.4	0.5	0.4	0.2	0.5	0.7	0.8		0.6	0.4	0.6
16	0.1	0.6	0.6	0.5	0.6	0.5	0.5	0.7	0.4	0.5	0.3	0.5	0.6	0.5	0.6		0.5	0.7
17	0.1	0.3	0.3	0.4	0.2	0.3	0.1	0.3	0.0	0.3	0.1	0.2	0.3	0.3	0.4	0.5		0.4
18	0.0	0.3	0.4	0.4	0.4	0.3	0.2	0.3	0.4	0.3	0.2	0.4	0.5	0.5	0.6	0.7	0.4	
18	0.0	0.3	0.4	0.4	0.4	0.3	0.2	0.3	0.4	0.3	0.2	0.4	0.5	0.5	0.6	0.7	0.4	

Fig. 3 Correlation matrix

Table 1 Principal components analysis results

	Eigenvalue	Proportion	Cumulative	Weights (eigenvector)
PC1	7.71678	0.45393	0.45393	-0.307 [13] - 0.298 [16] - 0.272 [12] - 0.269 [8] - 0.262 [10]
PC2	1.55603	0.09153	0.54546	-0.43 [14] - 0.413 [15]
PC3	1.21961	0.07174	0.6172	-0.487 [9] + 0.382 [17] - 0.353 [12]
PC4	1.2037	0.07081	0.68801	0.507 [7] - 0.424 [17] - 0.409 [4] - 0
PC5	0.90434	0.0532	0.7412	-0.514 [2] + 0.369[17] + 0.3 [18] +
PC6	0.75875	0.04463	0.78584	0.511 [6] + 0.476 [5] - 0.471 [3]

Table 2 Most important attributes for each eigenvector

PC1	PC2	PC3	PC4	PC5	PC6
Head circumference G-Op	G-V-Op	Ft-Ft	N-Ns	Weight	Al-Al
V-Gn	Eu-V-Eu	V-Po	V-Po	V-Po	Head length G-Op
Zy-Zy		Zy-Zy	Eu-Eu		Height
N-Gn					
Ex-Ex					

are those that had relatively high coefficients magnitudes. Because the data were standardized, they do not depend on the variances of the corresponding attributes.

The components that the Table 1 shown are only those that cover the 75% of the standardized data variation. This leads us to representative information and, as we can see in Table 2, which shows in more detail each attribute that correspond to one of the six components, the first component shows that the head circumference G-Op, the V-Gn length and the biocular distance Ex-Ex decrease together and conform the most important principal component of the data (it represents 45.39% of the data). Later in this paper will be explained why we remark those specific attributes. Also, other important attributes are shown with a bold emphasis.

Attribute Selection

Since the instances per attributes quotient ($122/18 = 6.77$) is too small, it was necessary to delete some attributes from the analysis. The way to do this is by using an attribute selector. As it was discussed, we will focus on the attributes that the difference between anthropometric is relatively high. Nevertheless, Weka’s Wrapper Subset Evaluator need a classifier algorithm to work out. An experiment was designed to observe the different type of algorithms running in the data. Different means were used to compare the quality of classification of each one as it is observed in Table 3. They are described in the following list:

- Percent correct. Instances that were classified successfully accordingly to the class (sex).
- Kappa statistic. It indicates how much better a classifier is performing over the performance of a classifier that simply guesses at random according to the frequency of each class. [8] provide a way to interpret the values: 0 or less is indicating no agreement 0–0.20 as slight; 0.21–0.40 as fair; 0.41–0.60 as moderate; 0.61–0.80 as substantial; 0.81–1 as almost perfect agreement.
- Mean absolute error. MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It’s the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight.
- Relative absolute error. The relative absolute error (RAE) can be compared between models whose errors are measured in the different units.

Table 3 Experiment design for classification algorithms

Correct percentage	Kappa statistic	Mean absolute error	Relative absolute error	F measure	MCC	Area under ROC
<i>Naïve Bayes</i>						
0.83	0.66	0.17	0.34	0.84	0.67	0.92
0.06	0.13	0.06	0.12	0.05	0.13	0.05
<i>J48</i>						
0.76	0.51	0.25	0.50	0.75	0.52	0.78
0.07	0.14	0.07	0.14	0.08	0.14	0.09
<i>Logistic</i>						
0.81	0.63	0.21	0.42	0.83	0.63	0.87
0.09	0.17	0.06	0.13	0.08	0.17	0.07
<i>OneR</i>						
0.77	0.54	0.23	0.45	0.79	0.55	0.77
0.06	0.12	0.06	0.12	0.05	0.12	0.06
<i>ZeroR</i>						
0.53	0.00	0.50	1.00	0.69	NaN	0.50
0.01	0.00	0.00	0.00	0.01	NaN	0.00
<i>IBk</i>						
0.80	0.58	0.22	0.44	0.83	0.60	0.85
0.05	0.10	0.04	0.08	0.04	0.09	0.05
<i>Part</i>						
0.74	0.48	0.26	0.53	0.74	0.49	0.76
0.08	0.15	0.07	0.15	0.08	0.15	0.08

- F measure. Is a measure of a test’s accuracy and is defined as the weighted harmonic mean of the precision which recall of this innovative test to analyze this society in a Smart City.
- MCC. The Matthews Correlation Coefficient has a range of -1 to 1 where -1 indicates a completely wrong binary classifier while 1 indicates a completely correct binary classifier. Using the MCC allows one to gauge how well their classification model/function is performing,
- Area under ROOC. The accuracy of the test depends on how well the test separates the group being tested into the class attribute in question. A rough guide for classifying the accuracy of a diagnostic test is the traditional academic point system: $90-100 =$ excellent; $80-90 =$ good (B); $70-80 =$ fair (C); $60-70 =$ poor (D); $50-60 =$ fail (F).

The first row of Table 3 shows de index magnitude while the second row indicates the standard deviation. As we can see, the tested algorithms do not differ more than 5% in almost all the indexes. Thus, we are not able to stablish what algorithm performs

better. Probably the Naïve Bayes algorithm showed better coefficients values, but anything that would categorically discard any other classifier. We faced a multipaths problem-solving scenario. Nevertheless, it is important to observe that the algorithm that had de worst performance was ZeroR, which departs from trivial suppositions and leads us to know that the other attributes are meaningful to make the women/man distinction since they had a higher values of kappa statistic and correct classified instances percentage.

Therefore, the chosen algorithm was the Logistic Regression which allows a flexibility decision since it establishes a threshold value to determine whether we are dealing with a male or female instance. Provided that information, Weka’s Wrapper-SubsetEval attribute selector could perfectly work with the logistic classifier. Also, it was set to perform a 10 fold-cross-validation without using the BestFirst as search method and configurated to work in backward direction. The output information shows how many times a specific attribute plays an important role each time that the logistic model was performed using different seed numbers. The results appear bellow in Table 4.

Height is, without any doubts, the most determining factor as it appeared in every single fold. However, this is not high significant for the purpose of this paper. However, the attribute will be used to refine the clustering algorithms. In order to implement the last ones, the criterion to reduce data dimensions was to discard all the attributes that had five or less folds.

Clustering

Table 4 Wrapper subset evaluator results

Num. of folds	Attribute
10	Height
9	Biocular distance (Ex-Ex)
9	G-V-Op length
8	Nasal width (Al-Al)
7	V-Gn length
6	Nasal height (N-Ns)
6	Forehead width (Ft-Ft)
6	Head circumference (G-Op)
5	Face height (N-Gn)
5	Intercanthal distance (En-En)
5	Face width (Zy-Zy)
4	Head length (G-Op)
4	Eu-V-Eu length
3	Weight
3	Head width (Eu-Eu)
3	Head height (V-N)
2	V-Po length

Having the attributes that were considered most relevant for our tasks, the new minable sight was submitted to a grouping analysis. To do this, the data set was split into two subsets: 57 instances corresponding to women and 65 corresponding to men. This with the firm intention to find out patterns and even other subsets that could gave a deeper description. The algorithm used was Kmeans, however, two variants of this method were used: SimpleKMeans and X-Means. Its way of operating is practically the same, with the only difference that X-means detects the number of clusters to consider automatically accordingly to the data itself, while the SimpleKMeans must receive as a parameter the number of desired clusters before being executed. The general operation of KMeans is described in the Fig. 4, using different estimators in diverse kind of design of experiments as are described in specific types in [9–12].

Therefore, to take advantage of these functionalities, the XMeans was executed in each subset to determine the number of clusters appropriate in each case.

The algorithm results allow us to identify two groups in each subset, that is, four groups that will be labeled as group A, B C and D, where the first two correspond to the clusters found for the women’s group, while the other two correspond to the man’s subset as it the Fig. 5 shows. The location of each of the centroids of the clusters are shown in Appendix B. Each sex is in different circumstances. In the men’s case, there is a majority group (made up of 48 people) and a minority group

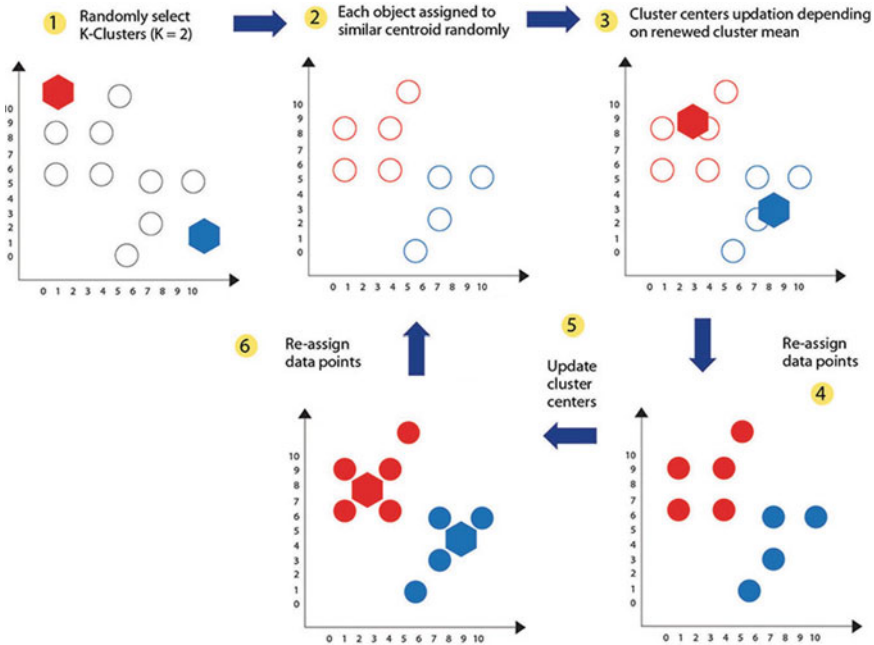


Fig. 4 KMEANS functionality

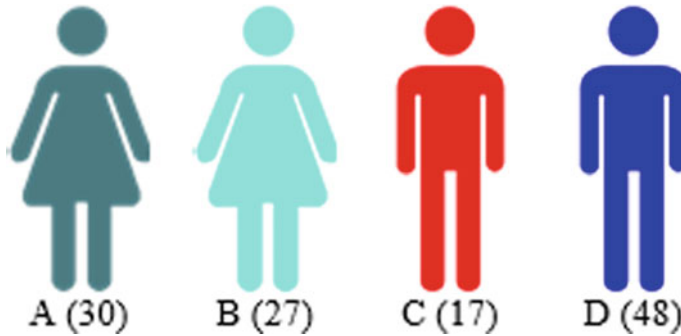


Fig. 5 Clustering resulting groups

(of 17 people), while with women, the situation is more balanced, since there is a group of 30 instances and another of 27.

In Fig. 6a–c, the centroids locations were normalized in order to compare each of the attributes between the different groups. One point to have in mind is that the minority men’s group is considerably above the average. This have leads us to underestimate group C in the analysis.

Lastly, we ran a discretize filter to observe the region of each attribute where most people are. The rows in bold represents at least 75% of the four groups (Table 5).

Lastly, we ran a discretize filter to observe the region of each attribute where most people are. The rows in bold represents at least 75% of the four groups.

Knowing that ranges, we can run again a cluster analysis against the data (ignoring the class) and determine which are the measures that fit better with the sample population. The centroid locations are in Table 6.

Data Projection

To confirm that the data will not present variations that go beyond the limits given by the previously presented analysis, a projection of the data was made. For this, we used the Forecasting tool, an additional package offered by Weka. An artificial time stamp and an AdditiveRegression base learner were configured in order to get the projection. The results are shown in the GUI’s Projection section.

4 Conclusions and Future Research

Although the curse of dimensionality transcends big data scenarios, the ideas that involve this concept cannot be dismissed. That is why, as mentioned in this paper, we did not have enough data to build a complete anthropometric model. We found measures that can be the starting point to design the helmet. A greater number of instances are required to make more forceful affirmations. This follows the comfort need: “good fit is essential. Too small and a helmet will become very uncomfortable,

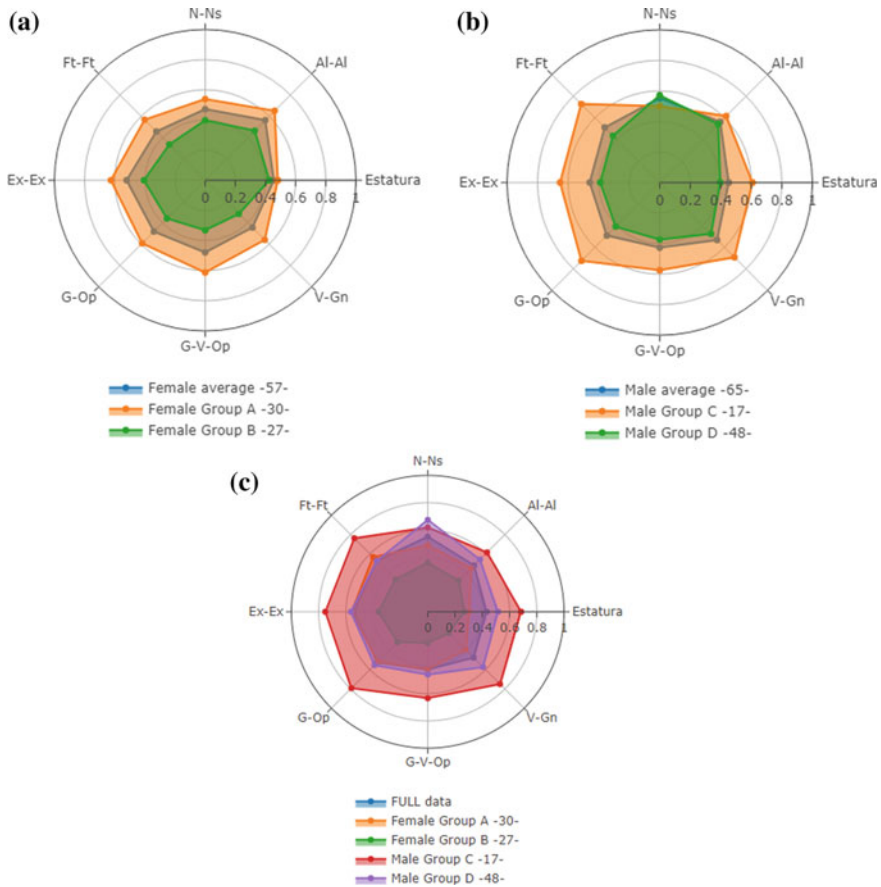


Fig. 6 Female clusters (a) man cluster (b) full clusters radial charts (c)

distracting your attention from the task in hand. Too big and it won't give adequate protection. The correct fit should exert light pressure evenly over your head with no excessive pressure at any point" [13]. However, we include for each table with the centroids that describe each single group that were found. It is a reality: "all countries should have and enforce a helmet law that makes wearing a helmet obligatory for riders of all powered two-wheelers" [14]. In order to be able to design a helmet there are some things to have in mind: "according to current standards, the performance of a helmet is only tested against radial impact which is not commonly seen in real accidents. Some studies about helmet design have been published but those helmets have been tested for only a few loading conditions" [4, 15]. Also, it's important to that "the shell and the inner liner are the key protective components of a helmet. The outer shell is there to resist abrasion and prevent objects from penetrating in the event of an accident. Shells at the lower end of the market are usually injection-molded polycarbonate, although this is sometimes found in some higher-specification

Table 5 Measurements ranges

Al-Al	Perc.	Acum.	N-Ns	Perc.	Acum.	Ft-Ft	Perc.	Acum.
3.09–3.22	25.5	25.5	5.06–5.27	18.1	18.1	10.15–10.64	21.9	21.9
3.48–3.61	22.9	48.3	4.85–5.06	15.2	33.3	10.64–11.13	18.1	40.0
3.35–3.48	16.2	64.5	5.27–5.48	14.3	47.6	9.17–9.66	17.1	57.1
3.22–3.35	14.3	78.8	4.64–4.85	13.3	61.0	9.66–10.15	17.1	74.3
2.96–3.09	7.6	86.4	4.43–4.64	12.4	73.3	11.13–11.62	8.6	82.9
2.83–2.96	4.8	91.2	4.22–4.43	9.5	82.9	8.19–8.68	7.6	90.5
min–2.83	3.8	95.0	5.48–5.69	6.7	89.5	8.68–9.17	5.7	96.2
3.87–max	2.9	97.9	5.69–max	4.8	94.3	12.11–max	1.9	98.1
3.61–3.74	1.9	99.8	min–4.01	2.9	97.1	min–8.19	1.0	99.0
3.74–3.87	0.0	99.8	4.01–4.22	2.9	100.0	11.62–12.11	1.0	100.0
Ex-Ex	Perc.	Acum.	G-Op	Perc.	Acum.	G-V-Op	Perc.	Acum.
9.95–10.2	26.7	26.7	55.45–56.38	28.6	28.6	31.68–32.61	21.0	21.0
10.2–10.45	17.1	43.8	56.38–57.31	21.0	49.5	29.82–30.75	18.1	39.0
9.7–9.95	15.2	59.0	54.52–55.45	13.3	62.9	28.89–29.82	15.2	54.3
9.45–9.7	12.4	71.4	53.59–54.52	9.5	72.4	30.75–31.68	12.4	66.7
10.45–10.7	11.4	82.9	57.31–58.24	9.5	81.9	32.61–33.54	8.6	75.2
10.7–10.95	7.6	90.5	52.66–53.59	6.7	88.6	27.96–28.89	7.6	82.9
9.2–9.45	3.8	94.3	58.24–59.17	5.7	94.3	27.03–27.96	5.7	88.6
10.95–max	2.9	97.1	59.17–max	2.9	97.1	33.54–34.47	4.8	93.3
8.95–9.2	1.9	99.0	51.73–52.66	1.9	99.0	34.47–max	3.8	97.1
min–8.95	1.0	100.0	min–51.73	1.0	100.0	min–27.03	2.9	100.0
			V-Gn	Perc.	Acum.			
			20.66–21.2	15.2	15.2			
			21.2–21.74	15.2	30.5			
			21.74–22.28	13.3	43.8			
			20.12–20.66	10.5	54.3			
			22.82–23.36	9.5	63.8			
			19.04–19.58	8.6	72.4			
			22.28–22.82	7.6	80.0			
			min–19.04	6.7	86.7			
			19.58–20.12	6.7	93.3			
			23.36–max	6.7	100.0			

helmets. More expensive helmets tend to use laminates of materials such as glassfibre, carbon fibre and Kevlar” [7, 8, 16, 17]. The way as we recommend for dealing with these criteria in order to design the helmet is described by the hierarchically tree shown in Fig. 7.

Table 6 Characteristics measurements

Attribute	1	2	3
Height	1.6222	1.7621	1.6639
Al-Al	3.155	3.4294	3.3355
N-Ns	4.6175	5.3324	4.9032
Ft-Ft	9.65	10.3235	10.4419
Ex-Ex	9.775	10.3824	10.3097
G-Op (Head circumference)	54.5675	56.7618	56.7032
G-V-Op	29.4475	31.3324	32.1516
V-Gn	19.9675	22.4147	21.6677

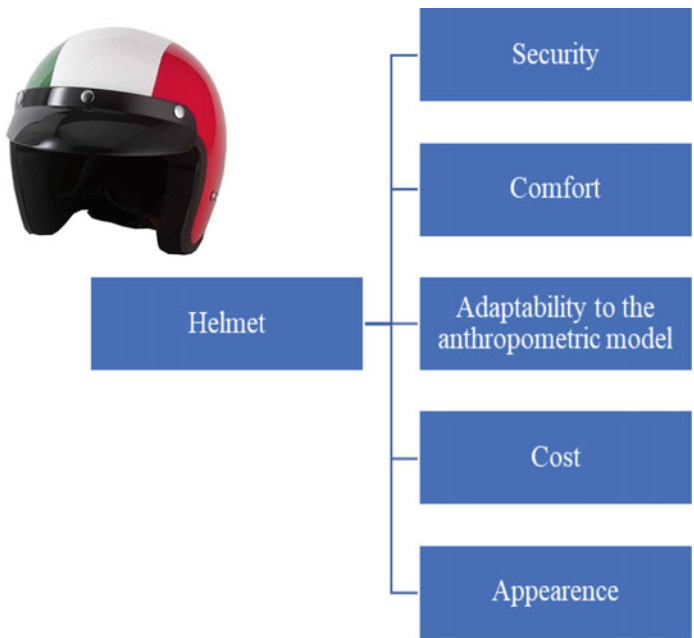


Fig. 7 Hierarchicly tree for helmet design criteria

Finally we propose two models in 3D printer named Kelpian and Kaylon-1 to improve th security of people in a Smart City.

References

1. Bernety, H.M., Venkatesh, S., Schurig, D.: Performance analysis of a helmet-based radar system for impact prediction. *IEEE Access* **6**, 75124–75131 (2018)

2. Chen, Y.-W., Cheng, C.-Y., Li, S.-F., Yu, C.-H.: Location optimization for multiple types of charging stations for electric scooters. *Appl. Soft Comput.* **67**, 519–528 (2018)
3. Better Health Channel. (n.d.): Scooters and Child Safety. Retrieved from Better Health Channel. <https://www.betterhealth.vic.gov.au/health/HealthyLiving/scooters-and-child-safety?viewAsPdf=true>
4. Seeley, A.: *The Scooter Book*. Haynes Publishing, Sparkford (2004)
5. Chen, L.-B., Chang, W.-J., Su, J.-P., Chen, Y.-R.: i-Helmet: an intelligent motorcycle helmet for rear big truck/bus intimation and collision avoidance. *ICCE 2018*, pp. 1–2 (2018)
6. Fahlstedt, M., Halldin, P., Kleiven, S.: Importance of the bicycle helmet design and material for the outcome in bicycle accidents (2014)
7. OECD/ITF: *Improving Safety for Motorcycle, Scooter and Moped Riders*. OECD Publishing, Paris (2015)
8. Landis, J.R., Koch, G.G.: The measurement of observer agreement for categorical data. *Biometrics* **33**(1), 159–174 (1977)
9. Bai, D., Quan, Q., Tang, D., Deng, Z.: Design and experiments of a novel rotary piezoelectric actuator using longitudinal-torsional convertors. *IEEE Access* **7**, 22186–22195 (2019)
10. Blondet, G., Le Duigou, J., Boudaoud, N.: A knowledge-based system for numerical design of experiments processes in mechanical engineering. *Expert Syst. Appl.* **122**, 289–302 (2019)
11. Sun, Y.-G., Xu, J.-Q., Chen, C., Lin, G.-B.: Fuzzy H_∞ robust control for magnetic levitation system of maglev vehicles based on T-S fuzzy model: Design and experiments. *J. Intell. Fuzzy Syst.* **36**(2), 911–922 (2019)
12. Refael, G., Degani, A.: A single-actuated swimming robot: design, modelling, and experiments. *J. Intell. Rob. Syst.* **94**(2), 471–489 (2019)
13. Jeong, M., Lee, H., Bae, M., Shin, D.-B., Lim, S.-H., Lee, K.B.: Development and application of the smart helmet for disaster and safety. *ICTC 2018*, pp. 1084–1089 (2018)
14. Revindran, R., Vijayaraghavan, H., Huang, M.-Y.: Smart helmets for safety in mining industry. *ICACCI*, pp. 217–221 (2018)
15. e Silva, R.R.V., Aires, K.R.T., de Veras, M.S.: Detection of helmets on motorcyclists. *Multimedia Tools Appl.* **77**(5), 5659–5683 (2018)
16. World Health Organization. (n.d.). Why are Helmets Needed? Retrieved from World Health Organization. http://www.who.int/roadsafety/projects/manuals/helmet_manual/1-Why.pdf
17. Hebdige, D.: Travelling light: one route into material culture. *RAIN* **59**, 11–13 (1983). <https://doi.org/10.2307/3033466>

Improving Segmentation of Liver Tumors Using Deep Learning



José Mejía, Alberto Ochoa and Boris Mederos

Abstract Liver tumor segmentation from computed tomography images is an essential task for the automated diagnosis and treatment of liver cancer. However, such task is difficult due to the variability of morphologies, diffuse boundaries, heterogeneous densities, and sizes of the lesions. In this work we develop a new system designed for the segmentation of tumors from images acquired by computed tomography, the proposed system uses a network based on convolutional neural networks (CNN). The results are compared with a segmentation carried out by medical experts.

Keywords Segmentation liver tumor · Pattern recognition · Deep learning

1 Introduction

One of the most common methods for the acquisition of medical images of the liver is the computed tomography (CT) [1], however, such method has some drawbacks such as lack of contrast, noise and artifacts in the image. This means that the specialist could find problems for the analysis of the images, which implies the possibility of an erroneous analysis. In addition, regardless of the image modality used, there are multiple sources of error and challenges for liver segmentation, since several organs and structures show density and gray scale similar to liver. For example, the spleen and pancreas show a similar level of gray in the CT scan data. In addition, the shape of the liver is too large and the liver tissue itself often appears different in the images. This may be due to liver disease such as liver tissue scarring [2]. In clinical practice the abnormalities present in liver tissue are diagnosed mainly based on the experience and skill of the medical specialist, this process, due to the above factors, can be slow and monotonous and could cause fatigue to the specialist affecting the diagnosis [2]. At present there is a tendency to develop solutions to these problems by detecting injuries using automated image processing techniques [3–5]. In the standard clinical method for liver segmentation, the doctor manually draws the

J. Mejía · A. Ochoa (✉) · B. Mederos
Juarez City University, Ciudad Juárez, Mexico
e-mail: alberto.ochoa@uacj.mx

© Springer Nature Switzerland AG 2020
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_52

771

outline of the liver, however, this is time consuming and laborious, computer-aided methods have been developed for semi-automatic and automatic liver segmentation [6]. Several algorithmic methods for liver tissue segmentation have been proposed in the literature.

Initial approaches for computer-assisted liver segmentation were based on thresholding of regions, the investigations carried out in [7, 8] proposed algorithms that use explicit thresholds or estimate the gray values in the liver, by means of a statistical analysis of manually segmented images or by the analysis of histograms, while other works were based on the analysis of regions in the liver, using the method of thresholding has been developed over time by researchers such as in [9–11]. In [12, 13], it is investigated other approaches, estimating not only the gray levels, but also identifying the organs near the liver in the computed tomography images. On the other hand, methods based on neural networks were implemented and subsequently applied to a new data set for the automatic detection of liver tissue, using region-based approaches applied for liver segmentation in order to separate the liver from adjacent organs and identifying structures as lesions that appear darker or brighter than tissue.

Hepatic segmentation, generally, cannot be solved by a simple approach based on identifying the region, and, therefore, investigations have been developed algorithms that combine several methods. In order to contribute to the treatment of liver diseases, it is proposed to develop an automated image processing system to segment lesions in the liver from images acquired by computed tomography. The proposed system implements a segmentation algorithm, based on convolutional networks, to detect lesions or abnormalities in liver tissue.

The rest of the paper is organized as follow, in Sect. 2 the proposed system is explained in detail, in Sect. 3, the results of applying the proposed system to liver CT images are evaluated, and finally, Sect. 4 provides a conclusion.

2 Methods

In this section the proposed system to segment liver lesions is explained in detail, the main components of the system are show in Fig. 1.

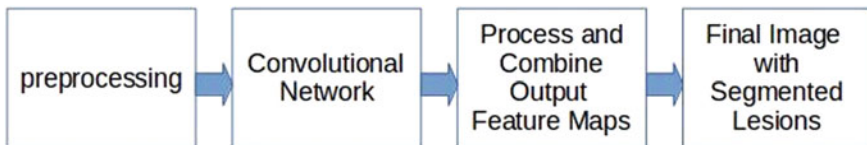


Fig. 1 Proposed system to segment liver lesions

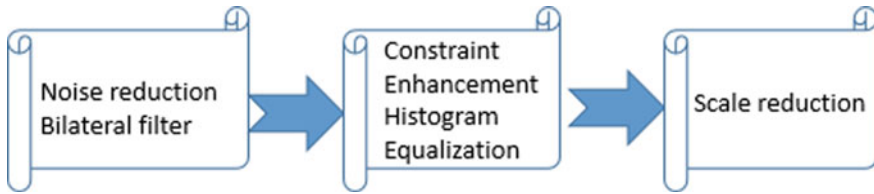


Fig. 2 Preprocessing steps of the input images, consisting in noise reduction, contrast enhancement, and reduction of the size of the images

2.1 Preprocessing

The quality of a medical image depends on the technique and acquisition machine. In the case of CT, especially in reduced radiation dose acquisitions, there exists an increased level of noise in the measured data and the respective reconstructed image, thus degrading the diagnostic value of the CT examination and making difficult for algorithms to analyze the images [14, 15]. Therefore, the pre-processing of the image is an important task in order to obtain a higher quality in the images [16]. Improving the quality of an image consists of addressing one or several problems, such as eliminating noise, increasing contrast, emphasizing edges and relevant structures, detecting points of interest, uniforming lighting, eliminating acquisition artifacts, in another [17]. In this research the preprocessing consisted in noise reduction, contrast enhancement, and reduction of the size of the images. Figure 2 is show a flowchart of the steps of the preprocessing system.

For the noise reduction step, the bilateral filter [18] was used, since the images processed did not have a heavy noise, the bilateral filter is effective a faster than other methods such as wavelet based denoisers. In the next step, a histogram equalization technique was used to improve the contrast. Finally, it was decided to reduce the dimensions of the image that were originally 460 380 so that the dimensions were 256 256 in order to delimit even more the area where the liver was, and also so that at the moment of train the network with the data, the process will be faster, and required less memory.

2.2 Network Architecture

This work focuses on the segmentation of medical images of computed tomography images, for this task we select the SegNet network architecture [19] and adapt it to segment the liver lesions. For this end, the proposed network is made up of 5 layers of convolution, with no pooling layers. The proposed network can be seen in Fig. 3. Note that the network has three outputs.

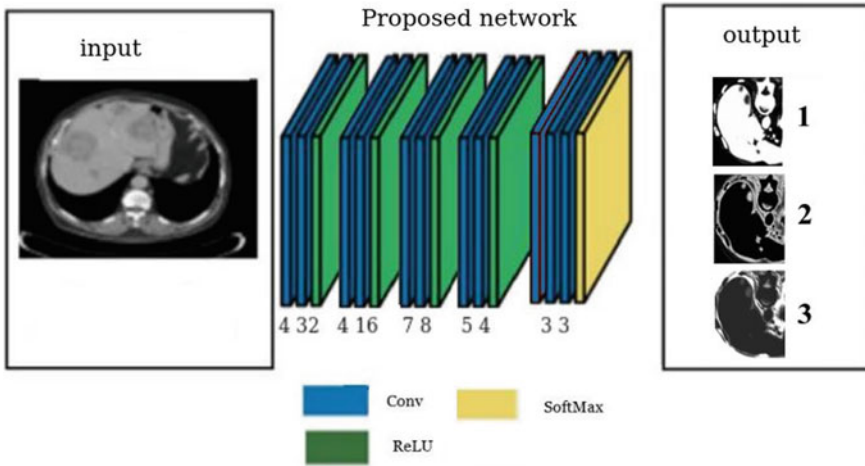


Fig. 3 Proposed deep learning network

Table 1 Layers of the proposed architecture

Layer	Number of	Size of
	Kernels	The kernel
1	32	4 4
2	16	4 4
3	8	7 7
4	4	5 5
5	3	3 3

In Table 1 is show the number and size of the kernels for each layer, in the last layer we have three outputs, intended for detection of the tumor, the liver, and background, respectively.

2.3 Post-processing

To obtain the final image, the outputs of the final layer were combined. The first two output images were used: the segmentation of the liver (output 1) and the segmentation of the lesions (output 2). This processing was necessary since the output 2 contains other structures outside the liver, this is described below. From the network output 1, a liver mask was constructed, which was obtained by thresholding the output image with a threshold of a gray level of 120 and a morphological opening operation (circular structuring element of 30 pixels) followed by an area filtering of areas greater than 10,000 pixels (see Fig. 4). Subsequently, the resulting mask was combined using an AND operation with the output 2 of the network (see Fig. 4), to

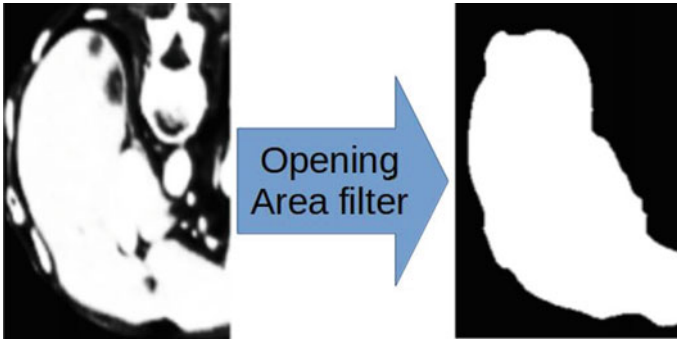


Fig. 4 The network output 1 is thresholded, opened morphologically, and filtered by area, to obtain a binary mass of the liver

obtain a segmented image of the liver with the lesions, this image is further thresholded to obtain a binary mask of the lesions. For the implementation of the network, the Caffe deep learning framework [20] was used and the MatlabTM interface was used for the preprocessing and the post-processing.

The time for training of the network was approximately one day on a computer with intel Corei7 processor and 8 Gbytes of RAM. Also, for testing and training, images from the database maintained by Clark et al. [21] and images from the Liver Tumor Segmentation Challenge (LiTS). In total 70 images were augmented with linear operations, to obtain up to 700 images used for training.

2.4 Validation of Group of Instances

To validate the proposed method, the areas of injury found by the network were compared against the areas segmented by the medical expert. Three types of areas were determined (Fig. 5).

{Type 1: area drawn by the medical expert and not found by the network, (False negative).

{Type 2: area determined by the medical expert and the network, (True positive).

{Type 3: area determined by the network and not by the medical expert. (False positive).

3 Results

In this section we show intermediate results of the processing done and the experimentation for validate our approach.

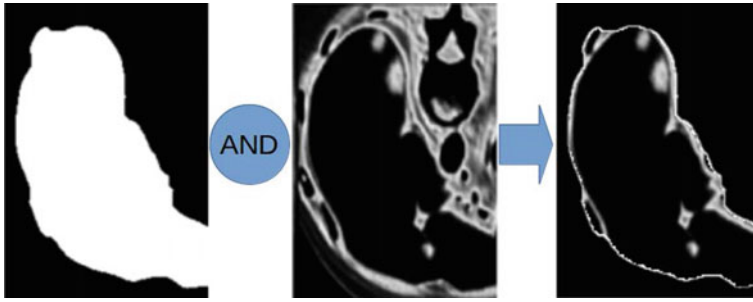


Fig. 5 An AND operation of network output 2 and a mask of the liver is performed to filter tissues outside the liver

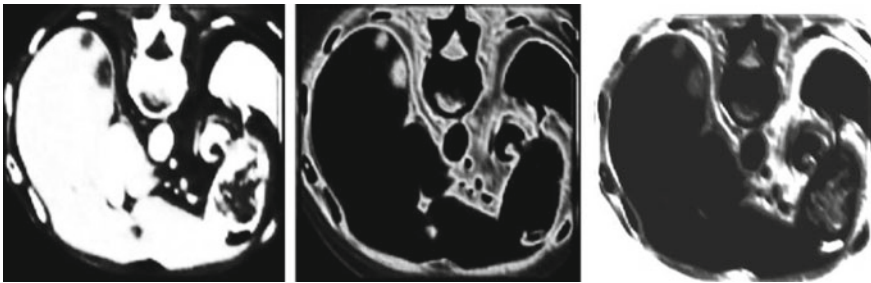


Fig. 6 Images of the output filters of the proposed network. From left to right: liver segmentation, lesions segmentation, the rest. The output image of the last filter was not used

Figure 6 show the output of the proposed network, the first two images are further processed while the last output was discarded.

From the first filter which output is liver segmentation a mask was obtained, an example is show in Fig. 7. Segmentation of liver images to test the performance of the proposed sys-tem, we use several images from the test set and compared with the segmentation made by an expert. For quantitative results we evaluated with the measures of Sect. 2.4.

The results of the computed tomography images by the proposed segmentation system can be seen in Fig. 8 right, while the Fig. 8 left shows the segmentation of the organ with its respective lesions segmented by the medical expert (database).

The experimental evaluation was performed on the CT images listed in Table 2. It can be observed that the average of the Type 2 (True Positive) area of the lesion present in the liver tissue reached was 82.81%, with the best accuracy at 92.307%.

The average of the percentages of the Type 2 area, shown in Table 2, indicates that the system has an acceptable performance for the segmentation of the hepatic lesions. Figure 9, show the segmented area of the lesion in the liver of tomographies 1 and 6 in more detail (Table 2), likewise, the larger area can be seen in purple in Fig. 9a of the lesion that detected the proposed system, which was the Type 2 area,

Fig. 7 Mask of the liver, obtained from the filter 1 of the proposed network

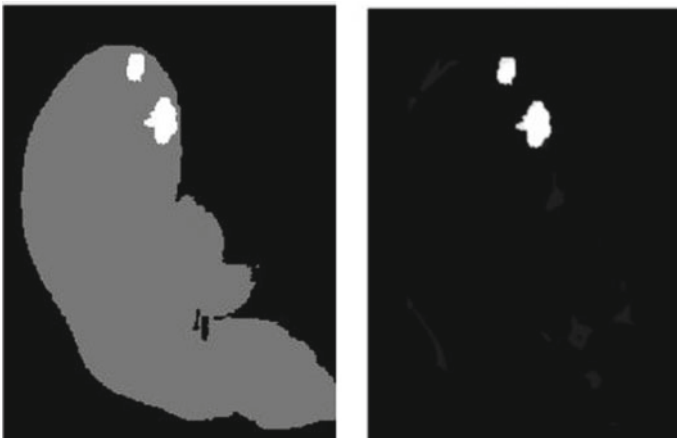


Fig. 8 Segmentation of lesions, left, by medical expert, right, by the proposed system

Table 2 Percentage of area type 2

CT tomography area type 2	
Tomography 1	80.224%
Tomography 2	79.918%
Tomography 3	92.307%
Tomography 4	80.152%
Tomography 5	91.542%
Tomography 6	72.772%

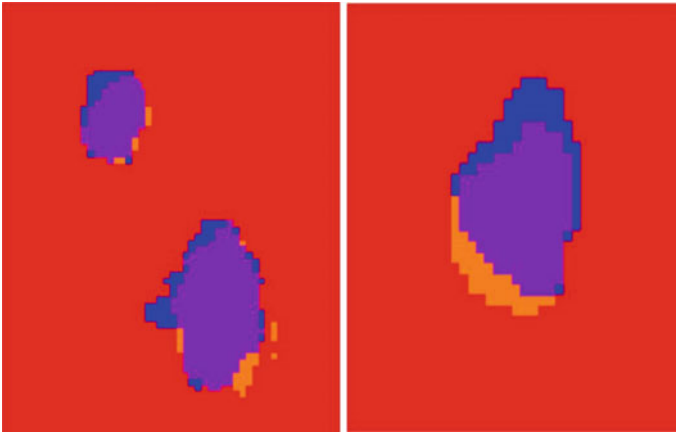


Fig. 9 Segmentation comparison: In blue, (Type 1) area drawn by the medical expert and not found by the network; in purple (Type 2) area determined by the medical expert and the network, and in orange (Type 3) area determined by the network and not by the medical expert. **a** Liver with two lesions. **b** Liver with one lesion

while, that the Type 1 areas in blue (True negative) and Type 3 in orange (False Positive), are smaller compared to the area and percent-age of the area that detected the system, which was 80.2247%. Figure 9b also shows the largest area in purple color of the lesion that detected the proposed system network, that is, the Type 2 area, while the Type 1 areas in blue (True negative) and Type 3 in orange (False Positive), are smaller compared to the area and the percentage of detected area by the system, which was 72.7723%.

4 Conclusions

The purpose of this work was to develop a system that allows the automatic segmentation of lesions in the liver in computed tomography images based on neural networks. The tests performed on the computed tomography images to segment the lesions present in the hepatic volumes showed that the proposed method detect correctly areas of lesions in the liver (type 2) with an average percentage of 82.81%. Future research to improve the effectiveness of the proposed segmentation method are the following: To explore new network architectures that allow to better segment the liver tissue, and therefore obtain a higher precision during the process of segmentation in the computed tomography images. Expand the validation of the segmentation method exposed in this work, by obtaining new volumes of the liver, as well as the comparison with other segmentation algorithms.

References

1. Chartrand, G., Cresson, T., Chav, R., Gotra, A., Tang, A., De Guise, J.A.: Liver segmentation on ct and mr using laplacian mesh optimization. *IEEE Trans. Biomed. Eng.* **64**(9), 2110–2121 (2017)
2. Bonny, S., Chanu, Y.J., Singh, K.M.: Speckle reduction of ultrasound medical images using Bhattacharyya distance in modified non-local mean filter. *SIViP* **13**(2), 299–305 (2019)
3. Chen, E.L., Chung, P.C., Chen, C.L., Tsai, H.M., Chang, C.I.: An automatic di-agnostic system for ct liver image classification. *IEEE Trans. Biomed. Eng.* **45**(6), 783–794 (1998)
4. Diamant, I., Hoogi, A., Beaulieu, C.F., Safdari, M., Klang, E., Amitai, M., Greenspan, H., Rubin, D.L.: Improved patch-based automated liver lesion classification by separate analysis of the interior and boundary regions. *IEEE J. Biomed. Health Inform.* **20**(6), 1585–1594 (2016)
5. Wein, W., Brunke, S., Khamene, A., Callstrom, M.R., Navab, N.: Automatic ct-ultrasound registration for diagnostic imaging and image-guided intervention. *Med. Image Anal.* **12**(5), 577–585 (2008)
6. Yang, X., Yu, H.C., Choi, Y., Lee, W., Wang, B., Yang, J., Hwang, H., Kim, J.H., Song, J., Cho, B.H., et al.: A hybrid semiautomatic method for liver segmentation based on level-set methods using multiple seed points. *Compu. Methods Programs Biomed.* **113**(1), 69–79 (2014)
7. Bae, K.T., Giger, M.L., Chen, C.T., Kahn, C.E.: Automatic segmentation of liver structure in ct images. *Med. Phys.* **20**(1), 71–78 (1993)
8. Gao, L., Heath, D.G., Kuszyk, B.S., Fishman, E.K.: Automatic liver segmentation technique for three-dimensional visualization of ct data. *Radiology* **201**(2), 359–364 (1996)
9. Platero, C., Poncela, J.M., Gonzalez, P., Tobar, M.C., Sanguino, J., Asensio, G., Santos, E.: Liver segmentation for hepatic lesions detection and characterisation. In: *Biomedical Imaging: From Nano to Macro*. In: 5th IEEE International Symposium on ISBI 2008, pp. 13–16. IEEE (2008)
10. Hegde, R.B., Prasad, K., Hebbar, H., Singh, B.M.K.: Image processing approach for detection of leukocytes in peripheral blood smears. *J. Med. Syst.* **43**(5), 114:1–114:11 (2019)
11. Soler, L., Delingette, H., Malandain, G., Montagnat, J., Ayache, N., Koehl, C., Dourthe, O., Malassagne, B., Smith, M., Mutter, D., et al.: Fully automatic anatomical, pathological, and functional segmentation from ct scans for hepatic surgery. *Comput. Aided Surg.* **6**(3), 131–142 (2001)
12. Koss, J.E., Newman, F., Johnson, T., Kirch, D.: Abdominal organ segmentation using texture transforms and a Hopfield neural network. *IEEE Trans. Med. Imaging* **18**(7), 640–648 (1999)
13. Lee, C.C., Chung, P.C., Tsai, H.M.: Identifying multiple abdominal organs from cut image series using a multimodal contextual neural network and spatial fuzzy rules. *IEEE Trans. Inform. Technol. Biomed.* **7**(3), 208–217 (2003)
14. Kang, E., Min, J., Ye, J.C.: A deep convolutional neural network using directional wavelets for low-dose x-ray ct reconstruction. *Med. Phys.* **44**(10) (2017)
15. Jahangiri, A., Sepulveda, F.: The relative contribution of high-gamma linguistic processing stages of word production, and motor imagery of articulation in class separability of covert speech tasks in EEG data. *J. Med. Syst.* **43**(2), 20:1–20:9 (2019)
16. Hammon, R., Seuss, H., Hammon, M., Grillhösl, C., Heiss, R., Zeilinger, M., Bayerl, N., Vuylsteke, P., Wanninger, F., Schroth, M., Uder, M., Rompel, O.: Improved visualization of peripherally inserted central catheters on chest radiographs of neonates using fractional multiscale image processing. *BMC Med. Imaging* **19**(1), 3:1–3:7 (2019)
17. Pérez, M.A.B.: Desarrollo de algoritmos de procesamiento de imagen avanzado para interpretación de imágenes médicas. Aplicación a segmentación de hígado sobre imágenes de resonancia magnética multisequencia. Ph.D. thesis, Universidad del País Vasco-Euskal Herriko Unibertsitatea (2016)
18. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: *Sixth International Conference on Computer Vision*, pp. 839–846. IEEE (1998)
19. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: a deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561* (2015)

20. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Ca e: Convolutional architecture for fast feature embedding. arXiv preprint [arXiv:1408.5093](https://arxiv.org/abs/1408.5093) (2014)
21. Clark, K., Vendt, B., Smith, K., Freymann, J., Kirby, J., Koppel, P., Moore, S., Phillips, S., Maffitt, D., Pringle, M., et al.: The cancer imaging archive (TCIA): maintaining and operating a public information repository. *J. Digit. Imaging* **26**(6), 1045–1057 (2013)

Intuitionistic Fuzzy Sugeno Integral for Face Recognition



Gabriela E. Martínez and Patricia Melin

Abstract In this paper, an extension of the Sugeno integral using the operators of the intuitionistic fuzzy sets is presented. The proposed method consists of using the Sugeno integral as an integration method of multiple information sources using the degrees of membership and non-membership through the application of the operators of the intuitionistic fuzzy sets. The proposed method is used to combine the modules output of a modular neural network for face recognition. In this paper, the focus is on aggregation operator that use measures with intuitionistic fuzzy sets, in particular the Sugeno integral. The performance of the proposed method is compared with the traditional Sugeno integral using the Cropped Yale database.

Keywords Aggregation operators · Sugeno integral · Modular neural networks · Intuitionistic fuzzy sets

1 Introduction

In various applications, working with imprecise or inaccurate data is required, due to this, it is common to find certain obstacles due to the uncertainty that these generate. Zadeh proposed the fuzzy sets [26] in 1965 as a solution, later Michio Sugeno defined the fuzzy integral and fuzzy measure terms [22] as a more appropriate way to measure parameters that depend on human subjectivity, as a certain degree of uncertainty. The concept of intuitionistic fuzzy sets, which allows having degrees of membership and degrees of non-membership was introduced In 1983 by Atanassov [1]. There are several aggregation operators, such as the ordered weighted averaging (OWA), the OWA weighting, the Sugeno integral [22], the weighted average, the harmonic mean, the geometric mean, the arithmetic mean, and the Choquet integral [7]; each one results in the aggregation or the combination of the different sources of information that are considered as the input variables; when the information can

G. E. Martínez · P. Melin (✉)
Tijuana Institute of Technology, Tijuana, Mexico
e-mail: pmelin@tectijuana.mx

© Springer Nature Switzerland AG 2020
O. Castillo et al. (eds.), *Intuitionistic and Type-2 Fuzzy Logic Enhancements in Neural and Optimization Algorithms: Theory and Applications*, Studies in Computational Intelligence 862, https://doi.org/10.1007/978-3-030-35445-9_53

be mathematically formalized, combined and reduced into a single representative value, it is said that an aggregation operator is being used.

The Sugeno integral has been implemented successfully in several applications; in [21] the Sugeno integral is used for building the Sugeno based mean value for some specific fuzzy quantities; In [13] a comparison between Choquet and Sugeno integrals as aggregation operators for pattern recognition is presented; in [19] interval type-2 fuzzy logic is used for module relevance estimation in Sugeno integration of modular neural networks. On the other hand, the intuitionistic fuzzy sets have had a great boom in recent years [2–4]; In [11] the theories of intuitionistic fuzzy sets and artificial neural networks are combined for intrusion detection. in [15] the Choquet and Sugeno integrals and intuitionistic fuzzy integrals as aggregation operators are presented. in [12] the concepts of interval fuzzy-valued, intuitionistic fuzzy-valued and interval intuitionistic fuzzy-valued Sugeno integrals was introduced. For this reason, it is interesting to carry out the implementation of the Sugeno integral as aggregation operator of a modular neuronal network using intuitionist fuzzy logic to use the degrees of membership and degrees of non-membership using the Cropped Yale database.

Through the implementation of a modular network of three modules, where each one is considered as a source of information, the face recognition was carried out satisfactorily.

The rest of the paper is structured as follows: the basic concepts of the Sugeno measures and fuzzy integrals is described in Sects. 2, 3 the extension of the Sugeno integral using intuitionistic fuzzy sets is presented. Section 4 shows the case study using the modular neural network. We explain the simulation results and the advantages of the proposed technique with benchmark face databases in Sect. 5, and finally Sect. 6 offers some conclusions of the proposed method.

2 Sugeno Measures and Fuzzy Integrals

In this section, we can find the basic concepts of the Monotonic measures and the Sugeno integral.

2.1 *Monotonic Measures*

A measure is considered in the area of mathematics as one of the most important concepts. We can often find in the literature that monotonous measures refer to fuzzy measures. This is somewhat confusing, because fuzzy sets are not related to the definition of these measures, due to this, in families of fuzzy sets the term “fuzzy measures” must be used for the measures (not additive or additive).

If we have a universe of discourse X and a nonempty family C of subsets of X , a monotone measure μ on $\langle X, C \rangle$ is a function of the form $\mu: C \rightarrow [0, \infty]$. It is assumed

that the universal set X is finite and that $C = P(X)$. That is, normally assumed that the monotonic measures are sets of functions $\mu: P(X) \rightarrow [0, 1]$.

Definition 1 On space X , a monotonic set measure μ is mapping $\mu: P(X) \rightarrow [0, 1]$ such that the following properties hold [10, 25]:

- (1) $\mu(\phi) = 0$;
- (2) $\mu(X) = 1$;
- (3) For all if $A, B \in P(X)$, if $A \subseteq B$, then $\mu(A) \leq \mu(B)$.

2.2 Sugeno Measures

Special types of monotonic measures are the Sugeno λ -measures and they are defined as follows [6, 17].

Definition 2 If we have a finite set $X = \{x_1, \dots, x_n\}$. A discrete fuzzy measure on X is a function $\mu: 2^X \rightarrow [0, 1]$ with the following properties:

- (1) $\mu(\phi) = 0$ and $\mu(X) = 1$;
- (2) Given $A, B \in 2^X$ if $A \subset B$ then $\mu(A) \leq \mu(B)$ (monotonicity property).

The identifiers of sources of information are considered in the set X . For a subset $A \subseteq X$, $\mu(A)$ is considered to be the relevance degree of this subset of information.

Definition 3 Let $X = \{x_1, \dots, x_n\}$ be any finite set and let $\lambda \in (-1, +\infty)$. A Sugeno λ -measure is a function μ from 2^X to $[0, 1]$ with the following properties:

- (1) $\mu(X) = 1$;
- (2) if $A, B \subseteq X$ with $A \cap B = \phi$, then

$$\mu(A \cup B) = \mu(A) + \mu(B) + \lambda \mu(A)\mu(B) \tag{1}$$

where $\lambda > -1$, usually the Eq. (1) is called the λ -rule.

The densities are interpreted as the importance of the individual information sources. If X is a finite set, the fuzzy densities represented by $\mu(\{x\})$ are given for each $x \in X$, the measure of a set A of information sources is interpreted as the importance of that subset of sources [17].

For each $A \subset P(X)$, the value of $\mu(A)$ can be calculated by the recurrent application of the λ -rule, and can be represented in the following form.

$$\mu(A) = \left[\frac{\prod_{x \in A} (1 + \lambda \mu(\{x\}))}{\lambda} \right] \tag{2}$$

For each $x \in X$, given the values of the fuzzy densities $\mu(\{x\})$, the value of λ can be determined by using the constraint $\mu(\{x\}) = 1$, which applied in (2) results in (3):

$$\lambda + 1 = \prod_{i=1}^n (1 + \lambda \mu(\{x_i\})) \tag{3}$$

This type of measures uses the parameter λ and once the densities are known, these can be calculated applying (3); Sugeno proved that this polynomial has a real root greater than -1 . The property (1), specifying the n different densities, thereby reducing the number of free parameters from $2^n - 2$ to n [6], using Theorem 1, it is possible to determine the value of the λ parameter [23].

Theorem 1 *For each $x \in X$, let $\mu(\{x\}) < 1$ and $\mu(\{x\}) > 0$ for at least two elements of X . Then, it is possible to determine a unique parameter λ using (3) as follows:*

- If $\sum_{x \in X} \mu(\{x\}) < 1$, then λ is a value in the interval $(0, \infty)$.
- If $\sum_{x \in X} \mu(\{x\}) = 0$, then $\lambda = 0$;
- If $\sum_{x \in X} \mu(\{x\}) > 1$, then λ is a value in the interval $(-1, 0)$.

When μ is a λ -fuzzy measure, the values of $\mu(A_i)$ can be computed by means of (2), or recursively, reordering of the sets X and $\mu(\{x\})$, with respect to the values of the elements of set X [24].

2.3 Sugeno Integrals

Sugeno develops the concept of fuzzy integral as nonlinear functions defined with respect to fuzzy measures as λ -fuzzy measure using the concept of fuzzy measures; we can interpret the fuzzy integral as finding the maximum degree of similarity between the target and the expected value as shown in (4). The Sugeno integral generalizes “max-min” operators.

$$Sugeno_{\mu}(x_1, x_2, \dots, x_n) = \max_{i=1,n} (\min(f(x\sigma(i)), \mu(A\sigma(i)))) \tag{4}$$

with $A_0 = 0$, where $x_{\sigma(i)}$ indicates the indices that must be permuted as shown in (5), and where $A_{\sigma(i)} = \{A_{\sigma(i)}, \dots, A_{\sigma(n)}\}$.

$$0 \leq f(x\sigma(1)) \leq f(x\sigma(2)) \leq \dots \leq f(x\sigma(n)) \leq 1 \tag{5}$$

The Sugeno integral can be applied to solve several problems, which consider a finite set of n elements $X = \{x_1, \dots, x_n\}$.

3 Extension of the Sugeno Integral with Intuitionistic Fuzzy Sets

In this section, we can find the basic concepts of the Intuitionistic fuzzy set as well as the proposed method by using the Sugeno integral.

3.1 Intuitionistic Fuzzy Set

Let a set E be fixed and let $A \subset E$ be a fixed sets. Intuitionistic fuzzy sets A^* is defined as (6)

$$A^* = \{ \langle x, \mu_A(x), \nu_A(x) \rangle | x \in E \} \tag{6}$$

where function $\mu_A : E \rightarrow [0,1]$ and $\nu_A : E \rightarrow [0,1]$ define the degree of membership and the degree of non-membership of the element $x \in E$ to the set A , respectively, and for every $x \in E$

$$0 \leq \mu_A(x) + \nu_A(x) \leq 1 \tag{7}$$

Obviously, every ordinary fuzzy set has the form

$$\{ \langle x, \mu_A(x), 1 - \mu_A(x) \rangle | x \in E \} \tag{8}$$

if

$$\pi_A(x) = 1 - \mu_A(x) - \nu_A(x) \tag{9}$$

then $\pi_A(x)$ is the degree of non-determinacy (uncertainty/hesitancy) of the membership of element $x \in E$ to the set A . in the case of ordinary fuzzy sets, $\pi_A(x) = 0$ for every $x \in E$ [5].

3.2 Sugeno Integral with Intuitionistic Fuzzy Set

The Sugeno integral (4) was extend using the concept of intuitionistic fuzzy sets by the use of degrees of membership and degrees of non-membership, obtaining (10).

$$\begin{aligned}
 & IFS_{u(\mu_A, \nu_A)}(x_1, x_2, \dots, x_n) \\
 &= \left(\max_{i=1..n} \left(\min(f(x_{\sigma(i)}), \mu_A(A_{\sigma(i)})) \right), \max_{i=1..n} \left(\min(f(x_{\sigma(i)}), \nu_A(A_{\sigma(i)})) \right) \right) \tag{10}
 \end{aligned}$$

when in a problem exists a finite set of n elements $X = \{x_1, \dots, x_n\}$, this can be solved by applying the Sugeno integral, and the proposed method of intuitionistic fuzzy Sugeno integral.

3.3 Intuitionistic Fuzzy Sugeno Integral Using $\pi_A = 0.4$

If we have that $x = \{0.9, 0.6, 0.3\}$ represents the information sources and associated to each entry a fuzzy density or membership value as $\mu_i = \{0.3, 0.4, 0.1\}$, we calculate $\lambda = 1$.

The calculated fuzzy measures are $\mu_A(A_{\sigma(i)}) = \{0.3, 0.82, 1\}$. Using the intuitionistic fuzzy densities or not membership values as $\nu_A(\mu_i) = \{0.3, 0.2, 0.5\}$, it was calculated a $\lambda = 0$ and the intuitionistic fuzzy measures calculated are $\nu_A(A_{\sigma(i)}) = \{0.3, 0.5, 1\}$. After that, using (10) we calculate the intuitionistic Sugeno integral

$$IFSU(\mu_A, \nu_A) = (\max(\min((0.9, 0.3), (0.6, 0.82), (0.3, 1))), \max(\min((0.9, 0.3), (0.6, 0.5), (0.3, 1))))$$

once the minimums have been calculated, the maxima are calculated

$$IFSU(\mu_A, \nu_A) = (\max(0.3, 0.6, 0.3), \max(0.3, 0.5, 0.3))$$

we obtain the maxima of μ_A, ν_A

$$IFSU(\mu_A, \nu_A) = (0.6, 0.5)$$

and finally we calculate the mean of the obtained interval, however, is possible determine other way for extract a representative value of the interval obtained.

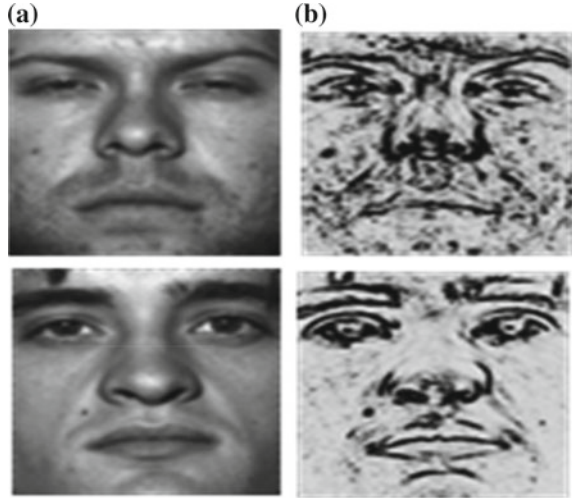
$$\overline{IFSU(\mu_A, \nu_A)} = 0.55.$$

The information from different sources can be integrated with the proposed operator.

4 Sugeno Integral with Intuitionistic Fuzzy Sets in a Modular Neural Network

A neural network is a computational attempt to simulate the behavior observed in human brain. The proposed method was used for the integration of the modules of a modular neural network (MNN), using a benchmark database.

Fig. 1 **a** Original images of the Cropped Yale database, **b** image with MGT1



4.1 The Cropped Yale Database

The Cropped Yale contains images of the faces of 38 people with 60 samples per person giving 2280 images. Each image of the Cropped Yale database has a size of 168×192 [8]. In Fig. 1a, we can see two original images of faces and in Fig. 1b the same images are shown to which a preprocessing with the morphological gradient edge detector type-1 (MGT1) is applied [18].

4.2 Modular Neural Network (MNN)

We define a MNN of three modules using as integration of responses the Sugeno integral with intuitionistic fuzzy sets. For the MNN, we used the 80% of the images for training and 20% per testing.

Each image of the Cropped Yale database was pre-processed to improve the performance of the MNN. After that, each of the images was divided into three horizontal sections each of which serves as input to the modules of the MNN. In Fig. 2, we can appreciate a scheme of the proposed architecture of the MNN.

4.3 Training Parameters

We used three modules for the training of the MNN. The parameters used for the training of the MNN can be seen in Table 1.

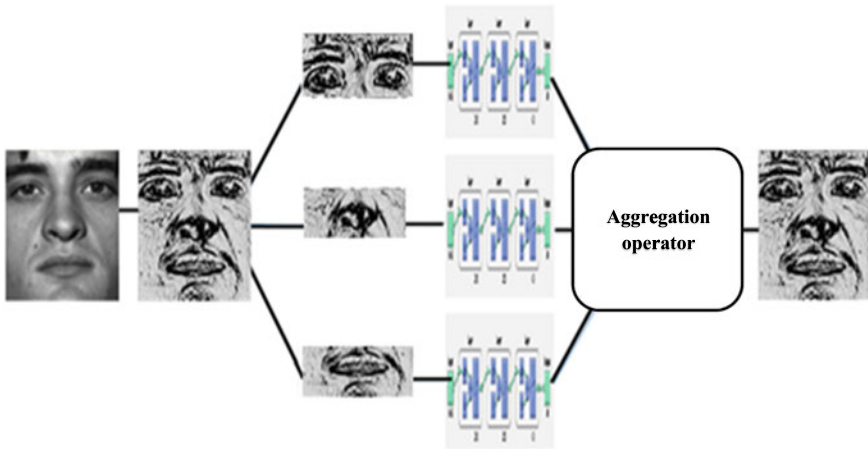


Fig. 2 The modular neural network architecture

Table 1 Training parameters of the MNN

Parameters	
Modules of the MNN	3
Hidden layers per module	2
Neurons per layer	200
Error goal	0.0001
Epoch	500
Training method	Traingdx

With the implementation of this integral, each information source has an associated fuzzy density. The values 0.1, 0.5 and 0.9 was arbitrary chosen and some permutations were made with these values to obtain different simulations. Using the intuitionistic fuzzy Sugeno integral is possible to determine the final decision by the combination of the simulation vectors of the outputs of the modules trained into a simple vector.

5 Simulation Results

This section provides a comparison of the recognition rate achieved by the MNN for face recognition system were the intuitionistic Sugeno integral was applied. In the experimental results, the MGT1 was analyzed. The tests were performed using the CROPPED YALE database. The combination of the modules output of the MNN for face recognition was performance using the MGT1 edge detector, and applying the Sugeno integral as integration of the modules of the MNN.

Table 2 Results using the Sugeno integral applying the MGT1

Test number	Test 1			Test 2		
	Mean	Std	Max	Mean	Std	Max
1	1	0	1	0.9974	0.0059	1
2	0.9921	0.0118	1	0.9816	0.0150	1
3	0.9921	0.0118	1	0.9789	0.0177	1
4	1	0	1	1	0	1
5	0.9974	0.0059	1	0.9947	0.0072	1
6	0.9974	0.0059	1	0.9921	0.0072	1
7	1	0	1	0.9974	0.0059	1
8	0.9974	0.0059	1	0.9947	0.0072	1
9	0.9974	0.0059	1	0.9947	0.0072	1
10	1	0	1	0.9974	0.0059	1
11	0.9974	0.0059	1	0.9974	0.0059	1
12	0.9974	0.0059	1	0.9947	0.0072	1
13	1	0	1	0.9974	0.0059	1
14	0.9921	0.0072	1	0.9974	0.0059	1
15	0.9921	0.0072	1	0.9921	0.0118	1
Mean	0.9968	0.0049	1	0.9939	0.0077	1

The mean rate, standard deviation and the max values achieved by the system, are shown in Table 2; in the first test we can appreciate that was obtained a mean rate of 99.68%, a standard deviation of 0.0049 and a max rate of 100%, and in the second test we were obtained a mean rate of 99.39%, a standard deviation of 0.0077 and a max rate of 100%.

In another test, the system was considered using the intuitionistic fuzzy Sugeno integral with a $\pi_A = 0.2, 0.3$ and 0.4 using the preprocessing of the MGT1 edge detector over the images of the CROPPED YALE database. In Table 3, is possible to appreciate the results obtained using the intuitionistic fuzzy Sugeno integral as aggregation operator with the preprocessing of the images with the MGT; we can see that the mean rate is of 99.70%, a standard deviation of 0.0061 and a max rate of 100% using $\pi_A = 0.2$ and a mean rate is of 99.72%, a standard deviation of 0.0060 and a max rate of 100% using $\pi_A = 0.3$.

In Table 4, we can appreciate that the mean rate is of 99.70%, a standard deviation of 0.0061 and a max rate of 100% using $\pi_A = 0.4$; for this test also the preprocessing of the MGT1 images was applied.

The results obtained over the Cropped Yale database are presented in Table 5; we can see that the recognition rate using the Intuitionistic fuzzy Sugeno integral is better than the Sugeno integral. In addition, we can notice that the mean rate value is better when the MGT1 edge detector is applied with a value of 99.72% using the

Table 3 Results applying intuitionistic fuzzy Sugeno integral with $\pi_A = 0.2$ and $\pi_A = 0.3$ using MGT1 edge detector

Test number	$\pi_A = 0.2$			$\pi_A = 0.3$		
	Mean	Std	Max	Mean	Std	Max
1	0.9974	0.0059	1	0.9974	0.0059	0.9974
2	0.9974	0.0059	1	0.9974	0.0059	0.9974
3	0.9947	0.0072	1	0.9974	0.0059	0.9974
4	0.9974	0.0059	1	0.9974	0.0059	0.9974
5	0.9974	0.0059	1	0.9974	0.0059	0.9974
6	0.9974	0.0059	1	0.9974	0.0059	0.9974
7	0.9974	0.0059	1	0.9974	0.0059	0.9974
8	0.9974	0.0059	1	0.9947	0.0072	0.9947
9	0.9974	0.0059	1	0.9974	0.0059	0.9974
10	0.9974	0.0059	1	0.9974	0.0059	0.9974
11	0.9974	0.0059	1	0.9974	0.0059	0.9974
12	0.9947	0.0072	1	0.9974	0.0059	0.9974
13	0.9974	0.0059	1	0.9974	0.0059	0.9974
14	0.9974	0.0059	1	0.9974	0.0059	0.9974
15	0.9974	0.0059	1	0.9974	0.0059	0.9974
Mean	0.9970	0.0061	1	0.9972	0.0060	0.9972

Table 4 Results applying intuitionistic fuzzy Sugeno integral with $\pi_A = 0.4$ using MGT1 edge detector

Test number	$\pi_A = 0.4$		
	Mean	Std	Max
1	0.9974	0.0059	1
2	0.9974	0.0059	1
3	0.9974	0.0059	1
4	0.9974	0.0059	1
5	0.9974	0.0059	1
6	0.9974	0.0059	1
7	0.9974	0.0059	1
8	0.9974	0.0059	1
9	0.9974	0.0059	1
10	0.9947	0.0072	1
11	0.9974	0.0059	1
12	0.9947	0.0072	1
13	0.9974	0.0059	1
14	0.9974	0.0059	1
15	0.9974	0.0059	1
Mean	0.9970	0.0061	1

Table 5 Results using the Sugeno integral and intuitionistic Sugeno integral

Integration method	Mean	Std	Max	π_A
Sugeno integral	0.9954	0.0063	1	–
Intuitionistic Sugeno integral	0.9970	0.0061	1	0.2
	0.9972	0.0060	1	0.3
	0.9970	0.0061	1	0.4

proposed method. In these results we can also appreciate that the mean rate, standard deviation and max rate were better when we applied a $\pi_A = 0.3$.

6 Conclusion

In this paper, we have presented the intuitionistic fuzzy Sugeno integral as an integration operator for a modular neural network. In Table 5, we can appreciate that the recognition results with the proposed method are better or comparable to results produced by the Sugeno integral; however, it is necessary to perform more experimentation with different fuzzy densities, π values, and databases. As future work, it is proposed to vary the number of information sources, as well as perform the dynamic allocation of fuzzy densities to each of the sources. In addition, considering other applications as future work, as in [9, 14, 16, 20].

References

1. Atanassov, K.: Intuitionistic fuzzy sets, VII ITKR’s Session, Sofia, June 1983 (Deposed in Central Sci.- Techn. Library of Bulg. Acad. Of Sci., 1697/84, in Bulgarian). Reprinted: Int. J. Bioautomation, 2016, **20**(S1), S1–S6 (1983)
2. Atanassov, K.: Intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **20**(1), 87–96 (1986)
3. Atanassov, K.: New operations defined over the intuitionistic fuzzy sets. *Fuzzy Sets Syst.* **61**, 137–142 (1994)
4. Atanassov, K.: *Intuitionistic Fuzzy Sets: Theory and Applications*. Physica-Verlag, Heidelberg (1999)
5. Atanassov, K., Vassilev, P., Tsvetkov, R.: *Intuitionistic Fuzzy Sets, Measures and Integrals*, Bulgarian Academic Monographs, vol. 12, Sofia: “Prof. Marin Drinov” Academic Publishing House (2013)
6. Bezdek, J.C., Keller, J., Pal, N.R.: *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Springer-Verlag, New York (2005)
7. Choquet, G.: Theory of capacities. *Ann. Inst. Fourier, Grenoble* **5**, 131–295 (1953)
8. Database CROPPED YALE Face: Cambridge University Computer Laboratory. (November 2012). Retrieved from: <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
9. González, C.I., Melin, P., Castro, J.R., Castillo, O., Mendoza, O.: Optimization of interval type-2 fuzzy systems for image edge detection. *Appl. Soft Comput.* **47**, 631–643 (2016)
10. Klir, G.: *Uncertainty and Information*. Wiley, Hoboken, NJ (2005)

11. Lei, Y., Liu, J., Yin, H.: Intrusion detection techniques based on improved intuitionistic fuzzy neural networks. In: 2016 International Conference on Intelligent Networking and Collaborative Systems (INCoS), Ostrava, pp. 518–521 (2016)
12. Liu, Y., Kong, Z.: Interval intuitionistic fuzzy-valued Sugeno integral. In: Proceeding of 9th International Conference on Fuzzy Systems and Knowledge Discovery, Sichuan, pp. 89–92 (2012)
13. Martínez, G.E., Mendoza, O., Castro, J.R., Melin, P., Castillo, O.: Choquet integral with interval type 2 Sugeno measures as an integration method for modular neural networks. Proc. of WCSC **2014**, 71–86 (2014)
14. Melin, P., Castillo, O.: Intelligent control of complex electrochemical systems with a neuro-fuzzy-genetic approach. IEEE Trans. Industr. Electron. **48**(5), 951–955 (2001)
15. Melin, P., Martínez, G.E., Tsvetkov, R.: Choquet and Sugeno integrals and intuitionistic fuzzy integrals as aggregation operators. In: Proceeding of 4th International Intuitionistic Fuzzy Sets and Contemporary Mathematics Conference, pp. 95–99. Mersin, Turkey (2017)
16. Melin, P., Sánchez, D., Castillo, O.: Genetic optimization of modular neural networks with fuzzy response integration for human recognition. Inf. Sci. **197**, 1–19 (2012)
17. Mendez-Vazquez, A., Gader, P., Keller, J.M., Chamberlin, K.: Minimum classification error training for Choquet integrals with applications to landmine detection. IEEE Trans. Fuzzy Syst. **16**(1), 225–238 (2008)
18. Mendoza, O., Melin, P., Licea, G.: Interval type-2 fuzzy logic for edges detection in digital images. Int. J. Intell. Syst. **24**(11), 1115–1133 (2009)
19. Mendoza, O., Melin, P., Licea, G.: Interval type-2 fuzzy logic for module relevance estimation in Sugeno integration of modular neural networks. In: Soft Computing for Hybrid Intelligent Systems, Studies in Computational Intelligence, vol. 154, pp. 115–127. Springer (2008)
20. Sánchez, D., Melin, P., Castillo, O.: Optimization of modular granular neural networks using a firefly algorithm for human recognition. Eng. Appl. of AI **64**, 172–186 (2017)
21. Štajner-Papuga, I., Lozanov-Crvenković, Z., Grujić, G.: On Sugeno integral based mean value for fuzzy quantities. In: Proceeding of IEEE 14th International Symposium on Intelligent Systems and Informatics (SISY), pp. 155–160. Subotica (2016)
22. Sugeno, M.: Theory of Fuzzy Integrals and Its Applications. Doctoral Thesis, Tokyo Institute of Technology (1974)
23. Torra, V., Narukawa, Y.: Modeling Decisions, Information Fusion and Aggregation Operators. Springer-Verlag, Heidelberg (2007)
24. Verikas, A., Lipnickas, A., Malmqvist, K., Bacauskiene, M., Gelzinis, A.: Soft combination of neural classifiers: a comparative study. Pattern Recogn. Lett. **20**(4), 429–444 (1999)
25. Yager, R.: A knowledge-based approach to adversarial decision-making Int. J. Intell. Syst. **23**(1), 1–21 (2008)
26. Zadeh, L.A.: Fuzzy sets. Inform. Control **8**, 338–353 (1965)