PREPRINT

HYBRID EVOLUTIONARY MULTI-OBJECTIVE OPTIMISATION USING OUTRANKING-BASED ORDINAL CLASSIFICATION METHODS

Laura Cruz-Reyes[a], Eduardo Fernandez-Gonzalez[b], J Patricia Sanchez-Solis[c,*], Carlos A. Coello Coello[d], Claudia Gomez[a]

[a] Postgraduate & Research Division, National Mexican Institute of Technology/Madero Institute of Technology, 89440, Madero, Tamaulipas (Mexico), lauracruzreyes@itcm.edu.mx, claudia.gomez@itcm.edu.mx.
[b] Universidad Politécnica del Mar y la Sierra, 82740, La Cruz, Sinaloa (Mexico), eddyf171051@gmail.com
[c] Department of Electrical Engineering and Computing, Autonomous University of Ciudad Juarez, 32310, Ciudad Juarez, Chihuahua (Mexico), julia.sanchez@uacj.mx.
[d] Computer Science Department, CINVESTAV-IPN, 07360, Mexico, DF (Mexico), ccoello@cs.cinvestav.mx.

[*] Corresponding author: Plutarco Elias Calles 1210, Fovisste Chamizal, CP 32310, Cd. Juarez, Chihuahua, México, Tel. (+52) 656 688 21 00. E-mail: julia.sanchez@uacj.mx.

Abstract

A large number of real-world problems require optimising several objective functions at the same time, which are generally in conflict. Many of these problems have been addressed through multi-objective evolutionary algorithms. In this paper, we propose a new hybrid evolutionary algorithm whose main feature is the incorporation of the Decision Maker's (DM's) preferences through multi-criteria ordinal classification methods in early stages of the optimisation process, being progressively updated. This increases the selective pressure towards the privileged zone of the Pareto front more in agreement with the DM's preferences. An extensive experimental research was conducted to answer three main questions: i) to what extent the proposal improves the convergence towards the region of interest for the DM; ii) to what extent the proposal becomes more relevant as the number of objectives increases, and iii) to what extent the effectiveness of the hybrid algorithm depends on the particular multi-criteria method used to assign solutions to ordered classes. The issues used to evaluate our proposal and answer the questions were seven scalable test problems from the DTLZ test suite and some instances of project portfolio optimisation problems, with three and eight objectives. Compared to MOEA/D and MOEA/D-DE, the results showed that the proposed strategy obtains a better convergence towards the region of interest for the DM and also performs better characterisation of that zone on a wide range of objective functions.

Keywords
Evolutionary multi-objective optimisation; multi-criteria ordinal classification; preference incorporation.

1 Introduction

Nowadays, multi-objective optimisation is an important research field since many real-world problems involve optimising many objective functions simultaneously [1]. These problems are known as Multi-objective Optimisation Problems (MOPs), and their solution gives rise to a set of trade-off solutions, commonly known as Pareto-optimal solutions [2], characterised by the feature that their objectives are usually in conflict with each other (i.e., one objective cannot be improved without deteriorating another one). Multi-objective Evolutionary Algorithms (MOEAs) have been widely used for solving MOPs because they have the ability to deal with a set of possible solutions at the same time, allowing them to obtain an approximation of the Pareto frontier in a single run. This capability is an advantage of MOEAs over conventional multi-objective programming methods, which need to perform a set of separate single-objective optimisations to generate a set of compromise solutions [3]. Additionally, MOEAs are more robust regarding the mathematical characteristics of the objective functions and their constraints [4].
According to Bechikh et al. [5], some of the challenges faced by MOEAs when the number of objective functions increases are the following:

1) Ineffectiveness of the genetic operators (crossover, mutation and selection; e.g. [6,7]).
2) Remarkable difficulties to represent the Pareto front since many points are necessary to do it.
3) High computational cost to determine the extent of crowding (diversity measure estimation) of a solution in a population.
4) Difficulty of visualization of a high-dimensional Pareto front.
5) Increase in the number of non-dominated solutions, making it hard to obtain a representative sample of the Pareto front.
6) Increase in the number of the Dominance Resistant Solutions (DRSs), which according to Ikeda et al. [8], are solutions with a poor value in at least one of the objectives, but with near-optimal values in the remaining objectives, thus being very hard to be dominated.

Although MOEAs generate a set of efficient solutions, only one of these will be chosen as the final option to be implemented. Thus, besides finding Pareto-optimal solutions, it is equally important to provide support in the decision-making process. The Decision Maker (DM) is the entity in charge of choosing a single option (the best compromise) to be implemented [3].
As is widely known, preferences can be incorporated in three different stages: a priori, progressively (interactive) and a posteriori.
An a posteriori incorporation of preferences rests on two main assumptions: A) the approximation to the Pareto Frontier identified by the metaheuristic contains a representative subset of the Region of Interest (ROI) [9], i.e., the zone of the Pareto frontier more in agreement with the DM’s preferences; this means that no better solutions lied out of the known Pareto Frontier); B) the DM is able to make consistent judgments when compares solutions on the known Pareto Frontier, and hence (s)he can identify the best compromise.

Concerning Assumption B), the identification of the best compromise from a set of solutions could be an easy task in problems with two or three objective functions but, when these increase, this task becomes very hard due to the human mind cognitive restrictions. According to Miller [10], the human mind is limited to processing a small amount of information simultaneously. This condition is a severe obstacle for identifying the best compromise from a set of solutions, mainly in the problems having many objectives, since it is beyond the cognitive abilities of an average DM. The DM often selects the first solution that seems to match his/her aspiration levels, without making suitable judgments of the other alternatives. Additionally, the appropriateness of Assumption A) depends on the nature of the problem, its number of variables and objective functions and the specific metaheuristic used. But given a particular problem which has not been addressed before, this assumption is very strong.
The a priori and interactive incorporation of preferences can reduce the search space, filter non-dominated solutions and help the search to identify solutions closer to the Region of Interest. This is a real advantage in comparison with the posterior preferences incorporation.

There are many approaches in the literature that have incorporated this preference information into the optimisation process in order to direct it only towards the ROI and avoid unnecessary exploration of the entire search space. According to Bechikh [11], the information structures most often used to incorporate the preferences are the following:

- *weights* (e.g. [12,13]),
- *ranking of solutions* (e.g. [14,15]),
- *ranking of objective functions* (e.g. [16,17]),
- *reference point* (e.g. [18]),
- *trade-offs between objective functions* (e.g. [19]),
- *desirability thresholds* (e.g. [20]) and,
- *outranking relations* (e.g. [21,22]).

In our opinion, a more complete and systematic taxonomy of the methods for preference incorporation should cover the following issues:

- the model of aggregating multi-criteria preferences underlying behind the way in which preferences are incorporated (e.g. value functions, outranking relations, trade-offs between objective functions);
- the stage in which the preference information is articulated (e.g. a priori, progressively);
- the cognitive process required from the DM (e.g. ranking of solutions, pairwise comparisons, ordinal classification).

Some requirements for the DM are implicit in the above issues. For example: i) an a priori incorporation of preferences assumes that the DM has, before the optimisation process, a rather well-formed knowledge about his/her problem and preferences, and (s)he can set the decision model parameters ii) the use of value functions, ranking of solutions or pairwise comparisons assume that the DM can make decisions on sets of solutions (comparability), and with transitive preferences; iii) most interactive methods require also comparability and transitivity of preferences; iv) a cognitive process like ranking of solutions or pairwise comparisons may be very demanding for the DM, mainly due to his/her cognitive limitations in problems with many objectives [10]; this is even harder in interactive frameworks.

Interactive methods are more popular than methods with a priori articulation of preferences, because within an interactive framework the DM learns about his/her problem, discovers new solutions and goes inside the complex trade-offs among his/her objectives; such a learning process helps the DM to choose more appropriate settings of the decision model parameters. As stated by Miettinen [23], the DM can specify and correct his/her preferences and gain a better knowledge of his/her problem and its potentialities. Also, the DM should be more confident with the final results, since (s)he has been involved in the search process. However, most interactive methods require comparability and transitive preferences from the DM (cf. [24]). Unfortunately, as a consequence of the human cognitive limitations, there is abundant evidence about non-transitive judgments and incomparability situations in real decision making processes (see [25]).

In our opinion, the following features are desirable for a method of preference incorporation:

a) an easy interaction between the DM and the solution generator algorithm; this implies minimizing the cognitive effort from the DM when (s)he makes judgments about solutions.
b) no requirement of comparability and transitivity of preferences;
c) the model of aggregation of multi-criteria preferences should be compatible with relevant characteristics of real DMs;
d) there should be techniques to infer the decision model parameters from decision examples provided by the DM during his/her learning process.

Concerning preference incorporation, the cognitive process with the lowest cognitive demand on the DM is perhaps the classification into two ordered classes (also called categories in the related literature). Assigning solutions to classes 'good' and 'no good' does not require transitive preferences; comparability among 'good' and 'no good' solutions suffices. In an interactive process, assigning solutions to these classes is clearly less cognitively demanding than pairwise comparisons, ranking of solutions, or judgments about closeness to certain desired goals.

The use of outranking relations is a way to handle characteristics of many decision makers facing real world problems. Methods based on outranking relations are recommended to address problems that present any of the following characteristics [26]: i) at least one of the evaluation criteria is non-quantifiable, i.e., it is measured on an ordinal or qualitative scale; ii) it has criteria of heterogeneous nature; iii) compensation between criteria is not generally accepted (veto situations are possible) and iv) at least in one criterion the following is true: small differences in the evaluations are not significant in terms of preference, while the accumulation of many small differences becomes important. Non-transitive preferences and incomparability situations are consequences of points iii) and iv).

Outranking methods have been criticized for the difficulty to elicit the whole set of model parameters. However, there are several works that apply preference-disaggregation

approaches to make an indirect elicitation of outranking model parameters from assignment examples [27–29].
According to the above discussion, we strongly believe that outranking-based ordinal classification methods are good options to fulfill the four desirable features for any of the methods of preference incorporation stated above.

To the best of our knowledge, the first paper in using outranking-based multi-criteria ordinal classification was Oliveira et al. [30]. In this paper, the popular ELECTRE-TRI method was used for ordinal classification in a three-objective problem. The preferences were incorporated a priori, setting directly the outranking model parameters.
Recently, Cruz et al. [31] proposed an hybrid algorithm using outranking-based multi-criteria ordinal classification. It works on three phases. During the first phase, a metaheuristic algorithm (appropriate to the problem to be solved) obtains an approximation to the Pareto frontier. In the second phase (interactive), the DM assigns the solutions to two ordered classes ('satisfactory' and 'non satisfactory'), and the outranking model parameters are elicited by him/her. In the third phase, the THESEUS multi-criteria ordinal classification method is used to make selective pressure towards 'satisfactory' solutions. The first phase was implemented with NSGA-II and NOACO, a recently proposed Ant Colony Algorithm [4]. The proposal was tested in project portfolio problems with 4, 9 and 16 objectives; its results outperformed NSGA-II and the ant colony algorithm [4].
Four main limitations of [31] can be identified:

- Although in several instances and on a wide range of objectives, the proposal was only tested on non-interacting project portfolio optimisation problems;
- The lack of knowledge about the true Pareto front of the project portfolio optimisation problems prevents an appropriate evaluation of the quality of solutions; there was no information about the closeness to the region of interest;
- No state of the art representative metaheuristics were used during the first phase; neither for comparison of results;
- The proposal in [31] is not really interactive. There is only one step in which the DM assigns solutions to ordered categories. This does not allow the learning and preference updating process which is typical of interactive methods.The learning capacity provided by the method should be enriched through other interaction steps.

As a consequence of the above limitations, the work in Cruz et al. [31] left some open questions:

1. Is the high quality of its solutions kept in a wide range of difficult problems, different from the project portfolio optimisation problem?
2. Does the method outperform benchmark metaheuristics?
3. To what extent is the closeness to the region of interest degraded by increasing the number of objectives?
4. To what extent is the method effectiveness affected by the selection of a particular multi-criteria ordinal classification approach?

The above questions are addressed by the present paper. Instead of NSGA-II and NOACO, MOEA/D (MultiObjective Evolutionary Algorithm based on Decomposition) and

MOEA/D-DE (MOEA/D based on Differential Evolution) are used as metaheuristics in the first phase of the hybrid algorithm; they can be considered as metaheuristics representative of the state of the art, useful for an a posteriori incorporation of preferences. In addition to some instances of project portfolio optimisation problems, our proposal was evaluated on scalable test problems (the DTLZ test suite), using three and eight objectives; the use of the DTLZ test suite allows us to evaluate the closeness to a simulated ROI, and compare the performance with three and eight objectives. The DM's preferences are simulated through an outranking model. In addition to the THESEUS method, here we use the popular ELECTRE-TRI, and the results from both methods are compared.

This paper is organized as follows: Section 2 presents a brief summary of some evolutionary algorithms related to this work, a model of preferences based on outranking relations and the ELECTRE-TRI and THESEUS methods. The proposed algorithm is described in Section 3. The experimentation and results are shown in Section 4. Our concluding remarks are presented in Section 5.

2 Background

2.1 Some multi-objective approaches related to this work

In this section, two recent algorithms, used in the comparison with the approach presented here, will be briefly described.

2.1.1 MOEA/D

It is a MOEA based on a decomposition approach [32]; any decomposition method can be used. MOEA/D decomposes a MOP through aggregation functions into a number of optimisation subproblems, and optimises them at the same time. MOEA/D needs the following input data:

- a stopping condition;
- *N*, the number of subproblems and the population size;
- *N*, weight vectors uniformly distributed;
- *T*, the number of weight vectors in the neighborhood of each vector.

MOEA/D gives as output an external population (EP), used to store non-dominated solutions found during the optimisation process. The steps considered by the algorithm are concisely described below.

The first step is the initialisation where the following actions are carried out: the EP starts empty; the Euclidean distance between any two weight vectors is calculated and after that, the *T* closest weight vectors to each vector are related to it; an initial population is created and a reference point *z* is initialized. The second step is the update where, for each solution in the population, the following is done: two solutions of the population are randomly selected and used to generate a new solution by applying genetic operators; the new solution can be repaired or improved. The reference point *z* is updated taking into consideration the new solution; a population member can be replaced by the new solution when it obtains a better aggregation function value; finally, the EP is updated by removing from it all the solutions dominated by the new solution and, adding to the EP the new solution if it is non-dominated in the EP. The third step is to check the stopping condition; if it is satisfied, the algorithm finishes and reports the EP as its output; otherwise, the algorithm returns to the second step.

2.1.2 MOEA/D-DE

It is a new version of MOEA/D to deal with continuous MOPs [33]. It uses a differential evolution operator and a polynomial-based mutation operator for creating new solutions. The algorithm, additionally to MOEA/D input, requires the following:

- δ, the probability that parents are picked up from the neighborhood;
- n_r, the maximum number of solutions replaced by each new solution.

As output, the algorithm gives an approximation to the Pareto front. MOEA/D-DE considers three steps as MOEA/D. The first step has the same actions as MOEA/D, except that it does not initialize an external population. The second step is the update where, for each solution in the population, the following is done: a range of mating or update is selected taking into account the δ value; two solutions are randomly selected on it, and are used to create a new child by applying the differential evolution operator and the polynomial-based mutation operator. If an element of the new solution is out of the decision space, its value is reset to be inside the boundary. Also, the reference point z is updated considering the new solution; the population solutions are updated using the value of n_r and the aggregation function value. The third and last step consists in checking the stopping criterion.

2.2 The preference model

The model of preferences proposed by Fernandez et al. [3] uses the fuzzy outranking relation suggested by Roy [34] for modeling crisp preference relations. The crisp outranking relation represents the statement 'x is at least as good as y' and is denoted by xSy (x outranks y). The definition of the preference relations rests on the degree of credibility of xSy denoted by $\sigma(x,y)$. This value of truth σ can be calculated using outranking methods, such as ELECTRE-III [34,35] and PROMETHEE [36]. The determination of one preference relation between a pair of solutions is based on a threshold of acceptable credibility λ (determines a level of requirement of the outranking relation), a symmetry parameter ε, and an asymmetry parameter β. For each pair of alternatives (x,y), the model establishes one of the preference relations shown in Table 1.
The strict preference is defined when the DM has clear and well-defined reasons to prefer x over y. The indifference occurs when the DM notes a high degree of equivalence between x and y. The weak preference takes place when the DM hesitates between xPy or xIy. The incomparability arises when the DM perceives a high level of conflicting information when x is compared with y. The k-preference happens when the DM hesitates between xPy and xRy.

Table 1. The binary preference relations between each pair of solutions (x, y)

Outranking relation	*Conditions*[*]	
Strict preference (xPy)	x dominates $y \vee (\sigma(x, y) \geq \lambda \wedge \sigma(y, x) < 0.5) \vee (\sigma(x, y) \geq \lambda \wedge [0.5 \leq \sigma(y, x) < \lambda] \wedge [\sigma(x, y) - \sigma(y, x)] \geq \beta)$	(1)
Indifference (xIy)	$\sigma(x, y) \geq \lambda \wedge \sigma(y, x) \geq \lambda \wedge \lvert\sigma(x, y) - \sigma(y, x)\rvert \leq \varepsilon$	(2)
Weak preference (xQy)	$\sigma(x, y) \geq \lambda \wedge \sigma(x, y) > \sigma(y, x) \wedge \neg\, xPy \wedge \neg\, xIy$	(3)
Incomparability (xRy)	$\sigma(x, y) < 0.5 \wedge \sigma(y, x) < 0.5$	(4)
k-preference (xKy)	$0.5 \leq \sigma(x, y) < \lambda \wedge \sigma(y, x) < 0.5 \wedge \sigma(x, y) - \sigma(y, x) > \beta/2$	(5)

*Considering ($0 \leq \varepsilon \leq \beta \leq \lambda \leq 1$ and $\lambda > 0.5$).

Let us consider a set of feasible solutions O; the preference model establishes the sets and measures shown in Table 2.

Table 2. Sets and measures defined by the preferential system

Set / Measure	*Definition*	
$S(O, x) = \{y \in O \mid yPx\}$	Solutions that strictly outrank x	(6)
$NS(O) = \{x \in O \mid S(O, x) = \varnothing\}$	Called the non-strictly-outranked frontier	(7)
$W(O, x) = \{y \in NS(O) \mid yQx \vee yKx\}$	Non-strictly-outranked solutions that are weakly preferred to x	(8)
$NW(O) = \{x \in NS(O) \mid W(O, x) = \varnothing\}$	Named as the non-weakly-outranked frontier	(9)
$F_n(x) = \sum_{y \in NS(O) \setminus \{x\}} [\sigma(x, y) - \sigma(y, x)]$	The net flow score is an additional criterion to detect preferred solutions by the DM on the non-strictly-outranked frontier, where $F_n(x) > F_n(y)$ shows a certain preference for x over y	(10)
$F(O, x) = \{y \in NS(O) \mid F_n(y) > F_n(x)\}$	Non-strictly-outranked solutions that have a greater net flow than x	(11)
$NF(O) = \{x \in NS(O) \mid F(O, x) = \varnothing\}$	Denominated as the net-flow non-outranked frontier	(12)

In this paper, the degree of truth $\sigma(x,y)$ is calculated as the ELECTRE-III method [34] (with some simplifications) and is used in the same way along the whole document. The computation of σ is detailed below:

Let us consider that (x,y) is a pair of solutions defined by a set of N objective functions, $G = \{g_1, g_2, \ldots, g_j, \ldots, g_N\}$. Then,

$$\sigma(x,y) = c(x,y) \cdot N(d(x,y)), \tag{13}$$

PREPRINT

where $c(x,y)$ expresses the degree of truth of the assertion "there is a strong coalition of criteria in favor to xSy". $N(d(x,y))$ expresses the degree of truth of the assertion "there is no strong opposition to xSy".
This concordance degree $c(x,y)$ is computed using a set of weights w_j ($w_1 + w_2 + \ldots + w_N = 1$) and an indifference threshold q_j for each j-th objective as follows:

$$c(x,y) = \sum_{j: g_j \in C_{xy}} c_j(x,y) \tag{14}$$

where

$$c_j(x,y) = \begin{cases} w_j & iff\ \ xP_jy \ \vee\ \ xI_jy \\ 0 & \quad\text{otherwise} \end{cases} \tag{15}$$

such that $\quad xP_jy = g_j(x) > g_j(y)\ \wedge \neg xI_jy,$ (16)

$$xI_jy = \left|g_j(x) - g_j(y)\right| \le q_j$$

P and I are the predicates of strict preference and indifference, respectively, when the j-th objective is evaluated.
The criterion set $C_{xy} = \{g_j \in G \mid x_jP_jy_j \vee x_jI_jy_j\}$ is the so-called concordance coalition with xSy. The set $G–C_{xy}$ is the discordance coalition with xSy.

The marginal discordance index $d_j(x,y)$ measures how much each criterion disagrees with the statement xSy. This index is calculated using a veto (v_j) and a pre-veto (u_j) thresholds assigned to the j-th objective (see Eq. (17),

$$d_j(x,y) = \begin{cases} 0 & iff\ \nabla_j \le u_j \\ (\nabla_j - u_j)/(v_j - u_j) & iff\ u_j < \nabla_j < v_j \\ 1 & iff\ \nabla_j \ge v_j \end{cases} \tag{17}$$

Finally, the degree of truth of the predicate about the weakness of the discordance coalition is calculated by using the "min" operator combined with the strict negation "1–*" operator as follows:

$$N(d(x,y)) = \min_{j: g_j \in D_{xy}} \{1 - d_j(x,y)\} \tag{18}$$

where
$D_{xy} = G – C_{xy}$;
$\nabla_j = g_j(y) - g_j(x)$.

2.3 A brief outline of two multi-criteria ordinal classification methods

2.3.1 The THESEUS method

The THESEUS method [37] is an approach based on outranking relations to solve multi-criteria sorting problems. The term sorting refers to the assignment of a set of alternatives to preference-ordered classes, which are predefined in an ordinal way [38]. THESEUS assigns multi-criteria objects to an element of the set of ordered classes by using the information from various preference and indifference relations. These relations are determined from an outranking relation defined on the universe of objects. The assignment is not a result of the object's intrinsic characteristics, but rather a consequence of comparisons with other objects whose assignments are known. The THESEUS method is based on the following premises:

- There is a limited number of ordered classes $Ct = \{C_1, \ldots, C_M\}$, $(M \geq 2)$; where C_M indicates to be the best class.
- U represents the universe of objects x characterised by a set of N real-valued criteria, indicated as $G = \{g_1, g_2, \ldots, g_j, \ldots, g_N\}$, where $N \geq 3$.
- There is a set of reference objects T (also named *reference set* or *training set*), which is formed of objects $b_{kh} \in U$ assigned to classes C_k, $(k = 1,\ldots, M)$.
- There is an outranking relation $\sigma(x,y)$ defined on $U \times U$ (see Section 2.2), which models the degree of credibility of the declaration 'x is at least as good as y' from the DM's point of view.

The assignment of object x to a potential class is denoted as $C(x)$. In line with THESEUS premises, $C(x)$ should fulfill:

$$\forall x \in U,\ \forall b_{kh} \in T$$

$$xPb_{kh} \Rightarrow C(x) \gtrsim C_k \qquad (19.a)$$

$$b_{kh}Px \Rightarrow C_k \gtrsim C(x)$$

$$xQb_{kh} \Rightarrow C(x) \gtrsim C_k \qquad (19.b)$$

$$b_{kh}Qx \Rightarrow C_k \gtrsim C(x)$$

$$xIb_{kh} \Rightarrow (C(x) \gtrsim C_k) \wedge (C_k \gtrsim C(x)) \Leftrightarrow C(x) = C_k \qquad (19.c)$$

The symbol $\gtrsim$ expresses the statement 'is at least as good as' on the set of classes, which is associated with the decision-making framework. The relations P, Q, and I were previously formalized in Eqs. (1–3). THESEUS makes use of the inconsistencies with Eqs. (19.a–c) to examine the possible assignments of x. The sets of inconsistencies are defined below:

- The set of P-inconsistencies for x and $C(x)$ is defined as $D_P = \{(x,b_{kh}), (b_{kh},x), b_{kh} \in T$ such that (19.a) is FALSE$\}$;
- The set of Q-inconsistencies for x and $C(x)$ is defined as $D_Q = \{(x,b_{kh}), (b_{kh},x), b_{kh} \in T$ such that (19.b) is FALSE$\}$;
- The set of I-inconsistencies for x and $C(x)$ is defined as $D_I = \{(x,b_{kh}), (b_{kh},x), b_{kh} \in T$ such that (19.c) is FALSE$\}$.

Let us suppose that $C(x) = C_k$ and $b_{jh} \in T$. Some I-inconsistencies can be classified as follows:

- Second-order I-inconsistencies (D_{2I}): they occur when x and b_{jh} belong to adjacent classes and however xIb_{jh}; then, this kind of inconsistency may be explained by 'discontinuity' of the description; x may be close to the upper (lower) boundary of C_k and b_{jh} may be close to the lower (upper) boundary of C_j.
- First-order I-inconsistencies (D_{1I}): they occur when x and b_{jh} do not belong to adjacent classes and however xIb_{jh}. $D_{1I} = D_I - D_{2I}$.

Let $N_1 = n_P + n_Q + n_{1I}$, and $N_2 = n_{2I}$ where n_P, n_Q, n_{1I}, and n_{2I} represent the cardinalities of inconsistency sets specified before. THESEUS recommends an assignment that minimizes the above inconsistencies with lexicographic priority for N_1, which is the most relevant criterion. The THESEUS assignment rule is shown in Algorithm 1.

Algorithm 1. THESEUS assignment rule.

Input: U, T
Output: objects assigned to classes
Initialize: $k \leftarrow 1$, assign the minimum credibility level $\lambda > 0.5$, $C_j \leftarrow \emptyset$, $C_{aux} \leftarrow \emptyset$

  For each $x \in U$ **do**
    Do
      For each $b_{kh} \in T$ **do**
        $C_{aux} \leftarrow$ Calculate $N_1(C_k)$
      $k \leftarrow k + 1$
    Until $k \leq M$
    $C_j = argmin\,\{C_{aux}\}$
    $C_{k*} = \underset{\{C_j\}}{argmin}\{N_2(C_i)\}$
    Assign x to C_{k*}

The outcome of applying the assignment rule may be i) a single class which is called a 'precise assignment' or ii) a series of classes (multi-class) which is called an 'imprecise assignment'. Multi-class emphasizes the highest (C_H) and the lowest (C_L) class; every class in this interval may be adequate for the assignment of the object. In this paper, we adopt a 'pessimistic' attitude; the object will be assigned to C_L.

2.3.2 The ELECTRE-TRI method

ELECTRE-TRI is one of the methods of the ELECTRE (ELimination Et Choix Traduisant la REalité) family, which are based on the construction and exploitation of an outranking relation applied to different problems (choice, ranking, and sorting) in multi-criteria decision analysis. ELECTRE-TRI is a multi-criteria ordinal classification method that assigns a set of actions (solutions) $A = \{a_1, a_2, \ldots, a_l\}$ to predefined ordered classes. The assignment process is based on the DM's preferences which are modeled by a set of parameters, and is described below. The parameters are preference (p_j), indifference (q_j), and veto (v_j) thresholds; a set of weights (w_j), where j=1,…,N and N indicates the number of criteria; a set of reference profiles (b_i), where i=1,…,n and n represents the total number of reference profiles; and a cutting-level $\lambda \in [0.5, 1]$. Each reference profile is described by N criterion values.
There are n+1 classes, where C_1 is the worst class and C_{n+1} is the best one. Each class C_i is bounded by two reference profiles (b_{i-1} is the lower and b_i is the upper one) where the upper profile of a class corresponds to the lower profile of the next class. The profiles b_0 and b_{n+1}

correspond to the ideal and anti-ideal profiles, respectively. All these elements are illustrated in Fig. 1.

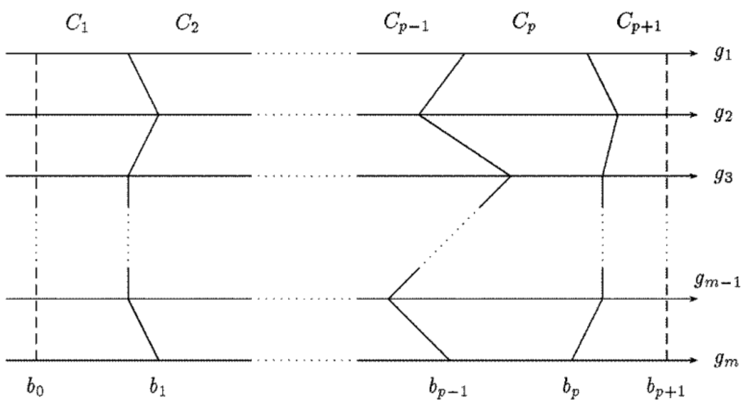

Fig. 1. Denotation of the classes and reference profiles [39]

The assignment process of an action *a* to a class results from the comparison made between the action *a* and the reference profiles established by the DM. This process is performed in two main consecutive steps:

- the construction of an outranking relation *S*, as described in Section 2.2, that defines how solutions compare to the reference profiles, and
- the exploitation of this relation in order to assign each solution to a precise class.

Exploitation procedure

This process consists in determining the way in which an action *a* is compared to the reference profiles in order to define the class to which *a* should be assigned. The following two procedures can be used:

Pessimistic procedure

a) Compare a to b_i, for $i = n, n\text{-}1, \ldots, 0$,
b) Action a will be assigned to class C_{i+1} ($a \rightarrow C_{i+1}$) for which b_i is the first profile such that aSb_i ($aSb_i \Leftrightarrow \sigma(a,b_i) \geq \lambda$).

Optimistic procedure

a) Compare a to b_i, for $i = 1, 2, \ldots, n$,
b) Action a will be assigned to class C_i ($a \rightarrow C_i$) for which b_i is the first profile such that $b_i \succ a$ ($b_i \succ a \Leftrightarrow b_iSa$ and *not* (aSb_i)).

It is worth mentioning that the pessimistic procedure is used in this work. More information regarding the ELECTRE-TRI method can be found in [40].

PREPRINT

3 A multi-criteria ordinal classification method within an evolutionary algorithm

The proposed approach aims to incorporate the DM's preferences into an evolutionary algorithm to guide the search process towards solutions more in agreement with his/her preferences, that is, the region of interest (ROI). In this paper, the solutions in the ROI are characterised by being non-dominated and belonging to the most preferred class. The DM's preference information is reflected on a set of parameters, a reference profile, and a reference set. The proposed procedure uses this preference information in a multi-criteria ordinal classification method, which is included in an evolutionary approach to lead the search towards the ROI. New solutions generated by the search process are assigned to a class by the multi-criteria classification method. In this work, ELECTRE-TRI [40] and THESEUS [37] are used independently as multi-criteria ordinal classification methods. The selective pressure of the evolutionary approach is strengthened by using the classification of the solutions. Our proposed approach is a hybrid algorithm consisting of three main phases: 1) initialisation of reference solutions; 2) the preference elicitation phase, and 3) the selective pressure phase. The purpose of the first phase (initialisation) is to obtain a set of solutions. In the second phase (elicitation), the solutions of that set are classified by the DM, that is, (s)he express his/her judgments. This set is used to identify some good solutions during the third phase (selective pressure). Once the third phase finishes, the DM improves his/her understanding about what a satisfactory solution is and (s)he can update his/her preferences, returning to the second phase. Then, new assignments are made and the preference model is updated. The process can be repeated until the DM feels satisfaction with the solutions in the most preferred class. The hybrid approach is called Hybrid Evolutionary Algorithm guided by Preferences (HEAP) and, depending on whether the ELECTRE-TRI or the THESEUS method is used, HEAP-ELECTRE or HEAP-THESEUS, respectively. Fig. 2 illustrates the HEAP algorithm and schematizes in detail the relation between its phases. Section 3.1 describes the first two phases and section 3.2 describes the third one.

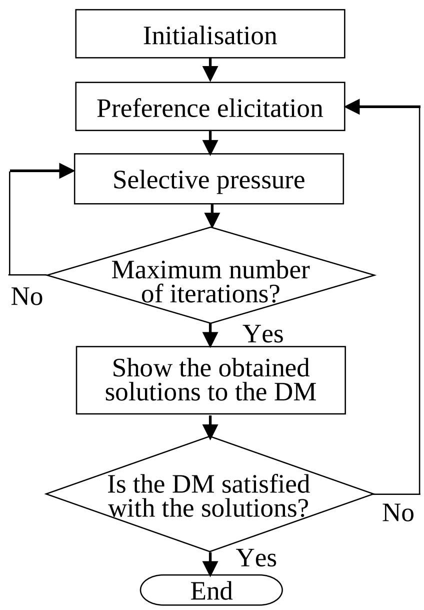

Fig. 2. Hybrid Evolutionary Algorithm guided by Preferences (HEAP)

3.1 Initialisation and Preference elicitation phase

The purpose of the first two phases is to obtain i) a whole set of outranking model parameters (criterion weights and different thresholds); ii.1) a reference profile as limiting boundary between the ordered classes of solutions when ELECTRE-TRI is used; or ii.2) a reference set with solutions assigned to the pre-defined classes when THESEUS is used. The classes considered in this work are two: '*satisfactory*' and '*unsatisfactory*'. The initialisation and preference elicitation phases are illustrated in Fig. 3. When the hybrid procedure starts, the first step is to run a multi-objective metaheuristic to generate a subset of the approximated Pareto frontier (Processes 1−2 in Fig. 3). It should be mentioned that any metaheuristic can be used in the first phase. The output set of this metaheuristic will be assigned by the DM to the previously defined classes. Once this has been done, the DM will make a direct or an indirect elicitation of preference model parameters; in case of ELECTRE-TRI, these parameters are weights, thresholds and the limiting profile; in case of THESEUS, only weights and thresholds (Processes 3−4 in Fig. 3). In successive interactions, if needed, the DM makes judgments on solutions provided by the third phase and updates model parameters (Processes 3−4 in Fig. 3).

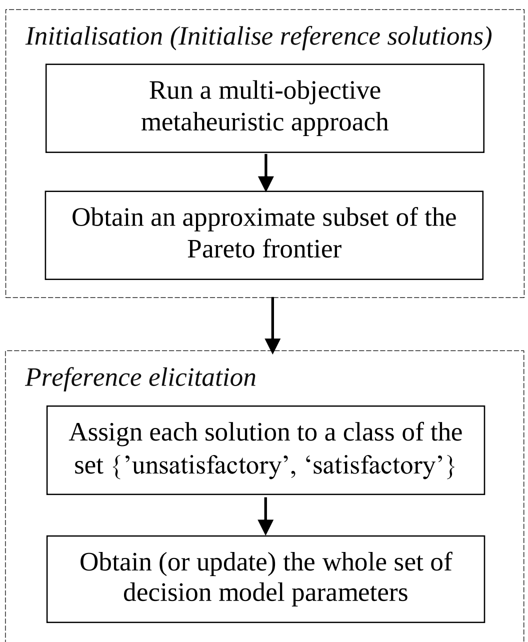

Fig. 3. Flowchart of the first two phases of HEAP

When direct elicitation is used, the DM is in charge of making a direct setting of the preference parameters values. According to Fernández et al. [29], the direct elicitation of veto thresholds can be a very hard task when ELECTRE type methods are used, since these parameters are very unfamiliar to typical decision makers. Particularly, ELECTRE-TRI has been criticized because of the difficulties coming from the appropriate definition of the limiting profiles. If a direct elicitation is performed within the first phase of the hybrid algorithm, the elicited model should be in agreement with the assignment of solutions (made by the DM) to the satisfactory and unsatisfactory classes; that is, the preference model should be consistent with the assignments made by the DM to avoid contradictions.

The alternative is the use of indirect elicitation methods. These are based on regression-inspired procedures for inferring the model's parameters from a set of decision examples. In this paradigm, the elicited model agrees with the known DM's preferences expressed through the set of decision examples. Doumpos et al. [27] and Fernández et al. [29] proposed metaheuristic algorithms to infer the whole set of the ELECTRE-TRI model parameters from assignment examples. Covantes et al. [28] proposed an evolutionary algorithm to infer the THESEUS model parameters.

Let us remark that the experiments shown in this paper were performed without a real DM. Then, the DM was simulated by using the outranking model described in Section 2.2. Thus, the first phase of the hybrid algorithm is performed without real human judgments; the experiments work as in a direct elicitation method in which an appropriate set of model parameters can be identified a priori. Algorithm 2 illustrates this process. The simulation proceeds as follows: let us take the solutions obtained by the metaheuristic that runs in the first phase, to create a reference set by using the outranking model described in Section 2.2; this set will be used by THESEUS. The satisfactory class is created with some solutions that belong to the non-strictly-outranked frontier (Line 2) and fulfill one of the following conditions: i) to belong to the non-weakly and net-flow non-outranked frontier (Line 7), (this set is called the 'Best Compromise' (BC)); ii) to be indifferent to any solution in BC (Line 13); iii) to be a non-dominated solution (minimization) with respect to the counting of weakness (Eq. 8) and net flow score (Eq. 10), considered as two indirect objectives (Line 17); iv) to be a solution with a positive net flow score (Line 21). The unsatisfactory class is created with some solutions that do not fulfill any of the conditions stated above.

On the other hand, the solution considered as reference profile (called 'ref*_profile'* in Line 25), which will be used by ELECTRE-TRI to identify the boundary between classes, is the one that meets the condition of belonging to the non-strictly-outranked frontier and being the last solution that has a positive net flow score (Line 25), without being a BC, indifferent to some BC, or a non-dominated solution with respect to the objectives' weakness count and net flow score. We consider that this condition is enough to define the boundary between satisfactory and unsatisfactory classes.

Algorithm 2. Procedure for creating a reference set and a reference profile through a simulated DM.

```
Input: A ← subset of the approximate Pareto frontier generated by any metaheuristic
Output: a reference set {satisfactory, unsatisfactory}, a reference profile {ref_profile}
Initialize: satisfactory ← Ø, unsatisfactory ← Ø, ref_profile← Ø, bestCompromise ← Ø, temp ← Ø
For each x ∈ A do
    If |S(A, x)| = 0 then                                        // Eq. (6)
        temp ← x
    else
        unsatisfactory ← x
For each x ∈ temp do
    If |W(A, x)| = 0 and |F(A, x)| = 0 then                      // Eq. (8) and (11) resp.
        bestCompromise ← x
        Delete x from temp
satisfactory ← bestCompromise
For each x ∈ temp do
    For each y ∈ bestCompromise do
        If xIy then                                              // Eq. (2)
            satisfactory ← x
            Delete x from temp
For each x ∈ temp do
    Compute dominance (min) of x in temp with respect to |W(A, x)| and |F(A, x)| as objectives
    If x is non-dominated in temp then
        satisfactory ← x
        Delete x from temp
For each x ∈ temp do
    If net_flow of x > 0 then                                    // Eq. (10)
        satisfactory ← x
        Delete x from temp
ref_profile ← satisfactory[last]
unsatisfactory ← temp
```

3.2 Selective pressure phase

The selective pressure phase aims to lead the search for solutions that are non-dominated and satisfactory. To achieve this, we include multi-criteria classification in an evolutionary algorithm, which ranks its population in preference-ordered fronts (see Fig. 4). The search for non-dominated solutions is carried out as in NSGA-II, by creating non-dominated fronts. The difference in our approach is that the solutions of the first front are classified by a multi-criteria classification method. After classification, performed either by ELECTRE-TRI or THESEUS, the first front is divided into two sub-fronts, i.e., one front per each class. Therefore, the first front is formed now by non-dominated solutions that belong to the best class (satisfactory), making selective pressure towards the ROI. Algorithm 3 presents the selective pressure process.

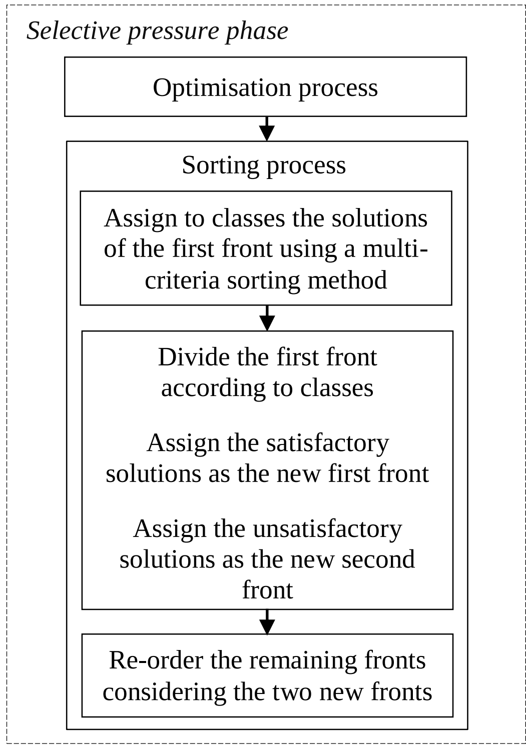

Fig. 4. Selective pressure towards the ROI. Process using preferences to enhance selective pressure towards dominated and satisfactory solutions

---

Algorithm 3. Selective pressure towards the ROI

Input: output of the Algorithm 2
Output: the new first front
Initialize: *first_front* ← Ø, *second_front* ← Ø
1. Create population ← parent ∪ offsprings
2. Create the fronts by non-dominance
3. Sort the first front using a multi-criteria sorting method
4. Divide the first front into two fronts by grouping solutions with the same class
5. *first_front* ← non-dominated solutions assigned as satisfactory
6. *second_front* ← non-dominated solutions assigned as unsatisfactory
7. Reorder the remaining fronts considering the two new fronts

---

4 Experimental results

In this section, we describe the problems used as case study as well as the experimental conditions used for the analysis of the results. Let us underline that, in the absence of a real DM, we renounce to implement the learning process provided by several iterations of the elicitation preference phase.
The performance of the proposed approach was tested on project portfolio optimisation problems and seven scalable DTLZ test problems. By using the DTLZ test problems we were able to evaluate the quality of the solutions concerning the closeness to the region of interest. Two metaheuristics representative of the state of the art MOEA/D and MOEA/D-DE, were used for the first phase of the hybrid approach and for comparison of results.

The reference sets used by THESEUS are composed of several elements per class. ELECTRE-TRI uses only one element as reference profile on the boundary between classes. The preferential model parameters (λ=0.67, β=0.2 and ε=0.1) have the same values as those suggested by Fernandez et al. [3].
All algorithms used in the experiments were run 30 times for each instance on an Intel CORE i7, 2.80 GHz processor with 16 GB of RAM. Our hybrid approach was developed in the Java language, using the JDK 1.8 compiler, and NetBeans 8.0.2 as IDE.

4.1 Scalable test problems

This kind of problems has been widely used in the field of multi-objective evolutionary algorithms mainly to assess the performance of new algorithms and to compare different approaches. The scalability to any number of objectives and decision variables, and knowledge of the particular form and position of the resulting Pareto-optimal front, are only some of the features of these problems. We address the DTLZ1–DTLZ7 problems with three and eight objectives. The metaheuristic used in the elicitation phase of preferences to create the reference profile and the reference set, was MOEA/D-DE and was applied on both problems with three and eight objectives. We have used MOEA/D-DE because it has shown the ability to solve problems with a few and many objectives. For the same reason, our hybrid approach is compared with respect to MOEA/D-DE.

4.1.1 Parameters settings

Control parameters used specifically by MOEA/D-DE are shown in Table 3. Their values were set according to the original paper [33]. Table 4 shows the crossover parameters values used particularly by HEAP; they were retrieved from [41].

Table 3. Control parameters used in MOEA/D-DE

Parameter	*Values*
CR	1.0
F	0.5
p_m	$1/n$
T	20
δ	0.9
n_r	2

Table 4. Crossover parameters values used by HEAP

Parameter	*Values*
SBX crossover probability p_c	1
SBX crossover index η_c	30

Some parameters values used in the optimisation phase are the same for both algorithms and were set according to [41,42]; these are provided by Table 5. In three-objective problems, the population size and the number of weight vectors were both set to 91. In eight-objective problems, the population size and the number of weight vectors were both set to 330. These values were the same for HEAP and MOEA/D-DE and were set according to [41,42]. It should be mentioned that for the same problem size (number of objectives), we adopted the same number of iterations for both algorithms. However, as HEAP has an Initialisation and a Selective pressure phases, for a fair evaluation, the total number of iterations is divided between them by assigning half of the iterations to each phase. Polynomial mutation values used by HEAP and MOEA/D-DE were retrieved from [41] and are shown in Table 6.

Table 5. Parameters used by HEAP and MOEA/D-DE in the optimisation phase

Problem	*Three objectives*		*Eight objectives*	
	No. var.	*Iterations*	*No. var.*	*Iterations*
DTLZ1	7	400	12	750
DTLZ2	12	250	17	500
DTLZ3	12	1000	17	1000
DTLZ4	12	600	17	1250
DTLZ5	12	500	17	1500
DTLZ6	12	500	17	1500
DTLZ7	22	500	27	1700

Table 6. Polynomial mutation values used by HEAP and MOEA/D-DE

Parameter	*Values*
Polynomial mutation probability p_m	$1/n$
Polynomial mutation index η_m	20

The parameters for calculating the outranking relation used by both multi-criteria ordinal classification methods are indicated in Table 7.

Table 7. The outranking model parameters in instances with three and eight objectives

Thresholds	*Values to three objectives*			*Values to eight objectives*							
Weights	0.4	0.3	0.3	0.26	0.16	0.11	0.11	0.09	0.09	0.09	0.09
Veto	0.3	0.4	0.4	0.3	0.4	0.4	0.4	0.4	0.4	0.4	0.4
Indifference	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02	0.02
Pre-veto	0.15	0.2	0.2	0.15	0.2	0.2	0.2	0.2	0.2	0.2	0.2

Characterisation of the ROI

The formulation of DTLZ1–DTLZ7 [43] allows knowing whether a generated solution corresponds to the true Pareto front (TPF). The TPFs used as reference fronts for each of the problems were taken from the literature[*]. The 10% of solutions from each TPF was used to characterise the ROI. This process is presented in Algorithm 4. Let us consider the following: for each solution $x \in$ TPF we count the number of solutions that strictly outrank x (Line 2); the set TPF was arranged in ascending order based on the outranking count, which means that the solutions were ordered according to the number of solutions that strictly outrank them. The ROI was characterised by the less-strictly-outranked solutions. We established that ten percent of the solutions in TPF is enough to characterise the ROI.

Algorithm 4. Characterisation of the ROI

Input: Solution set *TPF*
Output: A representation of the ROI
1. **For each** $x \in TPF$ **do**
2.   **Compute** $|S(TPF, x)|$ //Eq. (6)
3. **Arrange** the set *TPF* in ascending order according to $|S(TPF, x)|$
4. *roi* ← select ten percent of the solutions in *TPF* (the first solutions in *TPF*)

[*]https://github.com/jMetal/jMetal/tree/master/jmetal-core/src/main/resources/pareto_fronts
https://github.com/MOEAFramework/MOEAFramework/tree/master/pf

4.1.2 Results in scalable test problems

The performance of Evolutionary Multi-Objective Optimisation (EMO) algorithms is assessed on two aspects: convergence (how close the obtained non-dominated objective vectors are from the true Pareto optimal front), and the distribution of the obtained objective vectors. To assess the quality of Pareto fronts produced by algorithms, a certain number of quality indicators has been proposed and applied by the existing works, e.g., Generational Distance (GD) [44], Inverted Generation Distance [45], Hypervolume [46] and, Epsilon [47]. Most of these indicators involve not only the convergence towards the Pareto front but also the distribution. However, in preference-based MOEAs, the non-dominated solutions are obtained from the ROI, and not from the entire Pareto front as in classical MOEAs. Therefore, algorithms that incorporate preferences are less interested in the distribution and focus more on closeness to the ROI. Therefore, the quality indicators to measure the performance of preference-based MOEAs should be selected or even designed, considering mainly the convergence towards the ROI [48,49]. As far as we know, there is still no quality indicator recognized by the scientific community to evaluate the solutions of the algorithms that incorporate preferences.

Considering the above, in this work we have selected GD as the quality indicator because it measures, without regarding the distribution, the closeness between a portion of an approximated Pareto front and the corresponding portion on the true Pareto front. The fact that the GD as a convergence indicator does not measure the distribution is seen as a limitation by numerous researchers in MOEAs. Nevertheless, this situation is not a real drawback when the interest is to measure the closeness to the ROI. Also, according to [48,50], GD is considered as one of the most commonly used quality indicators, as shown by [49,51]. For these reasons, the GD to the ROI will be used in this work as our performance indicator. A lower distance value represents a better performance.

The aim of our experiments is to discover which approach achieves a set of solutions closest to the characterised ROI. We conducted comparative tests between our hybrid approach (using both multi-criteria classification methods) and MOEA/D-DE. The mean, minimum and maximum GD values are reported, and the best performance is shown in boldface. Additionally, this work used the STAC Web Platform* to carry out non-parametric statistical tests. The Friedman Aligned Ranks test was selected as our ranking test and the Holm's test as the post-hoc method. The significance level was set at 0.05. In the Friedman ranking test, the null hypothesis H_0 was: the means of the results of two or more algorithms are the same. In the Holm's test, the null hypothesis H_0 was**:** the mean of the results of each pair of groups is equal.

Table 8 shows the results of the average GD to the ROI obtained by each algorithm, and in accordance with this distance, the results of the Friedman statistical test for multiple comparison are also shown. As can be seen in Table 8, HEAP-ELECTRE achieved the smallest GD to the ROI in most experiments. Only in DTLZ5, HEAP-THESEUS achieved the smallest GD to the ROI. The results obtained from the Friedman test reject the null hypothesis. Therefore, there are significant differences in the performance of different methods. However, this result does not indicate which is the best algorithm. We have applied the Holm's post-hoc test to Friedman ranks, to determine if there are significant differences between pairs of algorithms and confirm rank ordering.

*http://tec.citius.usc.es/stac/

Table 8. Mean, minimum and maximum GD to the ROI by HEAP-THESEUS, HEAP-ELECTRE and MOEA/D-DE on three-objective DTLZ test problems. The best values are shown in **boldface**

Problem	*Algorithm*	*Mean GD*	*Min GD*	*Max GD*	*Friedman Aligned Ranks test*	
					Rank	*p-value and result*
DTLZ1	HEAP-THESEUS	0.0733580	0.0723256	0.0743692	45.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0649435**	**0.0634274**	**0.0662011**	15.5	
	MOEA/D-DE	0.1715672	0.1684358	0.1896797	75.5	
DTLZ2	HEAP-THESEUS	0.0589555	0.0571426	0.0603732	45.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0479993**	**0.0459080**	**0.0507352**	15.5	
	MOEA/D-DE	0.1055179	0.1049708	0.1058200	75.5	
DTLZ3	HEAP-THESEUS	0.0503874	0.0489506	0.0516676	45.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0366177**	**0.0347426**	**0.0389850**	15.5	
	MOEA/D-DE	0.1487710	0.1473552	0.1505630	75.5	
DTLZ4	HEAP-THESEUS	0.0706279	0.0670725	0.0751057	45.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0608331**	**0.0587930**	**0.0628753**	15.5	
	MOEA/D-DE	0.1311452	0.1265166	0.1374707	75.5	
DTLZ5	HEAP-THESEUS	**0.0762713**	**0.0748984**	**0.0771429**	15.5	0, H_0 is rejected
	HEAP-ELECTRE	0.0897435	0.0862568	0.0935900	45.5	
	MOEA/D-DE	0.6474479	0.6472327	0.6477862	75.5	
DTLZ6	HEAP-THESEUS	0.0912188	0.0895426	0.0930933	44.2	0, H_0 is rejected
	HEAP-ELECTRE	**0.0875286**	**0.0829908**	**0.0927482**	16.7	
	MOEA/D-DE	0.5262535	0.5262254	0.5262941	75.5	
DTLZ7	HEAP-THESEUS	0.0433866	0.0426132	0.0445156	45.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0312654**	**0.0288854**	**0.0338405**	15.5	
	MOEA/D-DE	0.0616323	0.0551616	0.0628087	75.5	

Table 9 exhibits the results of the Holm's test upon comparing pairs of algorithms. The statistical tests resulted in the rejection of the null hypothesis in all cases, that is, there were significant differences in the performance of algorithms, confirming rank order validity of the pair. All these results correspond to problems with three objectives.

PREPRINT

Table 9. Statistical test results of the comparison between pairs of algorithms with Holm's post hoc analysis, on three-objective DTLZ test problems

Problem	*Algorithms*	*Holm test*	
		p-value	*Result*
DTLZ1	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected
DTLZ2	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected
DTLZ3	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected
DTLZ4	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected
DTLZ5	MOEA/D-DE vs HEAP-ELECTRE	0.00002	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected
DTLZ6	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00001	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00005	H_0 is rejected
DTLZ7	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected

In order to display an example of the experimental results, we have plotted in Figs. 5 and 6 the solutions of one run of the problems DTLZ3 and DTLZ7 with three objectives.

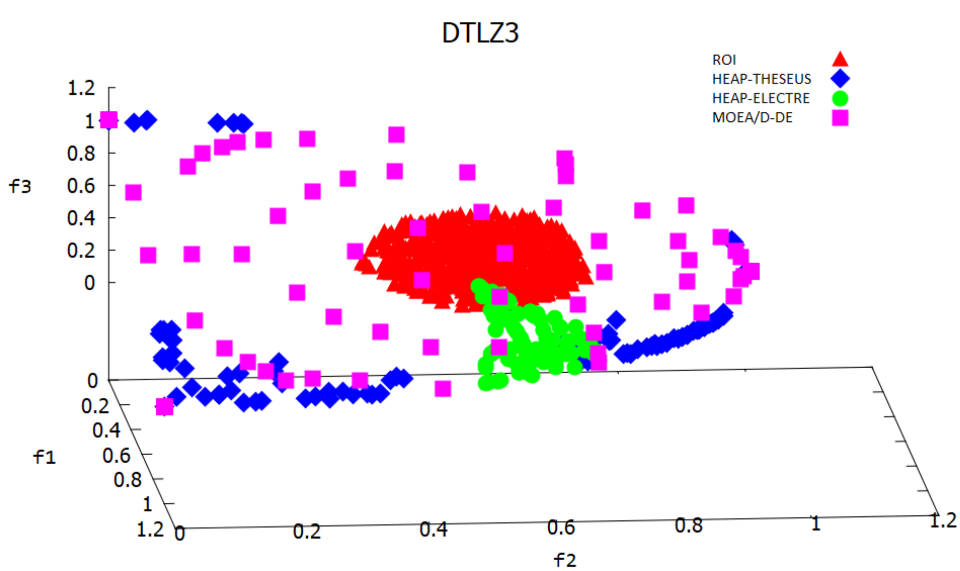

Fig. 5. Results of one run of the DTLZ3 problem

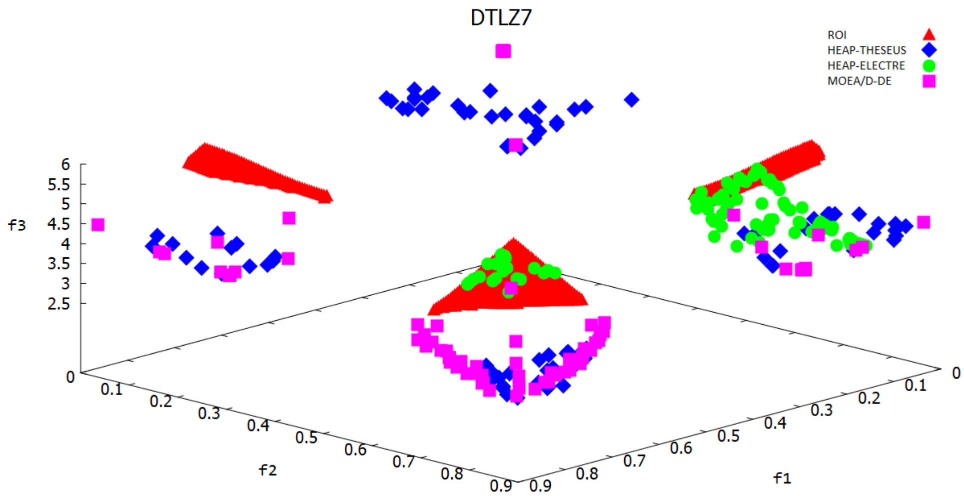

Fig. 6. Results of one run of the DTLZ7 problem

In problems with eight objectives, HEAP-ELECTRE obtained the smallest average GD towards the ROI in DTLZ1, DTLZ2, DTLZ5, DTLZ6, whereas HEAP-THESEUS obtained it in DTLZ3, DTLZ4, DTLZ7. These results were statistically validated by the Friedman Aligned Ranks test and the null hypothesis was rejected in all cases (see Table 10). Therefore, there are significant differences in the performance of different methods. Table 11 shows Holm's post hoc analysis of the comparison between pairs of algorithms. Statistical tests in most cases resulted in the rejection of the null hypothesis, that is, there were significant differences in the performance of the algorithms. Only in DTLZ3 and DTLZ7, H_0 was accepted upon comparing HEAP-THESEUS vs HEAP-ELECTRE and MOEA/D-DE vs HEAP-ELECTRE, respectively, which means that the performance of these algorithms is similar.

Table 10. Mean, minimum and maximum GD to the ROI by HEAP-THESEUS, HEAP-ELECTRE and MOEA/D-DE on eight-objective DTLZ test problems. The best values are shown in **boldface**

Problem	*Algorithm*	*Mean GD*	*Min GD*	*Max GD*	*Friedman Aligned Ranks test*	
					Rank	*p-value and result*
DTLZ1	HEAP-THESEUS	0.0458314	0.0379639	0.0546373	43.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0270424**	**0.0256780**	**0.0290760**	17.4	
	MOEA/D-DE	0.0795901	0.0606147	0.1592223	75.5	
DTLZ2	HEAP-THESEUS	0.0672396	0.0573536	0.0692370	75.1	0, H_0 is rejected
	HEAP-ELECTRE	**0.0497071**	**0.0473928**	**0.0509862**	15.8	
	MOEA/D-DE	0.0565191	0.0493613	0.0633782	45.6	
DTLZ3	HEAP-THESEUS	**0.0588044**	0.0553938	**0.0626393**	33.5	0.00006, H_0 is rejected
	HEAP-ELECTRE	0.0594392	0.0544581	0.0640900	37.7	
	MOEA/D-DE	1.0195812	**0.0528981**	5.8883738	65.2	
DTLZ4	HEAP-THESEUS	**0.0485468**	**0.0442056**	0.0563014	18.6	0, H_0 is rejected
	HEAP-ELECTRE	0.0542395	0.0534894	**0.0550693**	67.1	
	MOEA/D-DE	0.0529739	0.0502216	0.0555352	50.6	
DTLZ5	HEAP-THESEUS	0.1056569	0.1034018	0.1086645	41.4	0, H_0 is rejected
	HEAP-ELECTRE	**0.0918547**	**0.0773312**	**0.1064385**	19.5	
	MOEA/D-DE	1.4558328	1.4258210	1.5390851	75.5	
DTLZ6	HEAP-THESEUS	0.1187496	0.1147829	0.1221445	45.5	0, H_0 is rejected
	HEAP-ELECTRE	**0.0721288**	**0.0655664**	**0.0823243**	15.5	
	MOEA/D-DE	3.3151160	3.2773026	3.4163744	75.5	
DTLZ7	HEAP-THESEUS	**0.0490189**	**0.0400838**	0.0595119	28.6	0.0009, H_0 is rejected
	HEAP-ELECTRE	0.0510092	0.0493711	**0.0524904**	49.6	
	MOEA/D-DE	0.0519194	0.0485054	0.0582035	58.2	

PREPRINT

Table 11. Statistical test results of the comparison between pairs of algorithms with Holm's post hoc analysis, on eight-objective DTLZ test problems

Problem	*Algorithms*	*Holm test*	
		p-value	*Result*
DTLZ1	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00011	H_0 is rejected
DTLZ2	MOEA/D-DE vs HEAP-ELECTRE	0.00002	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0	H_0 is rejected
DTLZ3	MOEA/D-DE vs HEAP-ELECTRE	0.00009	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00001	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.54003	*H_0 is accepted*
DTLZ4	MOEA/D-DE vs HEAP-ELECTRE	0.01444	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0	H_0 is rejected
DTLZ5	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00115	H_0 is rejected
DTLZ6	MOEA/D-DE vs HEAP-ELECTRE	0	H_0 is rejected
	MOEA/D-DE vs HEAP-THESEUS	0.00002	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00002	H_0 is rejected
DTLZ7	MOEA/D-DE vs HEAP-ELECTRE	0.20408	*H_0 is accepted*
	MOEA/D-DE vs HEAP-THESEUS	0.00003	H_0 is rejected
	HEAP-THESEUS vs HEAP-ELECTRE	0.00358	H_0 is rejected

In order to know the robustness of the hybrid approach concerning the increment in the number of objectives, the Wilcoxon statistical tests were performed with a significance level of 0.05. The null hypothesis H_0 was: The medians of the differences between the two group samples are equal. The generational distances obtained in each DTLZ problem were grouped according to the number of objectives (3 and 8), giving rise to two groups which were compared. This was done for each classifier. The results of the statistical test are shown in Table 12. For HEAP-ELECTRE, the analysis indicates that its overall performance in the DTLZ test problems does not vary significantly when the number of objectives increases. For HEAP-THESEUS, the test shows that there are significant differences in their overall performance when the number of objectives increases.

Table 12. The overall performance of the hybrid approach when applying the Wilcoxon statistical test

Algoritmo	*SD* three obj*	*SD* eight obj*	*Statistic*	*p-value*	*Result*
HEAP-ELECTRE	0.0214275	0.01900082	10467	0.4886438	H_0 is accepted
HEAP-THESEUS	0.0152839	0.02769361	8189	0.0010517	H_0 is rejected

**Standard Deviation*

The statistical analysis for each problem is shown in Tables 13 and 14. In Table 13, the statistical test indicates that HEAP-ELECTRE results improve by increasing the number of objectives in the DTLZ1, DTLZ4 and DTLZ6 problems. The results are degraded in the DTLZ2, DTLZ3 and DTLZ7 problems. In the DTLZ5 problem, there are no significant differences.

Table 13. HEAP-ELECTRE statistical results for each DTLZ test problem

Problem	*SD* three obj*	*SD* eight obj*	*Mean GD three obj*	*Mean GD eight obj*	*Statistic*	*p-value*	*Result*
DTLZ1	0.000746671	0.001034030	**0.0649435**	**0.0270424**	0	1.7344E-06	H_0 is rejected
DTLZ2	0.001040563	0.000956612	0.0479993	0.0497071	29	2.8434E-05	H_0 is rejected
DTLZ3	0.000972166	0.003820774	0.0366177	0.0594392	0	1.7344E-06	H_0 is rejected
DTLZ4	0.001067292	0.000425914	**0.0608331**	**0.0542395**	0	1.7344E-06	H_0 is rejected
DTLZ5	0.001764457	0.006998189	0.0897435	0.0918547	154	0.10639417	*H_0 is accepted*
DTLZ6	0.002537881	0.003677811	**0.0875286**	**0.0721288**	0	1.7344E-06	H_0 is rejected
DTLZ7	0.001410511	0.000799243	0.0312654	0.0510092	0	1.7344E-06	H_0 is rejected

Table 14 provides the results for HEAP-THESEUS. The statistical test indicates that the results in problems with eight objectives improve in DTLZ1 and DTLZ4, and degrade in the remaining problems.

Table 14. HEAP-THESEUS statistical results for each DTLZ test problem

Problem	*SD* three obj*	*SD eight obj*	*Mean GD three obj*	*Mean GD eight obj*	*Statistic*	*p-value*	*Result*
DTLZ1	0.000591581	0.004916545	**0.0733580**	**0.0458314**	0	1.7344E-06	H_0 is rejected
DTLZ2	0.000582953	0.004136502	0.0589555	0.0672396	9	4.2856E-06	H_0 is rejected
DTLZ3	0.000702218	0.002215381	0.0503874	0.0588044	0	1.7344E-06	H_0 is rejected
DTLZ4	0.001968730	0.003100972	**0.0706279**	**0.0485468**	0	1.7344E-06	H_0 is rejected
DTLZ5	0.000558477	0.001483283	0.0762713	0.1056569	0	1.7344E-06	H_0 is rejected
DTLZ6	0.000980083	0.001531944	0.0912188	0.1187496	0	1.7344E-06	H_0 is rejected
DTLZ7	0.000454796	0.004694310	0.0433866	0.0490189	16	8.4660E-06	H_0 is rejected

4.2 A multi-criteria project portfolio optimisation problem

One of the main tasks of management in any organization is to evaluate and choose a set of projects competing for financial support to form a project portfolio. The DM is the entity in charge of selecting the portfolio that will be implemented by the corporation. The selected portfolio should satisfy budget constraints. The project portfolio problem is modeled as follows:

$$\underset{x \in R_F}{Max}(z(x))$$

where:

- $x = (x_1, x_2,..., x_p)$ is a portfolio of p projects, each x_i is a binary variable, $x_i = 1$ if the ith project is financed and $x_i = 0$ otherwise;
- $z(x) = (z_1(x), z_2(x), ... , z_N(x))$ is the union of the contribution of each of the projects that compose a portfolio x of N objectives, each $z_j(x) = \sum_{i=1}^{p} x_i f_j(i)$;
- $f(i) = (f_1(i), f_2(i),..., f_N(i))$ is an N-dimensional vector that denotes a project, each $f_j(i)$ is the contribution of project i to the jth objective;
- each project $f(i)$ is characterised by a cost (c_i), classifiable in m activity areas a_i (i=1,…,m) such as health, education, etc., and n geographic regions g_i (i=1,…,n);
- R_F is a feasible region;

s.t.

- a budget constraint defined as $\left(\sum_{i=1}^{p} x_i c_i\right) \leq B$ where B is a total budget;
- an area constraint $LA_k \leq \sum_{i=1}^{p} x_i q_i(k) c_i \leq UA_k$ where LA_k and UA_k are lower and upper limits, respectively, $q_i(k) = 1$ if $a_i = k$ and $q_i(k) = 0$ otherwise;
- a geographic region constraint $LR_k \leq \sum_{i=1}^{p} x_i t_i(k) c_i \leq UR_k$ where LR_k and UR_k are lower and upper limits, respectively, $t_i(k) = 1$ if $g_i = k$ and $t_i(k) = 0$ otherwise.

In this experiment, the comparison was performed between MOEA/D and HEAP and it was carried out by pairs of methods. The different approaches were compared in order to know which was able to get a better representation of the known ROI, that is, non-dominated (ND) and satisfactory solutions.

4.2.1 Parameters settings

We experimented with the project portfolio problem on three and eight objectives, using three instances for each problem. The parameters are the same for HEAP and MOEA/D, where the index of the T closest vectors was set to 10, according to [32]. The parameters of the evolutionary process and the configuration of each problem are shown in Tables 15 and 16, respectively. As mentioned previously, the hybrid algorithm used half of the total number of iterations in the first phase and the other half in the second phase. The parameters used to calculate the outranking relation used by both classification methods are indicated in Table 17.

Table 15. Parameters used in the evolutionary process

Parameter	*Values*
Crossover probability	1
Mutation probability	0.01

Table 16. Configuration of instances

Parameter	*Instance configuration*	
No. objectives	3	8
No. projects	100	100
Weight vectors	105	120
Population size	105	120
Total iterations	500	500

Table 17. The outranking model parameters in instances with three and eight objectives

Thresholds	*Values to three objectives*			*Values to eight objectives*							
Weights	40	11	49	10	13	10	12	7	13	10	7
Veto	102000	30000	1100	120000	90000	150000	100000	168000	120000	200000	156000
Indifference	3750	750	37.5	3750	3000	4500	3000	6000	3750	6000	5250
Pre-veto	54750	15750	587.5	63750	48000	79500	53000	90000	63750	106000	83250

4.2.2 Results in the project portfolio optimisation problem

The solutions obtained by the algorithms are combined in a single set to determine which of them gets a greater number of solutions corresponding to the approximated ROI, that will be called ROI for simplicity in what follows. It is worth mentioning that the solutions of MOEA/D were classified by the multi-criteria sorting methods (see section 2.3), used by the hybrid approach, with the aim of identifying which solutions belong to the ROI. That is, the solutions were sorted by ELECTRE-TRI and THESEUS in comparison with HEAP-ELECTRE and HEAP-THESEUS, respectively.

In instances with three objectives, the first comparison was performed between HEAP-ELECTRE and HEAP-THESEUS. The results showed that HEAP-THESEUS obtained a better representation of the ROI in instances 2 and 3 as it kept a higher percentage of non-dominated and satisfactory solutions than HEAP-ELECTRE. HEAP-THESEUS dominated between 0%–45% of the solutions generated by HEAP-ELECTRE. Meanwhile, the solutions obtained by HEAP-ELECTRE dominated between 2%–3% of the HEAP-THESEUS solutions. This information can be seen in Table 18.

Table 18. Results between HEAP-THESEUS and HEAP-ELECTRE in problems with three objectives

Instance	*Algorithm*	*Average of 30 runs*		
		Solution set size	*Solutions in the known ROI ($T \cup E$)**	*% Solutions in the known ROI*
1	HEAP- THESEUS	108	106	98%
	HEAP- ELECTRE	18	18	**100%**
2	HEAP- THESEUS	39	38	**97%**
	HEAP- ELECTRE	7	5	71%
3	HEAP- THESEUS	106	104	**98%**
	HEAP- ELECTRE	11	6	55%

**T* and *E* are the solution sets generated by HEAP-THESEUS and HEAP-ELECTRE, respectively.

Table 19 shows the comparison between MOEA/D and HEAP-THESEUS in three objectives. It allows us to observe that HEAP-THESEUS preserves more non-dominated solutions than MOEA/D (fifth column). Between 0%–6% of MOEA/D's solutions achieved to characterise the ROI, whereas 99%–100% of the solutions generated by HEAP-THESEUS do represent the ROI (seventh column).

Table 19. Comparative results between HEAP-THESEUS and MOEA/D in problems with three objectives

Instance	*Algorithm*	*Average of 30 runs*				
		Solution set size	*ND solutions*	*% ND solutions*	*Solutions in the known ROI*	*% Solutions in the known ROI*
1	HEAP-THESEUS	108	108	**100%**	108	**100%**
	MOEA/D	156	84	54%	9	6%
2	HEAP-THESEUS	39	39	**100%**	39	**100%**
	MOEA/D	134	85	63%	0	0%
3	HEAP-THESEUS	106	105	**99%**	105	**99%**
	MOEA/D	136	69	51%	6	4%

The comparison between MOEA/D and HEAP-ELECTRE in three objectives is presented in Table 20. It shows that HEAP-ELECTRE is able to maintain a larger number of solutions as non-dominated (fifth column). This happens although the average number of obtained solutions by HEAP-ELECTRE is smaller than that generated by MOEA/D. MOEA/D generated a poor representation of the ROI, whereas HEAP-ELECTRE was able to perform a good representation of this set (seventh column).

Table 20. Comparative results between HEAP-ELECTRE and MOEA/D in problems with three objectives

Instance	*Algorithm*	*Average of 30 runs*				
		Solution set size	*ND solutions*	*% ND solutions*	*Solutions in the known ROI*	*% Solutions in the known ROI*
1	HEAP-ELECTRE	18	18	**100%**	18	**100%**
	MOEA/D	156	124	79%	3	2%
2	HEAP-ELECTRE	7	7	**100%**	7	**100%**
	MOEA/D	134	108	81%	0	0%
3	HEAP-ELECTRE	11	11	**100%**	11	**100%**
	MOEA/D	136	131	96%	4	3%

Afterwards, we used the Wilcoxon's signed ranked test on solutions belonging to the ROI to validate the results. Table 21 shows the Wilcoxon test results in instances with three objectives. The comparison of pairs of algorithms shows that the null hypothesis is rejected in all cases, that is, there are significant differences in the performance of procedures in instances with three objectives.

Table 21. Results of the Wilcoxon test applied to solutions of the ROI in instances with three objectives

Instance	*Algorithm*	*Wilcoxon test (significance level of 0.05)*	
		p-value	*Result*
1	HEAP-THESEUS HEAP-ELECTRE	1.723E-06	H_0 is rejected
2	HEAP-THESEUS HEAP-ELECTRE	1.722E-06	H_0 is rejected
3	HEAP-THESEUS HEAP-ELECTRE	1.696E-06	H_0 is rejected
1	HEAP-THESEUS MOEA/D	1.701E-06	H_0 is rejected
2	HEAP-THESEUS MOEA/D	1.719E-06	H_0 is rejected
3	HEAP-THESEUS MOEA/D	1.688E-06	H_0 is rejected
1	HEAP-ELECTRE MOEA/D	7.631E-06	H_0 is rejected
2	HEAP-ELECTRE MOEA/D	9.05E-07	H_0 is rejected
3	HEAP-ELECTRE MOEA/D	0.0002435	H_0 is rejected

Figs. 7−9 show the graphs corresponding to one run of the project portfolio problems with three objectives.

PREPRINT

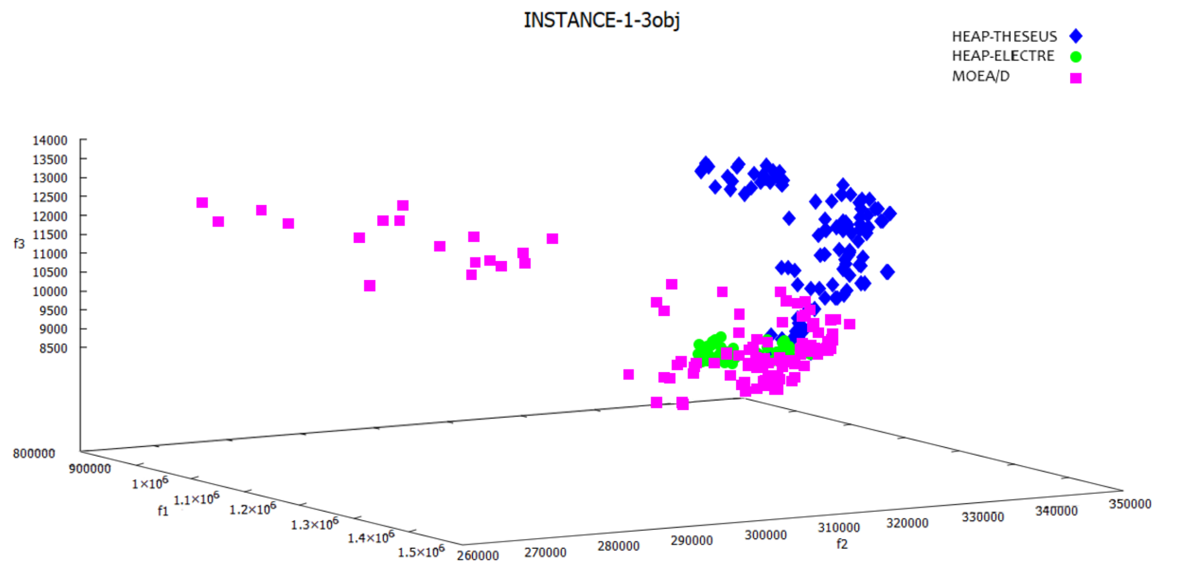

Fig. 7. Results of the instance no. 1 of the project portfolio problem with three objectives

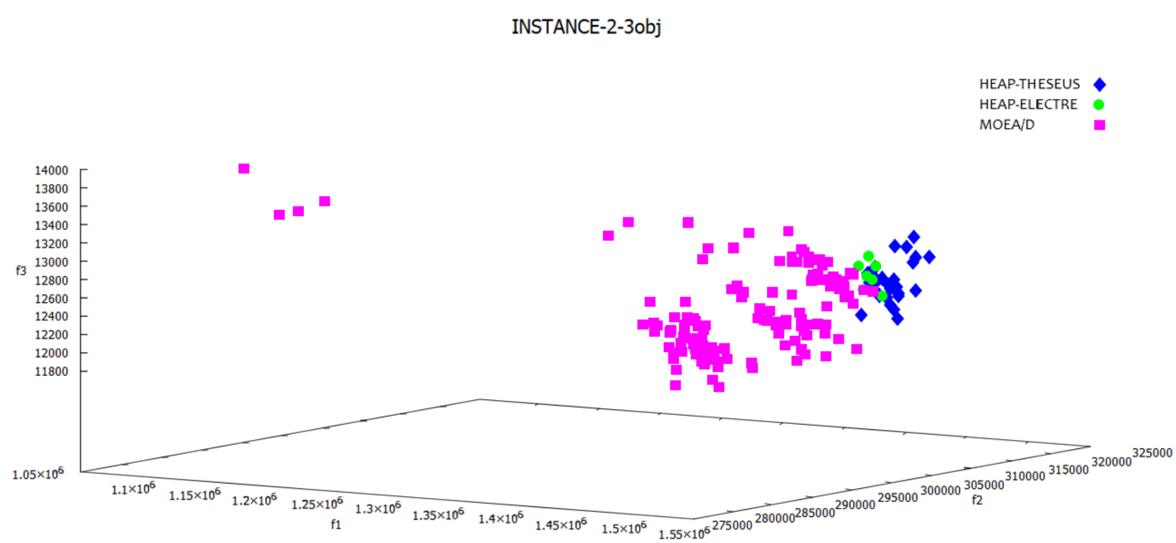

Fig. 8. Results of the instance no. 2 of project portfolio problem with three objectives

PREPRINT

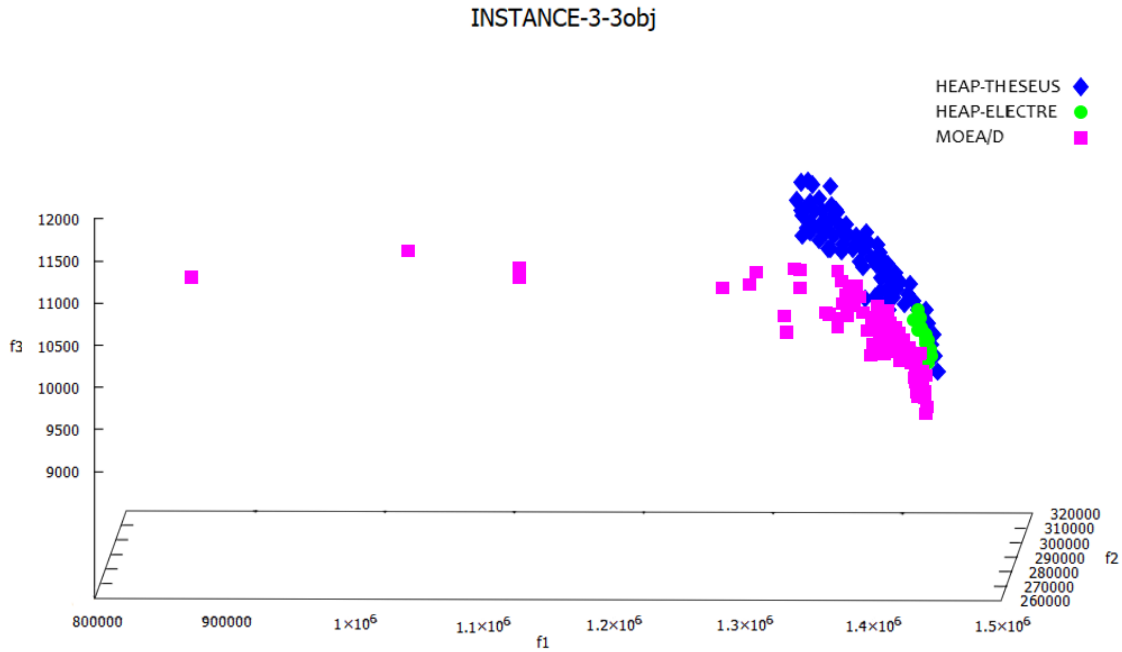

Fig. 9. Results of the instance no. 3 of the project portfolio problem with three objectives

The comparative results between HEAP-ELECTRE and HEAP-THESEUS solving the problem with eight objectives are shown in Table 22. We observe that both algorithms obtain a similar percentage of solutions that belong to the ROI.

Table 22. Results between HEAP-THESEUS and HEAP-ELECTRE in problems with eight objectives

Instance	*Algorithm*	*Average of 30 runs*		
		Solution set size	*Solutions in the known ROI ($T \cup E$)**	*% Solutions in the known ROI*
1	HEAP- THESEUS	126	125	99%
	HEAP- ELECTRE	123	123	**100%**
2	HEAP- THESEUS	124	124	**100%**
	HEAP- ELECTRE	23	22	96%
3	HEAP- THESEUS	126	126	**100%**
	HEAP- ELECTRE	126	126	**100%**

**T* and *E* are the solution sets generated by HEAP-THESEUS and HEAP-ELECTRE, respectively.

Table 23 shows that HEAP-THESEUS always maintains its solutions as non-dominated, whereas MOEA/D conserves between 90%–98% of its solutions as non-dominated (fifth column). Furthermore, all solutions obtained by HEAP-THESEUS characterise the ROI, whereas MOEA/D only obtained between 0%–1% of solutions belonging to this region (seventh column).

PREPRINT

Table 23. Comparative results between HEAP-THESEUS and MOEA/D in problems with eight objectives

Instance	*Algorithm*	*Average of 30 runs*				
		Solution set size	*ND solutions*	*% ND solutions*	*Solutions in the known ROI*	*% Solutions in the known ROI*
1	HEAP-THESEUS	126	126	**100%**	126	**100%**
	MOEA/D	3162	3102	98%	7	0%
2	HEAP-THESEUS	124	124	**100%**	124	**100%**
	MOEA/D	2575	2408	94%	20	1%
3	HEAP-THESEUS	126	126	**100%**	126	**100%**
	MOEA/D	3775	3403	90%	5	0%

The results given in Table 24 show a better performance of HEAP-ELECTRE than MOEA/D, since it maintained a slightly higher percentage of solutions as non-dominated in the problems with eight objectives (fifth column). MOEA/D did not get solutions belonging to the ROI. In contrast, HEAP-ELECTRE was able to perform a good characterisation by getting 100% of solutions belonging to the ROI (seventh column).

Table 24. Comparative results between HEAP-ELECTRE and MOEA/D in problems with eight objectives

Instance	*Algorithm*	*Average of 30 runs*				
		Solution set size	*ND solutions*	*% ND solutions*	*Solutions in the known ROI*	*% Solutions in the known ROI*
1	HEAP-ELECTRE	123	123	**100%**	123	**100%**
	MOEA/D	3162	3113	98%	4	0%
2	HEAP-ELECTRE	23	23	**100%**	23	**100%**
	MOEA/D	2575	2558	99%	4	0%
3	HEAP-ELECTRE	126	126	**100%**	126	**100%**
	MOEA/D	3775	3453	91%	3	0%

Table 25 exhibits the statistical results of the Wilcoxon test for instances with eight objectives. We can see that in all but one case, the null hypothesis is rejected. The case where the null hypothesis is accepted was in the comparison between HEAP-ELECTRE and HEAP-THESEUS on the instance three, which confirms the results shown in Table 22. It means that both algorithms have a similar performance.

Table 25. Results of Wilcoxon test applied to solutions in the ROI in instances with eight objectives

Instance	*Algorithm*	*Wilcoxon test (significance level of 0.05)*	
		p-value	*Result*
1	HEAP-THESEUS HEAP-ELECTRE	0.00073375	H_0 is rejected
2	HEAP-THESEUS HEAP-ELECTRE	1.7224E-06	H_0 is rejected
3	HEAP-THESEUS HEAP-ELECTRE	0.6229019	*H_0 is accepted*
1	HEAP-THESEUS MOEA/D	1.6721E-06	H_0 is rejected
2	HEAP-THESEUS MOEA/D	1.7181E-06	H_0 is rejected
3	HEAP-THESEUS MOEA/D	1.703E-06	H_0 is rejected
1	HEAP-ELECTRE MOEA/D	1.6731E-06	H_0 is rejected
2	HEAP-ELECTRE MOEA/D	2.98E-05	H_0 is rejected
3	HEAP-ELECTRE MOEA/D	1.6742E-06	H_0 is rejected

5 Concluding remarks

This paper presents a hybrid evolutionary algorithm to explore the effectiveness of using multi-criteria ordinal classification methods to incorporate the DM's preferences into the optimisation process to lead the search towards the ROI, that is, the region of the Pareto frontier where solutions that are more in agreement with the DM's preferences are located. The preferences are reflected by the parameters of an outranking relation, by a reference profile and by a reference set of assignment examples. In practice, this information should be elicited (directly or indirectly) by the DM, but with the purpose of evaluating our proposal, in the absence of a real DM, (s)he was simulated by an outranking model.

The proposal fulfills several desirable characteristics of a method of preferences incorporation:

a) an easy interaction between the DM and the solution generator algorithm, minimizing the cognitive effort from the DM when (s)he classifies solutions as "satisfactory" and "unsatisfactory";
b) no requirement of comparability and transitivity of preferences;
c) the outranking model of multi-criteria preferences is compatible with relevant characteristics of real DMs, such as non-transitive and non-compensatory preferences (veto effects);
d) we have several tools to infer the decision model parameters from the assignment examples provided by the DM during the interactive process.

ELECTRE and THESEUS were used as multi-criteria ordinal classification approaches combined with an evolutionary algorithm, giving rise to the HEAP-ELECTRE and HEAP-THESEUS hybrid procedures, respectively. These classification methods are in charge of assigning the new solutions generated during the evolutionary process to one of the ordered classes. To a certain extent, the ordinal classification method replaces the DM during the optimisation process.

Our approach provides as output a set of non-dominated solutions which were assigned to the best class. We carried out experiments with scalable benchmark problems (DTLZ1-DTLZ7) and with a project portfolio optimisation problem. We used instances with three and eight objectives in both types of problems. The results obtained by HEAP-ELECTRE and HEAP-THESEUS approaches were compared with each other as well as against MOEA/D and MOEA/D-DE. All the results were validated by analyzing their statistical significance using non-parametric tests.

Our experiments allow to rise the following conclusions:

1. Our proposal achieves a good convergence to the ROI in the DTLZ1–DTLZ7 problems with three and eight objectives; this can be argued from the lower values of the mean and maximum generational distance to the ROI.
2. In the DTLZ test problems, using ELECTRE TRI as a classification method, the convergence to the ROI is not degraded when the number of objectives increases from three to eight;
3. In the test instances of project portfolio optimization, the proposed method was able to maintain a larger number of non-dominated solutions than MOEA/D.
4. In the same instances mentioned above, MOEA/D identified only a few solutions of the known ROI (non-dominated and satisfactory solutions).
5. The larger generational distances obtained by MOEA/D in DTLZ test problems indicate that more solutions are far from the ROI in comparison to the solutions identified by our approach.

Point 3 suggests that in some problems the search with preferences already incorporated can identify solutions closer to the Pareto Frontier than metaheuristics representative of the state of the art. Points 4 and 5 strongly suggest that incorporating preferences allows a better characterization of the ROI. This is relevant when the DM's preferences are incorporated a posteriori. For example, in three instances shown in Table 19 MOEA/D only found 9, 0 and 6 solutions in the known ROI; in comparison, 108, 39 and 105 were identified by our method. In an a posteriori incorporation of preferences, the DM can hardly identify the best compromise solution since the ROI has not been well-characterized. Something similar can be seen in Tables 20, 23 and 24.

The hybrid approach obtained a better performance than a metaheuristic representative of the state of the art in all the test problems adopted for comparison, regardless of the specific multi-criteria ordinal classification method. The use of ELECTRE-TRI in the hybrid approach achieved a better performance than THESEUS in most of the DTLZ problems. In several DTLZ test problems, HEAP-ELECTRE improves its convergence when the number of objectives increases from three to eight. THESEUS slightly outperforms ELECTRE-TRI in some instances of project portfolio problems.
Let us remark that our results were obtained with a single run of the preference elicitation phase, since in the absence of a real DM, we renounced to the interactive process of updating the concept of what a satisfactory solution is, as well as to update the setting of

preference model parameters. We hope that, through several preference elicitation phases, even better results can be obtained.

Handling of constraints is an important issue that was not explicitly addressed in this paper. It is rather obvious that constraints (both in the decision and objective spaces) can be handled by HEAP through a version of the constrained dominance principle (e.g. Ma and Wang [52]); that is, an unfeasible solution cannot belong to the first non-dominated front. There could be more elaborated approaches, based on a re-definition of the concept of what a satisfactory solution is. This is a subjective preference statement in which some kind of compensation between objective values and fulfilment of constraints may be acceptable. So, unfeasible solutions (although close to the feasible region) with very good values of some objectives, may be classified by the decision maker as satisfactory solutions. This could lead to a more flexible balance between objective values and constraint satisfaction. More research is needed in this direction, and the tests proposed by Ma and Wang [52] and Liu and Wang [53] should be applied.

As another avenue for future research, the proposed hybrid algorithm should be proved by addressing real problems with real DMs.

Acknowledgements

This work was supported by CONACyT grant no. A1-S-11012 (Investigación Científica Básica). The second author acknowledges the support from CONACYT projects 236154 and 269890; the third author acknowledges the financial supports from the Mexican PRODEP UACJ-PTC-423; the fourth author gratefully acknowledges the support from CONACyT grant no. 2016-01-1920 (Investigación en Fronteras de la Ciencia 2016).

PREPRINT

References

[1] C.A. Coello, G. B. Lamont, & Van Veldhuizen, Evolutionary Algorithms for Solving Multi-Objective Problems, Springer US, Boston, MA, 2007. doi:10.1007/978-0-387-36797-2.

[2] K. Al-Somani, T., & Qureshi, Reliability optimization using genetics algorithms (M. Sc. Thesis), King Abdul-Aziz University, 2000.

[3] E. Fernandez, E. Lopez, F. Lopez, C.A. Coello Coello, Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method, Inf. Sci. (Ny). 181 (2011) 44–56. doi:10.1016/j.ins.2010.09.007.

[4] L. Cruz, E. Fernandez, C. Gomez, G. Rivera, F. Perez, Many-objective portfolio optimization of interdependent projects with “a priori” incorporation of decision-maker preferences, Appl. Math. Inf. Sci. 8 (2014) 1517–1531. doi:10.12785/amis/080405.

[5] S. Bechikh, M. Elarbi, L. Ben Said, Many-objective Optimization Using Evolutionary Algorithms: A Survey, in: Recent Adv. Evol. Multi-Objective Optim., Springer, Cham, 2017: pp. 105–137. doi:10.1007/978-3-319-42978-6_4.

[6] H. Ishibuchi, N. Tsukamoto, Y. Nojima, Evolutionary many-objective optimization: A short review, in: 2008 IEEE Congr. Evol. Comput. (IEEE World Congr. Comput. Intell., IEEE, 2008: pp. 2419–2426. doi:10.1109/CEC.2008.4631121.

[7] D. Corne, J. Knowles, Techniques for Highly Multiobjective Optimisation: Some Nondominated Points are Better than Others, in: Proc. 9th Annu. Conf. Genet. Evol. Comput., 2009: pp. 773–780. doi:10.1145/1276958.1277115.

[8] K. Ikeda, H. Kita, S. Kobayashi, Failure of Pareto-based MOEAs: does non-dominated really mean near to optimal?, in: Proc. 2001 Congr. Evol. Comput. (IEEE Cat. No.01TH8546), IEEE, 2001: pp. 957–962. doi:10.1109/CEC.2001.934293.

[9] K. Deb, J. Sundar, Reference point based multi-objective optimization using evolutionary algorithms, in: Proc. 8th Annu. Conf. Genet. Evol. Comput. - GECCO ’06, 2006: pp. 635–642. doi:10.1145/1143997.1144112.

[10] G.A. Miller, The magical number seven, plus or minus two: some limits on our capacity for processing information, Psychol. Rev. 63 (1956) 81.

doi:10.1037/h0043158.
[11] S. Bechikh, Incorporating Decision Maker's Preference Information in Evolutionary Multi-objective Optimization (Doctoral dissertation), University of Tunis, 2013.
[12] K. Deb, Multi-objective Evolutionary Algorithms: Introducing Bias Among Pareto-optimal Solutions, in: Springer, Berlin, Heidelberg, 2003: pp. 263–292. doi:10.1007/978-3-642-18965-4_10.
[13] J. Branke, K. Deb, Integrating User Preferences into Evolutionary Multi-Objective Optimization, in: Jin Y. Knowl. Inc. Evol. Comput., Springer, Berlin, Heidelberg, 2005. doi:10.1007/978-3-540-44511-1_21.
[14] G.W. Greenwood, X.S. Hu, J.G. D'Ambrosio, Fitness Functions for Multiple Objective Optimization Problems: Combining Preferences with Pareto Rankings, FOGA. 96 (1996) 437–455. https://www.semanticscholar.org/paper/Fitness-Functions-for-Multiple-Objective-Problems%3A-Greenwood-Hu/031d5554451c1d25493a226a0e1e9e5952866e87 (accessed June 1, 2019).
[15] K. Deb, A. Sinha, P.J. Korhonen, J. Wallenius, An interactive evolutionary multiobjective optimization method based on progressively approximated value functions, IEEE Trans. Evol. Comput. 14 (2010) 723–739. doi:10.1109/TEVC.2010.2064323.
[16] Y. Jin, B. Sendhoff, Incorporation Of Fuzzy Preferences Into Evolutionary Multiobjective Optimization, in: GECCO, 2002: p. 683.
[17] D. Cvetkovic, I.C. Parmee, Preferences and their application in evolutionary multiobjective optimization, IEEE Trans. Evol. Comput. 6 (2002) 42–57. doi:10.1109/4235.985691.
[18] J. Molina, L. V. Santana, A.G. Hernández-Díaz, C.A. Coello Coello, R. Caballero, g-dominance: Reference point based dominance for multiobjective metaheuristics, Eur. J. Oper. Res. 197 (2009) 685–692. doi:10.1016/j.ejor.2008.07.015.
[19] J. Branke, T. Kaußler, H. Schmeck, Guidance in evolutionary multi-objective optimization, Adv. Eng. Softw. 32 (2001) 499–507. doi:10.1016/S0965-9978(00)00110-1.
[20] T. Wagner, H. Trautmann, Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions, IEEE Trans. Evol. Comput. 14 (2010) 688–701. doi:10.1109/TEVC.2010.2058119.
[21] E. Fernandez, E. Lopez, S. Bernal, C.A. Coello Coello, J. Navarro, Evolutionary multiobjective optimization using an outranking-based dominance generalization, Comput. Oper. Res. 37 (2010) 390–395. doi:10.1016/j.cor.2009.06.004.
[22] E. Oliveira, C.H. Antunes, An evolutionary algorithm based on an outranking relation for sorting problems, in: Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern., 2010: pp. 2732–2739. doi:10.1109/ICSMC.2010.5641799.
[23] K. Miettinen, Nonlinear Multiobjective Optimization, Springer US, 1998.
[24] S. French, Decision theory : an introduction to the mathematics of rationality, Ellis Horwood, 1993. https://books.google.com.mx/books?id=HcxBPwAACAAJ&dq=%22Decision+Theory.+An+introduction+to+the+mathematics+of+rationality%22&hl=es&sa=X&ved=0ahUKEwiX8rPBz8niAhVOIqwKHaE8CKEQ6AEIKTAA (accessed June 1, 2019).
[25] B. Roy, The outranking approach and the foundations of electre methods, Theory Decis. 31 (1991) 49–73. doi:10.1007/BF00134132.
[26] E. Fernandez, E. Lopez, G. Mazcorro, R. Olmedo, C.A. Coello Coello, Application

of the non-outranked sorting genetic algorithm to public project portfolio selection, Inf. Sci. (Ny). 228 (2013) 131–149. doi:10.1016/j.ins.2012.11.018.
[27] M. Doumpos, Y. Marinakis, M. Marinaki, C. Zopounidis, An evolutionary approach to construction of outranking models for multicriteria classification: The case of the ELECTRE TRI method, Eur. J. Oper. Res. 199 (2009) 496–505. doi:10.1016/j.ejor.2008.11.035.
[28] E. Covantes, E. Fernández, J. Navarro, Handling the Multiplicity of Solutions in a Moea Based PDA-THESEUS Framework for Multi-Criteria Sorting, Found. Comput. Decis. Sci. 41 (2016) 213–235. doi:10.1515/fcds-2016-0013.
[29] E. Fernández, J.R. Figueira, J. Navarro, An indirect elicitation method for the parameters of the ELECTRE TRI-nB model using genetic algorithms, Appl. Soft Comput. 77 (2019) 723–733. doi:10.1016/J.ASOC.2019.01.050.
[30] E. Oliveira, C.H. Antunes, Á. Gomes, A comparative study of different approaches using an outranking relation in a multi-objective evolutionary algorithm, Comput. Oper. Res. 40 (2013) 1602–1615. doi:10.1016/j.cor.2011.09.023.
[31] L. Cruz-Reyes, E. Fernandez, P. Sanchez, C.A. Coello Coello, C. Gomez, Incorporation of implicit decision-maker preferences in multi-objective evolutionary optimization using a multi-criteria classification method, Appl. Soft Comput. J. 50 (2017) 48–57. doi:10.1016/j.asoc.2016.10.037.
[32] Q. Zhang, H. Li, MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition, IEEE Trans. Evol. Comput. 11 (2007) 712–731. doi:10.1109/TEVC.2007.892759.
[33] H. Li, Q. Zhang, Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II, IEEE Trans. Evol. Comput. 13 (2009) 284–302.
[34] B. Roy, The Outranking Approach and the Foundations of Electre Methods, in: Readings Mult. Criteria Decis. Aid, Springer Berlin Heidelberg, Berlin, Heidelberg, 1990: pp. 155–183. doi:10.1007/978-3-642-75935-2_8.
[35] B. Roy, R. Słowiński, Handling effects of reinforced preference and counter-veto in credibility of outranking, Eur. J. Oper. Res. 188 (2008) 185–190. doi:10.1016/j.ejor.2007.04.005.
[36] J. P. Brans, B. Mareschal, PROMETHEE methods, in: Mult. Criteria Decis. Anal. State Art Surv. Int. Ser. Oper. Res. Manag. Sci., 2005: pp. 163–186. doi:https://doi.org/10.1007/0-387-23081-5_5.
[37] E. Fernandez, J. Navarro, A new approach to multi-criteria sorting based on fuzzy outranking relations: The THESEUS method, Eur. J. Oper. Res. 213 (2011) 405–413. doi:10.1016/j.ejor.2011.03.036.
[38] M. Doumpos, C. Zopounidis, Multicriteria decision aid classification methods, Kluwer Academic Publishers, 2002.
[39] V. Mousseau, R. Slowinski, P. Zielniewicz, A user-oriented implementation of the ELECTRE-TRI method integrating preference elicitation support, Comput. Oper. Res. 27 (2000) 757–777. doi:10.1016/S0305-0548(99)00117-3.
[40] W. Yu, Aide multicritère à la décision dans le cadre de la problématique du tri : concepts, méthodes et applications, Paris 9, 1992. https://www.theses.fr/1992PA090032 (accessed June 1, 2019).
[41] K. Li, K. Deb, Q. Zhang, S. Kwong, An evolutionary many-objective optimization algorithm based on dominance and decomposition, IEEE Trans. Evol. Comput. 19 (2015) 694–716. doi:10.1109/TEVC.2014.2373386.

[42] M. Elarbi, S. Bechikh, A. Gupta, L. Ben Said, Y.S. Ong, A New Decomposition-Based NSGA-II for Many-Objective Optimization, IEEE Trans. Syst. Man, Cybern. Syst. 48 (2018) 1191–1210. doi:10.1109/TSMC.2017.2654301.

[43] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, Scalable Test Problems for Evolutionary Multiobjective Optimization, in: Evol. Multiobjective Optim., Springer-Verlag, London, 2005: pp. 105–145. doi:10.1007/1-84628-137-7_6.

[44] D.A. Van Veldhuizen, Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, 1999.

[45] C.A.C. Coello, N.C. Cortés, Solving multiobjective optimization problems using an artificial immune system, Genet. Program. Evolvable Mach. 6 (2005) 163–190. doi:10.1007/s10710-005-6164-x.

[46] E. Zitzler, Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications, Swiss Federal Institute of Technology (ETH), 1999.

[47] E. Zitzler, L. Thiele, M. Laumanns, C.M. Fonseca, V. Da Fonseca Grunert, Performance assessment of multiobjective optimizers: An analysis and review, 2002.

[48] E. Filatovas, O. Kurasova, K. Sindhya, Synchronous R-NSGA-II: An Extended Preference-Based Evolutionary Algorithm for Multi-Objective Optimization, Informatica. 26 (2015) 33–50. doi:10.15388/informatica.2015.37.

[49] M. Li, T. Chen, X. Yao, A Critical Review of “A Practical Guide to Select Quality Indicators for Assessing Pareto-Based Search Algorithms in Search-Based Software Engineering”: Essay on Quality Indicator Selection for SBSE, in: 2018 IEEE/ACM 40th Int. Conf. Softw. Eng. New Ideas Emerg. Technol. Results, 2018: pp. 17–20.

[50] S. Wang, S. Ali, T. Yue, Y. Li, M. Liaaen, A practical guide to select quality indicators for assessing pareto-based search algorithms in search-based software engineering, in: Int. Conf. Softw. Eng., 2016: pp. 631–642. doi:10.1145/2884781.2884880.

[51] S.M. Alzahrani, N. Wattanapongsakorn, Comparative Study of Knee-Based Algorithms for Many-Objective Optimization Problems, ECTI Trans. Comput. Inf. Technol. 12 (2018) 7–16.

[52] Z. Ma & Y. Wang, "Evolutionary Constrained Multiobjective Optimization: Test Suite Construction and Performance Comparisons," in IEEE Trans. Evol. Comput., vol. 23, no. 6, pp. 972-986, 2019. doi: 10.1109/TEVC.2019.2896967

[53] Z. Liu & Y. Wang, "Handling Constrained Multiobjective Optimization Problems With Constraints in Both the Decision and Objective Spaces," in IEEE Trans. Evol. Comput., vol. 23, no. 5, pp. 870-884, 2019. doi: 10.1109/TEVC.2019.2894743