



## INFERRING PARAMETERS OF A RELATIONAL SYSTEM OF PREFERENCES FROM ASSIGNMENT EXAMPLES USING AN EVOLUTIONARY ALGORITHM

Eduardo FERNANDEZ<sup>1</sup>, Nelson RANGEL-VALDEZ<sup>2\*</sup>, Laura CRUZ-REYES<sup>2</sup>,  
Claudia GOMEZ-SANTILLAN<sup>2</sup>, Gilberto RIVERA-ZARATE<sup>3</sup>,  
Patricia SANCHEZ-SOLIS<sup>3</sup>

<sup>1</sup>*Faculty of Civil Engineering, Autonomous University of Sinaloa Culiacan, Culiacan, México*

<sup>2</sup>*Postgraduate and Research Division, National Mexican Institute of Technology/Madero Institute  
of Technology Cd. Madero, Cd. Madero, México*

<sup>3</sup>*National Lab of Information Technology, Autonomous University of Juarez City,  
32310 Ciudad Juárez, CHIH, Mexico*

Received 29 May 2018; accepted 13 November 2018

**Abstract.** Most evolutionary multi-objective algorithms perform poorly in many objective problems. They normally do not make selective pressure towards the Region of Interest (RoI), the privileged zone in the Pareto frontier that contains solutions important to a DM. Several works have proved that a priori incorporation of preferences improves convergence towards the RoI. The work of (Fernandez, E. Lopez, F. Lopez, & Coello Coello, 2011) uses a binary fuzzy outranking relational system to map many-objective problems into a tri-objective optimization problem that searches the RoI; however, it requires the elicitation of many preference parameters, a very hard task. The use of an indirect elicitation approach overcomes such situation by allowing the parameter inference from a battery of examples. Even though the relational system of Fernandez et al. (2011) is based on binary relations, it is more convenient to elicit its parameters from assignment examples. In this sense, this paper proposes an evolutionary-based indirect parameter elicitation method that uses preference information embedded in assignment examples, and it offers an analysis of their impact in a priori incorporation of DM's preferences. Results show, through an extensive computer experiment over random test sets, that the method estimates properly the model parameter's values.

**Keywords:** decision making, multi-objective optimization, outranking methods, fuzzy preferences, parameter elicitation, evolutionary algorithms.

**JEL Classification:** C61, D70, G11.

---

\*Corresponding author. E-mail: [nelson.rangel@itcm.edu.mx](mailto:nelson.rangel@itcm.edu.mx)

## Introduction

A consequence of the multi-objective conflicting nature of optimization problems is the difficulty to reach the ideal solution (cf, Deb, 2001; Coello, Lamont, & Van Veldhuizen, 2007; Zhu & Luo, 2016). Instead, the solution of a Multi-objective Optimization Problem (MOP) can be seen as the search for the best compromise, i.e. the solution that best matches the decision maker's (DM's) preferences. However, in order to find such solution, the DM should provide additional information related to its particular system of preferences (Fernandez et al., 2011).

Multi-Objective Evolutionary Algorithms (MOEAs) are robust algorithms used to approximate the Pareto frontier in situations where mathematical programming has troubles dealing with the characteristics of objective functions and constraints (cf. Coello, 1999). These strategies provide set of solutions that in general do not dominate each other. Alternatively, for a better approximation of the best compromise, an evolutionary approach can incorporate information about the DM's preferences. According to (Branke, Salvatore, Greco, Slowiński, & Zielniewicz, 2016), the incorporation of preferences of a DM in the search process can be motivated by the sampling of the Pareto frontier, the reduction of the DM's cognitive effort to handle only the Region of Interest (RoI), and because the DM's preference information reinforces the necessary selective pressure. Let us point out that the incorporation of preference can be seen as a method that offers support to the limited capacity of the human mind to handle several conflicting objectives at the same time (Miller, 1956). So, the incorporation of DM preferences can be a very useful tool in solving many-objective problems.

The research works presented by Branke (2015) and Bechikh (2013) show a survey about approaches that incorporate preferences in MOEAs. Some examples are value reservation utility functions (Phelps & Koksalan, 2003; Deb, Sinha, Korhonen, & Wallenius, 2010; Branke, Greco, Slowiński, & Zielniewicz, 2015), reservation points (Deb, 2007), reference points (Molina, Santana-Quintero, Hernández-Díaz, Coello Coello, & Caballero, 2009), desirability thresholds (Wagner & Trautmann, 2010), decision rules (Greco, Slowinski, Figueira, & Mousseau, 2010), and outranking relations (e.g. Fernandez et al., 2011). The use of outranking approaches is recommended because they can handle imprecise, ordinal and qualitative information, as well as phenomena like intransitivity, incomparability and veto situations.

The work of (Fernandez et al., 2011) proposes the use of fuzzy outranking relations to incorporate preferences in MOEAs. This work defines a surrogate model that reduces a many-objective optimization problem to one with three objectives to find the best compromise; it has been successfully employed in the solution of portfolio optimization problems in (Fernandez, Lopez, Mazcorro, Olmedo, & Coello, 2013; Fernandez, Gomez, Rivera, & Cruz-Reyes, 2015; Cruz-Reyes, Fernandez, Sanchez, Coello Coello, & Gomez, 2017). A disadvantage of the approach from (Fernandez et al., 2011) is that it requires the elicitation of many preference parameters such as weights, thresholds, and others. Information about the model's parameters can be obtained either directly or indirectly. In the direct-eliciting method, the DM, perhaps in collaboration with a decision-analyst, is responsible for making a direct setting of the preference model's parameters values. This task is certainly difficult for a DM because the meaning that (s)he personally assigns to such parameters is barely clear (Fernandez, Navarro, & Mazcorro, 2012). The direct elicitation has been the subject of strong

criticism in the literature. According to Bouyssou et al. (2006), the only valid preference information is that arising from pair-wise comparisons of actions, or evaluation of actions. On the other hand, Preference Disaggregation Analysis (PDA) is an ad-hoc regression-like method that properly estimates the parameters' values from a battery of examples (Doumpos, Marinakis, Marimaki, & Zopounidis, 2009). These indirect elicitation methods require less time and cognitive effort from the DM (Greco, Mousseau, & Slowinski, 2008).

In order to infer all the parameters required by a preference model, the PDA methods solve a nonlinear problem that commonly involves non-convex constraints (see Fernandez et al., 2012). As shown in (Mousseau & Slowinski, 1998; Dias & Mousseau, 2006), this type of problem is very complex, and it becomes impractical working large data sets (Doumpos et al., 2009). The most viable alternative consists in the use of metaheuristics, such as evolutionary approaches, which can find with ease, and comparatively less time, approximated solutions convenient for real purposes, cf. (Kahraman, Engin, Kaya, & Kerim Yilmaz, 2012; Xiong, Molina, Leon Ortiz, & Herrera, 2015).

In the frame of outranking methods, indirect parameter elicitation approaches are used in multi-criteria ordinal classification problems addressed by ELECTRE TRI (e.g. Mousseau and Slowinski, 1998; Mousseau, Figueira, & Naux, 2001; Doumpos et al., 2009). In these papers, the reference preference information is given through assignment examples. Since the approach by Fernandez et al. (2011) is based on several binary preference relations it would be natural the use of pair-wise comparisons to form a battery of examples as in (Fernandez et al., 2012). However, a PDA strategy can be benefited from a reference set formed by assignment examples. Medium size sets of assignment examples can contain more preference information from the DM than many pairwise comparisons of actions; often the DM feels more comfortable assigning conveniently chosen actions than expressing preferences (strict, weak, indifference, incomparability, K-preference, non-preference) on pairs of actions. Additionally, an important data source to create the reference set comes from past decisions made by the DM or accepted by her/him; usually, these past decisions are given as assignment examples (e.g. students with high, medium or low performance).

Based on the above, this paper proposes a PDA evolutionary approach that uses assignment examples to infer the preference parameters of the model by Fernandez et al. (2011). The inferred model can be used to solve many-objective problems as in (Fernandez et al., 2013, 2015). This paper proposes a method to generate random instances for the problem studied along with a performance model for the proposed PDA approach. This work complements its research by analysing the impact of the suggestions of a DM about the definition of ranges of values for the parameters over the quality of the obtained configuration from the parameter elicitation. The method of solution is analysed through different points of view (based on two metaheuristics, and different settings for the parameter search space). The research shows experimental evidence that the PDA method estimates properly the parameter values for the preference model considered, according to the reference set and by reducing the inconsistencies to zero. Also, it shows that the capacity of prediction on new decisions using the parameters identified results in a low level of inconsistencies. Finally, it also demonstrates the importance of involving the DM in the initial process of identifying proper bounds for the parameters.

This work is related to the popular outranking approach for multi-criteria decision making (e.g. Roy, 1991), which is an option to value or utility function-based methods, e.g. the work of (Gastelum Chavira, Leyva Lopez, Solano Noriega, Ahumada Valenzuela, & Alvarez Carrillo, 2017) uses it to model DM's preferences in credit ranking. The outranking approach is used to address problems in which only partial compensation among criteria is allowed, and non-transitive preferences, veto situations, incomparability, imprecise criterion scores, as well as ordinal and qualitative scales could be a real concern. In comparison to the rational paradigm in which the value function methods are based, the outranking approach is more flexible and capable to model ill-shaped preferences of real decision makers and consumers. In this sense, new methods able to make robust and cognitive less-demanding elicitation of preference model parameters are welcome. In addition, the outranking models are used to compare assets related with the economy of an enterprise, to model preferences of consumer, or to support decision making processes. Moreover, our approach is necessary in optimization strategies like the one in (Fernandez et al., 2011) because they require previous elicitation of parameters to compare alternatives based on the DM's preferences.

The remaining content of this article is organized as follows. Section 1 describes the optimization method proposed for the estimation of parameter values; the associated surrogate model is defined from a battery of examples formed by actions sorted in groups that are defined according to the DM preferences. Section 2 shows the method followed to construct random instances for the reference set organized in groups. Sections 3 and 4 analyse the performance of the approach through an extensive experimental design. Finally, the Section Conclusions shows the final remarks derived from the research.

## **1. An optimization approach for inferring the model's parameter values**

This section details the method followed to infer parameter values. The content is organized into four subsections. The first one defines the reference set utilized as input for the inference process. The second subsection briefly describes the relational system of preferences whose parameters are going to be estimated. In the third subsection, the optimization problem to inferring the model's parameters is formulated; the last subsection details the method followed to solve such problem.

### **1.1. The reference set**

The reference set, denoted as  $T_{n,k}$ , is a set of actions assigned to ordered classes according to the DM's preferences. Without loss of generality, this work considers that the classes are ordered from the worst to the best. The construction of  $T_{n,k}$  is described in Section 2.

### **1.2. The preference model**

The works of (Bechikh, 2013) and (Branke, 2015) are surveys about the use of DM preferences in multi-objective approaches to approximate the RoI. This task influence directly in the reduction of the cognitive effort required from the DM in the search process for a solution to a given optimization problem (cf. Branke et al., 2016). This not only offers a better

opportunity of sampling the Pareto frontier (cf. Bastiani, Cruz-Reyes, Fernandez, & Gomez, 2015) but also aids in the achievement of better solutions.

The ELECTRE methods (cf. Roy, 1996) are among the most widely used preference models in multi-criteria decision-making; their relevance arises from the fact that up to now they are studied in trending topics such as prospective multiple attribute decision making (cf. Zolfani, Maknoon, & Zavadskas, 2016) or its most recent application in prioritized multicriteria decision making (cf. Xiaohan, Suojan, Xianglin, & Xiuli, 2018). The ELECTRE methods have been successfully integrated with fuzzy outranking relational systems of preferences to develop metaheuristics that reinforce the selective pressure toward the RoI in many-objective optimization problems. Indeed, the strategy proposed by Fernandez et al. in 2011, and such relational system of preferences is the object of study.

The fuzzy outranking relational system of (Fernandez et al., 2011) contains six binary relations based on the credibility index of outranking  $\sigma(\eta, x, y)$ , where set  $\eta = \{w, q, p, u, v\}$  includes parameters such as weights  $w_i$ , indifference thresholds  $q_i$ , preference thresholds  $p_i$ , pre-veto thresholds  $u_i$  and veto thresholds  $v_i$ . The credibility index  $\sigma(\eta, x, y)$  is defined over each pair of actions  $(x, y)$  and it quantifies, in the range  $[0, 1]$ , the degree of truth of the predicate “the DM considers that option  $x$  is at least as good as  $y$ ”. The  $\sigma(\eta, x, y)$  can be calculated as in ELECTRE III (Roy, 1996) with the simplification introduced by (Mousseau & Dias, 2004), see Appendix for a brief description of its calculation. The use of  $\sigma(\eta, x, y)$  and the credibility, asymmetry, and symmetry thresholds (denoted  $\lambda, \beta$ , and  $\varepsilon$  respectively) results in relations of indifference, strict preference, weak preference, incomparability,  $K$ -preference, and Non-preference. The formal definition of the binary relations is briefly described in Table 1.

Let us point out that the dominance concept among two alternatives  $x$  and  $y$  is defined as follows (based on the definition presented in (Meghwani & Thakur, 2017)). Let  $x, y$  be two alternatives, each is characterized by a vector of  $n$  objectives, and, without loss of generality, let us consider that those objectives represent minimization functions. The vector  $x$  is said to *dominate*  $y$  (in short:  $x < y$ ), if  $x_i \leq y_i$  for every objective  $i \in \{1, \dots, n\}$ , and  $x_i < y_i$  for at least one of them. Whenever a vector  $x$  dominates a vector  $y$ , then  $x$  can be considered a better solution.

To characterize best compromise solutions (i.e. solutions based on preferences of a specific DM), the work of (Fernandez et al., 2011) uses  $\sigma(\eta, x, y)$  and the relations in Table 1 to define the tri-objective optimization model shown in Eq. (1).

$$\min_{x \in O} \left( \text{card}(N_S), \text{card}(W_{NS}), \text{card}(F_{NS}) \right). \tag{1}$$

The optimization problem in Eq. (1) relies on three different sets to define its objectives. For each feasible alternative  $x \in O$ , a set of strictly outranking solution  $(S_o)_x$  is defined as  $(S_o)_x = \{y \in O \mid yP(\lambda, \beta)x\}$ , i.e. the set  $(S_o)_x$  contains the best compromises  $x$  for which there is no other alternative that has a relation of type  $yPx$ . Then, the set  $(S_o)_x$  defines the  $(\lambda, \beta, \varepsilon)$ -non-strictly-outranked frontier  $N_S = \{x \in O \mid \text{card}(S_o)_x = \emptyset\}$ , where *card* stands for cardinality; the solutions in this frontier represents the subset of the Pareto frontier mostly in agreement with the DM’s preferences.

Table 1. Binary relations of the fuzzy outranking relational system defined in (Fernandez et al., 2011)

Relation	Notation	Formal Definition
$(\lambda, \beta)$ -strictly preferred	$xP(\eta, \lambda, \beta)y$	$x$ dominates $y$ ; <b>OR</b> $\sigma(\eta, x, y) \geq \lambda \wedge \sigma(\eta, y, x) < 0.5$ ; <b>OR</b> $\sigma(\eta, x, y) \geq \lambda \wedge (0.5 \leq \sigma(\eta, y, x) < \lambda) \wedge (\sigma(\eta, x, y) - \sigma(\eta, y, x)) \geq \beta$
$(\lambda, \varepsilon)$ -indifferent	$xI(\eta, \lambda, \varepsilon)y$	$\sigma(\eta, x, y) \geq \lambda \wedge \sigma(\eta, y, x) \geq \lambda \wedge  \sigma(\eta, x, y) - \sigma(\eta, y, x)  < \varepsilon$ .
$(\lambda, \beta, \varepsilon)$ -weakly preferred	$xQ(\eta, \lambda, \beta, \varepsilon)y$	$\sigma(\eta, x, y) \geq \lambda \wedge \sigma(\eta, x, y) > \sigma(y, x)$ ; <b>AND</b> <i>not.</i> $xP(\eta, \lambda, \beta)y$ ; <b>AND</b> <i>not.</i> $xI(\eta, \lambda, \varepsilon)y$ .
$(\lambda, \varepsilon)$ -K-preferred	$xK(\eta, \lambda, \varepsilon)y$	$0.5 \leq \sigma(\eta, x, y) < \lambda$ ; <b>AND</b> $\sigma(\eta, y, x) < 0.5$ ; <b>AND</b> $(\sigma(\eta, x, y) - \sigma(\eta, y, x)) > \varepsilon$ .
incomparability	$xR(h)y$	$\sigma(\eta, x, y) < 0.5 \wedge \sigma(\eta, y, x) < 0.5$ .
$(\lambda, \beta, \varepsilon)$ -non-preference	$x\sim(\eta, \lambda, \beta, \varepsilon)y$	<i>not.</i> $xP(\eta, \lambda, \beta)y \wedge$ <i>not.</i> $yP(\eta, \lambda, \beta)x$ ; <b>AND</b> <i>not.</i> $xQ(\eta, \lambda, \beta, \varepsilon)y \wedge$ <i>not.</i> $yQ(\eta, \lambda, \beta, \varepsilon)x$ ; <b>AND</b> <i>not.</i> $xK(\eta, \lambda, \varepsilon)y \wedge$ <i>not.</i> $yK(\eta, \lambda, \varepsilon)x$ ; <b>AND</b> <i>not.</i> $xI(\eta, \lambda, \varepsilon)y$ ; <b>AND</b> <i>not.</i> $xR(\eta)y$ .

The second objective in Eq. (1) is based on the  $(\lambda, \beta, \varepsilon)$ -non-weakly-outranked frontier ( $W_{NS}$ ). This frontier is formed by solutions  $x$  that does not have alternatives that weakly outranked it, i.e.  $W_{NS} = \{x \in O \mid card(W_{NS})_x = \emptyset\}$ , where  $(W_{NS})_x = \{y \in N_S \mid yQ(\lambda, \beta, \varepsilon)x \text{ or } yK(\lambda, \beta)x\}$ . Finally, the third objective uses the outranking net flow score  $F_n(x)$  of an alternative  $x$ , a very popular measure to rank a set of alternatives in which a fuzzy preference relation is defined. The cardinality of  $(F_{NS}) = \{y \in N_S \mid F_n(y) > F_n(x)\}$ , where  $F_n(a) = \sum_{c \in NS - \{a\}} [\sigma(a, c) - \sigma(c, a)]$  defines such third objective.

The problem in Eq. (1) reduces the original dimensionality of any many-objective optimization problem to only three objectives by exploiting the incorporation of preferences in the search process. This problem requires the elicitation of the parameters of the associated preference model, a task that is solved by the proposed approach presented in the following section.

### 1.3. The inference approach

An inference approach for the previous relational system would be a strategy that could properly estimate the values of the parameters  $(\eta, \lambda, \beta, \varepsilon)$ , i.e. that identifies the configurations of the parameters which minimize the inconsistencies among the preference relations established by the model, and those given by the DM, through a reference set  $T$ . This problem can be formally defined as follows. Let  $P_{fr}$  be the set of feasible parameter vectors  $(\eta, \lambda, \beta, \varepsilon)$ ,  $T$  a reference set formed by a finite set of actions  $\{x_1, x_2, \dots, x_k\}$ ,  $C(x): x \rightarrow C_i$  a function that defines for an action  $x_i \in T$  the class  $C_i$  in which is assigned to, and  $A = \{P, I, Q, K, \sim, R\}$  the set of preference relations; the classes (also called categories in the related literature) are ordered from the worst to the best, according to the DM's preferences in each particular problem. The

ideal solution of the parameter elicitation problem would be  $(\eta_0, \lambda_0, \beta_0, \varepsilon_0) \in P_{fr}$  such that the following equivalences are satisfied for all  $(x, y) \in T \times T$ :

$$E_1: C(x) - C(y) > 1 \Rightarrow xPb \vee xRb; \tag{2.a}$$

$$E_2: C(x) - C(y) = 1 \Rightarrow xIy \vee xQy \vee xPy \vee xKy \vee xRy; \tag{2.b}$$

$$E_3: xPy \Rightarrow C(x) \geq C(y); \tag{2.c}$$

$$E_4: xQy \Rightarrow C(x) = C(y) \text{ or } C(x) - C(y) = 1; \tag{2.d}$$

$$E_5: xIy \Rightarrow |C(x) - C(y)| \leq 1; \tag{2.e}$$

$$E_6: xKy \Rightarrow C(x) = C(y) \text{ or } C(x) - C(y) = 1. \tag{2.f}$$

where  $P, Q, K, I$  and  $R$  are defined as in Table 1, but now depending on  $(\eta, \lambda, \beta, \varepsilon)$ . As a consequence, the truth values of the above implications are functions of the parameters to be inferred. The two first implications, i.e.  $E_1$  and  $E_2$ , ensure that the assignments in the reference set  $T$  agrees with the established preference relation; for example, an action that lies in a class that is non-adjacent to another must establish a strict preference or incomparability relation. The remaining implications are related with a proper order of the classes, in concordance with the relations established by the preference model and its parameter values.

The best parameter setting should be the closest solution to the ideal one in the sense of a certain acceptable metric. Following the works of (Fernandez et al., 2011, 2012) we propose a metric based on inconsistencies. The above implications can identify inconsistencies among the preferences relations coming from the inferred parameters and the DM's preferences contained in the reference set  $T$ . Shortly, an inconsistency arises when an implication in Eqs (2) is *false*. For example, given  $(x, y) \in T$  and  $(\eta, \lambda, \beta, \varepsilon) \in P_{fr}$  we call inconsistency with (1.a) the fact that  $C(x) - C(y) > 1$  but the preference relation is  $xQ(\eta, \lambda, \beta)y$ . Let  $N_1, N_2, \dots, N_6$  be the number of inconsistencies that appear in  $T$  derived from Eqs (2.a), (2.b), (2.c), (2.d), (2.e) and (2.f) respectively.

Since the method by Fernandez et al. in 2011 gives priority to find the non-strictly out-ranked set,  $N_1$  and  $N_3$  are far the most important measures. So, the best values for  $(\eta, \lambda, \beta, \varepsilon)$  can be obtained by solving the optimization problem:

$$\begin{aligned} & \min (f_1, f_2) \\ & \text{s.t. } (\eta, \lambda, \beta, \varepsilon) \in P_{fr} \end{aligned} \tag{3}$$

where  $P_{fr}$  is the feasible region of the parameter values,  $f_1 = N_1 + N_3$ , and  $f_2 = N_2 + N_4 + N_5 + N_6$ , with pre-emptive priority favouring  $f_1$  over  $f_2$ .

Note that the preference information derived from actions organized in classes cannot directly be transformed into relations between pairs of actions. For example, the assignment of two actions to the same class does not imply indifference; within the same class we can have weak preference, K-preference, even strict preference; besides, the assignment of two actions to adjacent classes does not imply strict preference favouring the action in the higher class; the actions could be close to the boundary between the classes and fulfil an indifference (or incomparability) condition. Hence, this work is proposed given that it is not possible to assume preference relations directly from a reference set with actions organized in an ordered set of classes, and exploiting that possibility opens the opportunity to increase preference information favouring the development of more robust parameter elicitation strategies.

#### 1.4. Optimization approach based on a PDA method

The proposed indirect elicitation approach solves Problem (3) using a single-objective optimization Genetic Algorithm (GA). The approach uses the pre-emptive priority established for the problem to solve it in two stages. During the first stage, it searches for the best value  $Bound_{f_1}$  that it could find for the first objective  $f_1$ ; after that, in the second stage, it introduces  $Bound_{f_1}$  as a constraint that must be satisfied when minimizing the value of second objective  $f_2$ . Finally, the approach reports the best sets of parameters  $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$  found at the end of the second stage. The general approach can be seen depicted in Figure 1, and the elements of the GA are described in the remaining of this section.

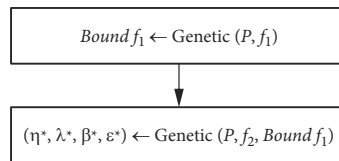


Figure 1. The strategy followed to solve Problem (3) using a single-objective algorithm

**Genetic Representation of the Solution.** A solution in the GA population is represented by a vector  $\rho$  of real numbers. This vector represents the parameter vector  $(\eta, \lambda, \beta, \varepsilon)$  to be estimated to solve Problem (3). Given that  $\eta$  involves the set of parameters  $(w, q, p, u, v)$  per criterion, the size of  $\rho$  is  $5n + 3$ , where  $n$  is the number of criteria.

**Fitness Function.** The objectives  $f_1$  and  $f_2$  of Problem (3) are used to assess the quality of a solution vector  $\rho$ ; they are the fitness functions used by the GA during the first and second stages, respectively (see Figure 1).

**Initialization Function.** The GA initializes its population with 100 individuals randomly generated. Each individual is defined by the vector  $\rho$  previously described. The GA requires the definition of lower and upper bounds for each criterion on  $\rho$ , then it initializes each of them with a value randomly chosen from the interval formed by the bounds.

**Selection.** The binary tournament selection is the chosen method of selecting parents in the GA. This tournament method is used each time that a parent must be selected among the actual population. The method randomly chooses two different individuals as possible parents, then, it selects the one with the best fitness, accordingly to the stage. Ties are broken arbitrarily.

**Genetic Operators.** In order to generate the next population of individuals, or solutions, each pair of selected parents is combined through crossover and/or mutation operators. First, the Simulated Binary Crossover (SBXCrossover) operator simulates the operation of a single-point binary crossover directly on real variables under a probability distribution, cf. (Deb & Agrawal, 1995). The resulting offspring from the SBXCrossover are two children that will be the object of the Polynomial Mutation process, which is used to produce a slight variation on the children, cf. (Deb & Tiwari, 2008).

**Stop Criterion.** Finally, the main loop of the GA is repeated until a maximum number of evaluations on the fitness function is reached. Let us note that one evaluation is accumulated each time that a new child is generated. The number of evaluations is user defined, and in this research, it was set to 100000.



Figure 2 shows the complete pseudocode of the GA. The algorithm generates  $Population_{new}$  each iteration, it is the set of new individuals created by the selection, crossover and mutation operators. In Line 16 the original set of solutions, called  $Population$ , and  $Population_{new}$  are combined and ranked to select the next generation of individuals. The ranking is made through the specified fitness function Objective, and the **size** of the set and the maximum number of evaluation **MaxEvaluations** are left to be defined by the user.

## 2. Instance generator

The PDA strategy proposed in Section 1.4 (see Figure 1) uses  $Problem_{instance}$  as input, i.e. an instance of Problem (3) which is defined as  $\Lambda = (DM, T_{n,k})$  where  $DM$  is the decision maker who is wanted to be modelled, and  $T_{n,k}$  is the training set provided by him/her and that contains  $n$  actions organized in  $k$  classes reflecting his/her preferences.

```

GeneticAlgorithm( $Problem_{instance}$ , Objective)
1:    $Population \leftarrow$  GenerateRandomIndividuals( $Problem_{instance}$ , size)
2:    $DM \leftarrow$  GetDM( $Problem_{instance}$ )
3:    $T \leftarrow$  GetReferenceSet( $Problem_{instance}$ )
4:    $eval \leftarrow 0$ 
5:   repeat
6:      $Population_{new} \leftarrow \emptyset$ 
7:     for  $i = 0$  to size do
8:        $parent_1 \leftarrow$  BinaryTournament( $Population$ , Objective,  $DM$ ,  $T$ )
9:        $parent_2 \leftarrow$  BinaryTournament( $Population$ , Objective,  $DM$ ,  $T$ )
10:       $\{child_1, child_2\} \leftarrow$  SBXCrossover( $parent_1$ ,  $parent_2$ )
11:       $child_1 \leftarrow$  PolynomialMutation( $child_1$ )
12:       $child_2 \leftarrow$  PolynomialMutation( $child_2$ )
13:       $Population_{new} \leftarrow Population_{new} \cup \{child_1, child_2\}$ 
14:       $eval \leftarrow eval + 2$ 
15:    end for
16:     $Population \leftarrow$  RankingByFitness( $Population$ ,  $Population_{new}$ , Objective,  $DM$ ,  $T$ , size)
17:  until  $eval = \text{MaxEvaluations}$ 

```

Figure 2. Genetic Algorithm pseudocode

According to the revised scientific literature, there are no DM's benchmarks who could support an extensive study of the performance of a PDA approach. To address this situation, this work proposes the definition of a mechanism to simulate DMs, and with it the construction of reference sets. Both topics are discussed in the following subsections. Note that this mechanism can be used as reference points for comparison in future researches.

### 2.1. Simulation of a decision maker

A simulated DM, called  $DM_{sim}$ , can be defined in general, for the preference model under study, by specifying values for the associated parameters. In this sense, different sets of model parameter values would represent different DMs, and therefore, those parameter values can

be taken from previous works that provide elicitations for the same preference model, or they can be random values.

The preference model studied on this work is characterized by the criteria parameter set  $\eta = \{w, q, p, u, v\}$ , and the thresholds  $(\lambda, \beta, \varepsilon)$ . Hence, simulated DMs are defined by giving specific values to those parameters. Particularly, one  $DM_{sim}$  is denoted by a vector  $\tau_0 = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$  with specific values on its parameters.

In this work, the simulated DMs are generated artificially. For this purpose, one  $DM_{sim}$  is created by randomly defining the vector  $\tau_0$ . Firstly, for the vector  $\eta_0$ , each parameter value of each criterion is randomly chosen from a predefined interval, verifying the satisfaction of the constraints established on them by (Fernandez, Navarro, & Bernal, 2009; Fernandez et al., 2011). After that, the values of  $\eta_0$  are paired with one of the following tuples  $(\lambda_0, \beta_0, \varepsilon_0) = \{(0.51, 0.15, 0.07), (0.67, 0.15, 0.07), (0.70, 0.20, 0.10), (0.75, 0.20, 0.10)\}$ . The simulated DM generated using this particular method is denoted by  $DM_{sim-parameters}$ , and its specific values  $\tau_0 = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$  are considered as the “true” preference parameter values that should be approximated by the inference approach.

## 2.2. Generation of reference sets $T$ from a $DM_{sim}$

The present study considers two type of sets  $T$ . The training or reference set, denoted by  $T_{n,k}$ , that forms part of the instance  $\Lambda = (DM_{sim}, T_{n,k})$  of Problem (3) and that is used by the proposed PDA approach; and, the test set, called  $T_n$ , that forms part of the test case  $\Gamma = (DM_{sim}, T_n)$  which is used in the performance evaluation of the elicitation process. Let us observe that for a proper evaluation of the proposed PDA, the instance  $P$  and the test case  $Q$  must involve the same DM (in this case  $DM_{sim}$  because (s)he is being simulated).

### Training set $T_{n,k}$

The training reference set  $T_{n,k}$  must be given by the DM. It is formed by  $n$  actions (alternatives, objects) each of which should be assigned to one of  $k$  classes. It is important to note that the classes are labelled from  $k$  down to 1, implying that they are organized from the worst to the best. Each alternative on the set is characterized by  $m$  criteria.

Figure 3 shows the general method to create a training reference set  $T_{n,k}$  from a  $DM_{sim}$ . The algorithm requires as input the specific parameter values  $\tau_0$  of the simulated decision maker, a sampling size  $N$  of actions, the number of classes  $k$ , the number of criteria  $m$ , the interval that will be used to define the values of each criterion, and the number of actions per class  $n_{class}$ . The first phase of the algorithm generates one action per class  $\{g_1, g_2, \dots, g_k\}$  that works as *central element*; these elements are used as reference of each class  $g_i$  in the second phase to incorporate new actions until the training reference set  $T_{n,k}$  is completely formed.

<p><b>GenerateInstance</b>(<math>\tau_0, N, k, m, [a, b], n_{class}</math>)</p> <p>1: <math>\{T, I\} \leftarrow</math> <b>GenerateCentralElements</b>(<math>\tau_0, N, k, m, [a, b]</math>)</p> <p>2: <math>\{T_{n,k}\} \leftarrow</math> <b>ExtendClasses</b>(<math>T, I, \tau_0, n_{class}, k, m, [a, b]</math>)</p> <p>3: <b>return</b> <math>T_{n,k}</math></p>
---

Figure 3. General method to create training reference sets  $T_{n,k}$

Figure 4a presents the pseudocode used to generate the *central elements* in the method **GenerateCentralElements(...)**. This algorithm creates the set  $I$  formed by the subsets  $\{I_1, I_2, \dots, I_k\}$ . Each subset  $I_i$  will contain at the end one action that will be representative of the class  $g_i$ . The construction of  $I$  is based on a trial and error method. First, an initial set  $O$  (of size  $N$ ) of random actions is created using **GenerateRandomActions(...)**, this functions assigns a random value chosen in the interval  $[a, b]$  to each of the  $m$  objectives of every action  $x \in O$  (see Line 3), and its net flow value  $F_n(x)$  is computed (see Lines 4 and 5, and Table 2). After that, the interval of variation of the net flow value is estimated in the whole set  $O$  (see Lines 6 to 8), and it is used to define equally spaced intervals that will characterize each class  $g_i$  (see Lines 9 to 11). Next, each action  $x \in O$  will be assigned to a subset  $I_i$  if its specific net flow value  $F_n(x)$  lies in the interval characterizing the class  $g_i$  (see Lines 15 to 19). Finally, the *central element* of a subset  $I_i$  is the action  $x \in I_i$  whose net flow value  $F_n(x)$  is closest to the average net flow value computed among the actions belonging to subset  $I_i$  (Line 19); ties are broken arbitrarily. If this algorithm remains with one subset  $I_i$  empty, the whole process is repeated (see Lines 26 and 27), otherwise, it adds each central element to  $T$  and returns along with the set  $I$ .

<p><b>GenerateCentralElements</b> (<math>\tau_0, N, k, m, [a, b]</math>)</p> <pre> 1: <math>I \leftarrow \{I_1, I_2, \dots, I_k\}</math> 2: <b>repeat</b> 3: <math>O \leftarrow \text{GenerateRandomActions}(N, m, [a, b])</math> 4: <b>for each</b> <math>x \in O</math> 5:   <math>F_x \leftarrow F_n(x)</math> 6: <math>F_{min} \leftarrow \min_{x \in O} \{F_x\}</math> 7: <math>F_{max} \leftarrow \max_{x \in O} \{F_x\}</math> 8: <math>Variation_{netflow} \leftarrow (F_{max} - F_{min}) / k</math> 9: <b>for each</b> <math>I_i \in I</math> 10:  <math>I_{i,ini} \leftarrow F_{min} + (i-1) \times Variation_{netflow}</math> 11:  <math>I_{i,fin} \leftarrow F_{min} + i \times Variation_{netflow}</math> 12:  <math>T \leftarrow \emptyset</math> 13:  <b>for each</b> <math>I_i \in I</math> 14:    <math>I_i \leftarrow \emptyset</math> 15:    <b>for each</b> <math>x \in O</math> 16:      <b>if</b> <math>F_x \geq I_{i,ini}</math> <b>and</b> <math>F_x &lt; I_{i,fin}</math> <b>then</b> 17:        <math>C_x \leftarrow g_i</math> 18:        <math>I_i \leftarrow I_i \cup x</math> 19:        <math>I_{i,avgflow} \leftarrow I_{i,avgflow} + F_x</math> 20:      <b>if</b> <math> I_i  &gt; 0</math> <b>then</b> 21:        <math>I_{i,avgflow} \leftarrow I_{i,avgflow} /  I_i </math> 22:        <math>I_{i,central} \leftarrow \arg \min_{x \in O} \{ F_x - I_{i,avgflow} \}</math> 23:        <math>I_i \leftarrow I_{i,central}</math> 24:        <math>T \leftarrow T \cup I_{i,central}</math> 25:      <math>Empty \leftarrow \{I_i \mid  I_i  = 0\}</math> 26:    <b>until</b> <math> Empty  = 0</math> 27:  <b>return</b> <math>\{I, T\}</math> </pre>	<p><b>ExtendClasses</b>(<math>T, I, \tau_0, n_{class}, k, m, [a, b]</math>)</p> <pre> 1: <math>n \leftarrow n_{class} \times k</math> 2: <math>T_{n,k} \leftarrow T</math> 3: <b>repeat</b> 4:  <math>y \leftarrow \text{GenerateRandomActions}(1, m, [a, b])</math> 5:  <math>C'_y \leftarrow \{g_1, g_2, \dots, g_k\}</math> 6:  <b>for each</b> <math>x \in T_{n,k}</math> 7:    <math>i \leftarrow C(x)</math> 8:    <b>if</b> <math>yP(\tau_0)x</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_i, g_{i+1}, \dots, g_k\}</math> 9:    <b>if</b> <math>xP(\tau_0)y</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_1, g_2, \dots, g_i\}</math> 10:   <b>if</b> <math>xQ(\tau_0)x</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_i, g_{i+1}\}</math> 11:   <b>if</b> <math>xQ(\tau_0)y</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_{i-1}, g_i\}</math> 12:   <b>if</b> <math>yI(\tau_0)x</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_i, g_{i+1}\}</math> 13:   <b>if</b> <math>yK(\tau_0)x</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_i, g_{i+1}\}</math> 14:   <b>if</b> <math>xK(\tau_0)y</math> <b>then</b> <math>C'_y \leftarrow C'_y \cap \{g_{i-1}, g_i\}</math> 15:   <b>if</b> <math> C'_y  \geq 0</math> <b>and</b> <math>\exists \{I_i \mid  I_i  &lt; n_{class}, g_i \subseteq C'_y\}</math> <b>then</b> 16:     <math>j \leftarrow \min_{g_i \subseteq C'_y} \{ I_i  &lt; n_{class}\}</math> 17:     <math>C_y \leftarrow g_j</math> 18:     <math>T_{n,k} \leftarrow T_{n,k} \cup y</math> 19:     <math>I_j \leftarrow I_j \cup y</math> 20:   <math>Full \leftarrow \{I_i \mid  I_i  = n_{class}\}</math> 21:  <b>until</b> <math> Full  = k</math> 22:  <b>return</b> <math>T_{n,k}</math> </pre>
---	--

a) Pseudocode to generate central elements of  $T_{n,k}$

b) Pseudocode to complete the training reference set  $T_{n,k}$

Figure 4. Pseudocode for methods **GenerateCentralElements(...)** and **ExtendClasses(...)**

The training set  $T_{n,k}$  is initialized with the set  $T$  obtained from **GenerateCentralElements(...)**, and it is completed in the algorithm shown in Figure 4b. This algorithm generates one new action  $y$  at a time (see Line 4), and it identifies the possible classes where it could belong according to its relation with the actions  $x$  already in  $T_{n,k}$  (see Lines 6 to 14). If there are one or more classes for action  $y$ , it is chosen the class  $g_i$  with the smallest value  $i$  and whose related subset  $I_i$  has not been completed (i.e.  $|I_i| < n_{class}$ ); otherwise the new action  $y$  is discarded (see Lines 15 to 20). The **ExtendClasses(...)** algorithm continues until all the classes are completed, and return the generated training set  $T_{n,k}$  (see Lines 21 and 22).

The functions used by methods of Figure 4 are: a) the **GenerateRandomActions( $n, m, [a, b]$ )**, which returns  $n$  actions with  $m$  objectives each with values defined in the range  $[a,b]$ ; b) the function **C( $x$ )**, which provides the index  $i$  of the class  $g_i$  to which action  $x$  belongs; and, c) the function  **$F_n(x)$** , used to compute the net flow score (and defined in Table 2).

Table 2. The computation of the net flow value

Set / Measure	Conditions
Net Flow (NF)	$F_n(x) = \sum_{y \in O \setminus \{x\}} \sigma(x,y) - \sigma(y,x),$ where $F_n(x) > F_n(y)$ denotes a certain preference of $x$ over $y$

Also, the algorithm uses several structures to keep track of the information required for the construction of  $T_{n,k}$ , which are: i)  $C_x$ , the class; ii)  $C_y$ , the group of possible classes to assign action  $y$  that is compatible with the actual set  $T_{n,k}$ ; and, iii) the elements  $F_x, I_{i,avgflow}, I_{i,central}$  which contain information of the net flow score of an action  $x$ , the average net flow score of the actions  $I_i$  of a class  $g_i$ , and the central element of a class  $g_i$ , respectively. The algorithm ends when each class has  $N_{max}$ , then it combines them to form  $T_{n,k}$  and returns it.

**Test set  $T_n$**

The test set  $T_n$  contains a larger number of actions than  $T_{n,k}$ . This set also defines its members (actions) by their  $m$  criterion values. Each pair  $(x,y) \in T_n \times T_n$  is associated with a binary relation of the set  $A = \{P, I, Q, K, R, \sim, P^{-1}, Q^{-1}, K^{-1}\}$  using  $\tau_0$ , the characteristic parameters of the  $DM_{sim}$ .

The set  $T_n$  is also generated through the function **GenerateRandomActions( $n,m,[a,b]$ )**, as done previously for  $T_{n,k}$ . It requires as input the number of actions  $n$ , the number of criteria  $m$  describing each action, and the range  $[a, b]$  within which each criterion value is randomly generated. The set  $A = \{P, I, Q, K, R, \sim, P^{-1}, Q^{-1}, K^{-1}\}$  contains the preference statements that the  $DM_{sim}$  establishes on every  $(x,y) \in T_n \times T_n$ , having in mind that he/she can only assign one relation per pair. To simulate the DM's responses, we assume that the relation  $x A_i y$  holds if and only if  $x A_i(\eta_0, \lambda_0, \beta_0, \epsilon_0) y$  where  $A_i \in A$  and  $A_i(\eta_0, \lambda_0, \beta_0, \epsilon_0)$  is one of the relations given in Table 1. That is, the preference statement made by simulated DMs when (s)he is questioned about his/her preference on a pair of actions  $(x, y)$  corresponds to  $x A_i(\eta_0, \lambda_0, \beta_0, \epsilon_0) y$ .

Finally, the creation of the test set  $T_n$  allows the assessment of the performance of the proposed PDA approach in different events than those used to elicit the parameter values. In other words, the set  $T_n$  represents future decisions of a DM that can be available to test the preference model generated to characterize the DM. This work defines such preference statements as relations established by the DM among pairs of actions.

### 2.3. Definition of the sets of instances

The PDA approach proposed in Section 1.4 was validated using different instances of Problem (3) as training sets, and different test sets.

The set  $S_{42,3}$  is the training instances set and it is formed by 16 instances that were used to train the PDA approach to elicit the parameters of 16 different simulated DMs. Particularly, each instance  $\Lambda_i = (DM_{sim\text{-}parameters}, T_{n,k})$  on  $S_{42,3}$  contains a specific randomly generated  $DM_{sim\text{-}parameters}$ , and a training reference set  $T_{42,3}$  created using **GenerateInstance(...)**. The inputs received by this algorithm were the parameter values  $\tau_0$  corresponding to that  $DM_{sim\text{-}parameters}$ , a sampling size  $N = 1000$ , a number of classes  $k = 3$ , a number of criteria of  $m = 10$ , the interval  $[1, 10]$  for each criterion, and  $n_{class}=14$  as the number of actions per class. The training reference set  $T_{42,3}$  size of 42 actions is considered a reasonable size for simulation due to it is a size that would not require an excessive effort from a real DM.

The set  $S_{100}$  is the test cases set, and it contains also 16 cases; each test case  $\Gamma_i = (DM_{sim\text{-}parameters}, T_n)$  on  $S_{100}$  contains the  $DM_{sim\text{-}parameters}$  associated to its corresponding instance  $\Lambda_i$  in  $S_{42,3}$ , and a test set  $T_{100}$ . The set  $T_{100}$  is formed by 100 actions characterized by 10 criteria, whose values are randomly defined in the interval  $[1, 10]$ , and, by 9,900 different relations among pairs of actions. These preference relations are defined using the  $DM_{sim}$ , or  $\tau_0$ , and according to Table 1; therefore, each test case of  $S_{100}$  contains 9,900 preference judgments that express preferences from simulated DMs.

## 3. Experimental design

The conducted experimental design evaluates the performance of the PDA based method proposed in Section 1.4. This method elicits parameters from a battery of sorted samples organized in classes that reflect the preferences of a DM. The content of this experiment is organized as follows. The first part describes the indicator of quality used to evaluate the solution set offered by the PDA; this indicator is based on the inconsistencies produced by the approach using the estimated parameters' values obtained from each reference set  $T_{n,k}$ . The second part describes the evaluation done over the optimization approach; this part details the statistical analysis used to demonstrate whether there is a significant difference between the performance of the optimization approach when using different metaheuristics in its implementation.

### 3.1. Performance evaluation

The indicator  $I_p$ , or estimated error, defined in this section, is used to evaluate the quality of the solutions provided by the approach presented in Section 1.4. This indicator evaluates

the prediction capacity of the best solution set  $\tau^* = (\eta^*, \lambda^*, \beta^*, \varepsilon^*)$  obtained for each instance  $\Lambda_i \in S_{42,3}$ . Such prediction capacity is measured by computing the error in the estimation of preferences on the corresponding test case  $\Gamma_i \in S_{100}$ .

Given an instance  $\Lambda_i = (DM_{sim\text{-}parameters}, T_{42,3})$  of  $S_{42,3}$ , its corresponding test case  $\Gamma_i = (DM_{sim\text{-}parameters}, T_{100})$ , the best solution  $\tau^*$  estimated by the proposed PDA approach, and the parameters' values setting  $\tau_0 = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$  associated with  $DM_{sim\text{-}parameters}$ , the calculation of the estimated error  $I_p$  can be defined as in Eq. (4).

$$I_p = \sum_{(x,y) \in T_{100} \times T_{100}} f_1(x,y), \text{ where } f_1(x,y) = \begin{cases} 1, & \text{if } xP(\tau_0)y \leftrightarrow xP(\tau^*)y \text{ is false} \\ 0, & \text{otherwise} \end{cases}. \quad (4)$$

The calculation of  $I_p$  assesses the inconsistencies existing between the best solution  $\tau^*$  to Problem (3) derived from the optimization approach, and the parameters  $\tau_0$  that define the simulated DM. This indicator defines as *inconsistency* the situation in which a strict preference relation  $xP(\tau_0)y$  established by the  $DM_{sim}$  on the training set  $T_{42,3}$  is not in concordance with the preference relation  $xA(\tau^*)y$  defined by the solution  $\tau^*$  given by the PDA, i.e the estimated parameters' values failed in the prediction of the preference relation and they predicted  $xA(\tau^*)y$  for a relation  $A \neq P$ .

In summary, the Eq. (4) can be used to measure the quality of the solutions obtained from  $S_{42,3}$  in the optimization approach by using  $\tau^*$  to define the preference relations in a new set  $S_{100}$ . Then,  $I_p$  is the indicator of the performance of the parameter elicitation method proposed in this work.

### 3.2. Statistical evaluation

The analysis of the performance of the optimization approach proposed in this work is evaluated using three different strategies. The first strategy uses the proposed GA that looks for a solution in a search space defined for all the parameters elicited within a range deviated from the expected value on  $\pm 30\%$ , this strategy is denoted  $\mu^{*1}$ . The second strategy (denoted  $\mu^{*2}$ ) uses also the same GA, and it also looks for a solution in a search space defined in the range of  $\pm 30\%$  of the expected values in all the parameters but  $\lambda$ , which is reduced to  $\pm 10\%$ . The purpose of this experiment is to examine the importance of a proper definition of this parameter. The last strategy uses the metaheuristic Simulated Annealing on its most basic definition, and it also considers the search space defined in a range deviated 30% from the expected parameter values, this strategy is denoted  $\mu^{*3}$ , and its purpose consists in to evaluate the performance of the proposed PDA using a different metaheuristic.

The evaluation is centered in the identification of the best strategy to estimate the preference parameters from the DM's judgments. The proposed optimization approach can be used to elicit parameters, and it is evaluated on the performance of different metaheuristics, and how the search space can influence in a proper identification of parameters. For this purpose, it is analysed in this section if there is or not significant differences among  $\mu^{*1}$ ,  $\mu^{*2}$ , and  $\mu^{*3}$ .

The statistical analysis uses the indicator  $I_p$  to measure the quality of the solutions derived from the optimization approach. Firstly, the indicator is computed using the best solutions

obtained by each strategy  $\mu^{*i}$  on each instance from  $S_{42,3}$ , and the test cases set  $S_{100}$ . After that, statistical tests were applied over the sorted data to assess the significance of the results. For this purpose, the STAC Web Platform<sup>1</sup>, (a statistical tool equipped with different statistical tests), was used. Within this platform, the selected tests where the *Non-parametric multiple groups All vs All*, with the particular case of the *Friedman test* as Ranking Test. Additionally, they were specified as Post-hoc methods for  $p$ -value adjustment the tests: a) Finner; b) Nemenyi; c) Holm; and d) Shaffer. The significance level was set at 0.05. In the Ranking Test, the null hypothesis  $H_0$  was: *The means of the results of two or more algorithms are the same*. In the post-hoc analysis, the null hypothesis  $H_0$  was: *The mean of the results of each pair of groups is equal*.

#### 4. Results

This section presents the results derived from analysis of the optimization approach and its implementations  $\mu^{*1}$ ,  $\mu^{*2}$ , and  $\mu^{*3}$  based on different metaheuristics and configurations. The results are organized into two subsections. The first subsection presents the summary of the performance indicator  $I_p$  using the estimations obtained with the set of instances  $S_{42,3}$ , and the test cases set  $S_{100}$ . The second subsection summarizes the results from the statistical evaluation; according to these data, the method reveals that the strategy using a change over the range of  $\pm 10\%$  from the expected values of the parameters has the best performance, showing significant difference in contrast with the other two strategies; this reveals the importance for a DM to invest a bit of time trying to figure out proper ranges of values to estimate parameters of a preference model.

Let us note that the performance of all the indicators  $\mu^*_{perf}$  in the set  $S_{42,3}$  was of 0 inconsistencies, i.e. the preference model with the estimated parameter values given by the optimization approach reflects the preferences of the DM. The estimation of parameters was used with strategies having a stop criterion of 100000 evaluations, and a population of 100 individuals (in the case of the GA). The results from the evaluation of the performance of each solution, and from the analysis of robustness are presented in the following section.

##### 4.1. Results from evaluation of the performance

The first part of the experiment measured the performance evaluation using the indicator  $I_p$  over each test case  $\Gamma_i \in S_{100}$ . Table 3 presents the values achieved for each instance  $\Lambda_i \in S_{42,3}$ . The 16 instances are organized in four sub-tables. Each sub-table is organized in sets of three columns representing an instance and the performance  $\mu^{*1}$ ,  $\mu^{*2}$ ,  $\mu^{*3}$  of each strategy given by  $I_p$ , i.e. the number of inconsistencies; this performance is contrasted against the expected number of strict preference relations (given by the row titled *Total*), and the number of correct predictions (shown in the row *Consisten.*), according to the corresponding test set of the case  $\Gamma_i$ .

<sup>1</sup> Statistical Web Tool found at URL: <http://tec.citius.usc.es/stac/ranking.html>

Table 3. Comparison of performance among PDA strategies using  $I_p$

a) Instances 1 to 4

Instance	$\Lambda_1$			$\Lambda_2$			$\Lambda_3$			$\Lambda_4$		
Strategy	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$
Total	529	555	558	540	501	578	322	338	376	309	313	354
$I_p$	75	15	28	72	41	66	30	26	56	29	25	62
Consisten.	454	540	530	468	460	512	292	312	320	280	288	292

b) Instances 5 to 8

Instance	$\Lambda_5$			$\Lambda_6$			$\Lambda_7$			$\Lambda_8$		
Strategy	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$
Total	517	521	504	319	291	326	227	247	285	211	216	248
$I_p$	61	29	28	55	27	46	31	21	43	33	26	56
Consisten.	456	492	476	264	264	280	196	226	242	178	190	192

c) Instances 9 to 12

Instance	$\Lambda_9$			$\Lambda_{10}$			$\Lambda_{11}$			$\Lambda_{12}$		
Strategy	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$
Total	527	553	562	325	340	375	365	385	411	251	270	306
$I_p$	45	23	22	35	30	47	47	27	55	55	30	64
Consisten.	482	530	540	290	310	328	318	358	356	196	240	242

d) Instances 13 to 16

Instance	$\Lambda_{13}$			$\Lambda_{14}$			$\Lambda_{15}$			$\Lambda_{16}$		
Strategy	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$
Total	426	435	416	395	397	443	243	281	297	229	248	310
$I_p$	32	23	18	37	33	49	35	27	37	27	26	78
Consisten.	394	412	398	358	364	394	208	254	260	202	222	232

### 4.2. Results from the statistical evaluation

Table 4 summarizes the values of the indicator  $I_p$ ; these values are used to make the statistical evaluation. The first column shows the instances, the remaining ones contain the value of  $I_p$  per strategy.

The result from the Friedman ranking test, with a significant level of 0.05, rejects the null hypothesis  $H_0$ : *the means of the results of two or more algorithms are the same*. Table 5 shows the ranking derived from this test, which presents the strategy  $\mu^{*2}$  as the best algorithm, i.e. the use of the GA with a better range of search for the value of the parameter  $\lambda$ .

Tables 6a, 6b, 6c, and 6d summarize the results of post-hoc analysis, for  $S_{100}$  using Finner, Nemenyi, Holm, and Shafer comparisons, respectively. The null hypothesis tested was  $H_0$ : *the mean of the results of each pair of groups is equal*. In all the comparisons, the hypothesis



Table 4. Summary of inconsistencies derived from indicator  $I_p$

Instance $\Lambda$	$\mu^{*1}$	$\mu^{*2}$	$\mu^{*3}$
$\Lambda_1$	75	15	28
$\Lambda_2$	72	41	66
$\Lambda_3$	30	26	56
$\Lambda_4$	29	25	62
$\Lambda_5$	61	29	28
$\Lambda_6$	55	27	46
$\Lambda_7$	31	21	43
$\Lambda_8$	33	26	56
$\Lambda_9$	45	23	22
$\Lambda_{10}$	35	30	47
$\Lambda_{11}$	47	27	55
$\Lambda_{12}$	55	30	64
$\Lambda_{13}$	32	23	18
$\Lambda_{14}$	37	33	49
$\Lambda_{15}$	35	27	37
$\Lambda_{16}$	27	26	78

Table 5. Ranking derived from Friedman test

Rank	Algorithm
1.18750	$\mu^{*2}$
2.37500	$\mu^{*1}$
2.43750	$\mu^{*3}$

is rejected when the strategy  $\mu^{*2}$  is compared against the other two, implying that there is significant difference in the performance achieved when using it. In the other side, the comparisons accepted the hypothesis when using the strategies  $\mu^{*1}$  vs  $\mu^{*3}$  which indicates that they have a similar performance.

Finally, derived from the above results, let us note that the proposed optimization approach estimates properly the parameter values of a preference model from a battery of assignment examples. This claim is supported by the fact that the elicitation performed by all the strategies reduced the number of inconsistencies to 0, in all the instances of the set  $S_{42,3}$ . Also, it was observed that a proper definition of the bounds of the parameters under study limits the search space with a positive impact on their elicitation; this is corroborated by the result obtained from the controlled reduction of the feasible region of  $\lambda$  in the strategy  $\mu^{*2}$  that shows a better estimation of parameter values, with a significant difference in its performance (given by  $I_p$ ), when it was compared against  $\mu^{*1}$  and  $\mu^{*3}$  configured with a different feasible region. The last fact reflects the importance that has the involvement of the DM on the proper definition of feasible bounds for the preference model’s parameters.

Table 6. Results of the post-hoc multiple comparisons derived from the ranking test

## a) Finner post-hoc comparison

Comparison	Statistic	Adjusted p-value	Result
$\mu^{*1}$ vs $\mu^{*2}$	3.35876	0.00122	$H_0$ is rejected
$\mu^{*2}$ vs $\mu^{*3}$	3.53553	0.00122	$H_0$ is rejected
$\mu^{*1}$ vs $\mu^{*3}$	0.17678	0.85968	$H_0$ is accepted

## b) Nemenyi post-hoc comparison

Comparison	Statistic	Adjusted p-value	Result
$\mu^{*1}$ vs $\mu^{*2}$	3.35876	0.00235	$H_0$ is rejected
$\mu^{*2}$ vs $\mu^{*3}$	3.53553	0.00122	$H_0$ is rejected
$\mu^{*1}$ vs $\mu^{*3}$	0.17678	1.00000	$H_0$ is accepted

## c) Holm post-hoc comparison

Comparison	Statistic	Adjusted p-value	Result
$\mu^{*1}$ vs $\mu^{*2}$	3.35876	0.00157	$H_0$ is rejected
$\mu^{*2}$ vs $\mu^{*3}$	3.53553	0.00122	$H_0$ is rejected
$\mu^{*1}$ vs $\mu^{*3}$	0.17678	0.85968	$H_0$ is accepted

## d) Shaffer post-hoc comparison

Comparison	Statistic	Adjusted p-value	Result
$\mu^{*1}$ vs $\mu^{*2}$	3.35876	0.00157	$H_0$ is rejected
$\mu^{*2}$ vs $\mu^{*3}$	3.53553	0.00122	$H_0$ is rejected
$\mu^{*1}$ vs $\mu^{*3}$	0.17678	0.85968	$H_0$ is accepted

## Conclusions

The present work deals with the parameter elicitation problem in multi-criteria decision problems based on a PDA strategy. It defines a bi-objective optimization problem to elicit preference parameter values, and it proposes a strategy based on single-objective metaheuristics to solve it. The strategy obeys a lexicographical order identified in the optimization approach, and it is tested extensively through two distinct metaheuristics, and different settings for the parameter search space.

The performance analysis of the proposed strategy was evaluated based on two experimental designs. The first one considers the application of two different metaheuristics, i.e. the Genetic and the Simulated Annealing algorithms. The results from this experiment reveal a significant statistical difference, in favour of the Genetic Algorithm (GA), on the capacity of prediction obtained from the estimated parameter values. In other words, the proposed elicitation strategy estimates better the parameter values for the preference model when the GA is used.

The second experimental design involves the use of the GA and two distinct configurations of the search space defined for the credibility threshold  $\lambda$ . The results show that a search space deviated  $\pm 10\%$  from the expected value of  $\lambda$  improves the quality of the estimated parameter values from those obtained from a search space deviated  $\pm 30\%$ . It was observed that the GA estimated parameters values that when used to predict new situations obtained a statistically significant reduction in the number of inconsistencies. This information provides evidence of the importance of an adequate definition of the range of possible values of the parameters to be elicited by the DM, as the case of  $\lambda$ , because the better the bounds that are provided, the better estimations that are achieved by the proposed elicitation strategy.

Finally, let us note that this research also contributes with an approach to simulate Decision Maker (DM) preferences using actions organized in an ordinal ordered set of classes, a strategy that can be used to generate benchmark of random reference sets to test future optimization approaches for related problems. Also, let us point out that the bi-objective strategy does not scale with the number of criteria involved in the actions considered, and hence it can deal with high dimensional actions.

## Acknowledgements

The authors acknowledge the support from: a) CONACYT projects with numbers 3058, 236154, and 1340; b) TecNM projects with numbers 6308.17-P, 6683.18-P; and, c) *Universidad Autónoma de Ciudad Juárez*.

## Author contributions

EF conceived the study and the inference model. NRV was the main responsible for the design and development of the optimization approach and the strategy followed to artificially simulate decision makers. LCR, CGGS, PSS and GRZ were in charge for data collection, experimental design and analysis, and EF, LCR and NRV performed the interpretation of data. The first draft of the article was conceived and reviewed by all the authors.

## Disclosure statement

The authors declare that they have no conflicts of interest.

## References

- Bastiani, S., Cruz-Reyes, L., Fernandez, E., & Gomez, C. (2015). Portfolio optimization from a set of preference ordered projects using an ant colony based multi-objective approach. *International Journal of Computational Intelligence Systems*, 8, 41-53. <https://doi.org/10.1080/18756891.2015.1129590>
- Bechikh, S. (2013). *Incorporating decision maker's preference information in evolutionary multi-objective optimization* (Doctoral dissertation, High Institute of Management of Tunis, University of Tunis, Tunisia). Retrieved from <http://delta.cs.cinvestav.mx/~ccoello/EMOO/thesis-bechikh.pdf.gz>
- Branke, J. (2015). MCDA and multiobjective evolutionary algorithms. In S. Greco, M. Ehrgott, & J. Figueira (Eds.), *Multiple criteria decision analysis: state of the art surveys* (2<sup>nd</sup> ed.). Springer. [https://doi.org/10.1007/978-1-4939-3094-4\\_23](https://doi.org/10.1007/978-1-4939-3094-4_23)

- Branke, J., Greco, S., Słowiński, R., & Zielniewicz, P. (2015). Learning value functions in interactive evolutionary multiobjective optimization. *IEEE Transactions on Evolutionary Computation*, 19(1), 88-102. <https://doi.org/10.1109/TEVC.2014.2303783>
- Branke, J., Salvatore, C., Greco, S., Słowiński, & Zielniewicz, P. (2016). Using Choquet integral as preference model in interactive evolutionary multiobjective optimization. *European Journal of Operational Research*, 250, 884-901. <https://doi.org/10.1016/j.ejor.2015.10.027>
- Coello, C. A. (1999). A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information Systems, an International Journal*, 1(3), 269-308. <https://doi.org/10.1007/BF03325101>
- Coello, C. A., Lamont, G. B., & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2<sup>nd</sup> ed.). New York: Springer. Retrieved from <http://www.springer.com/us/book/9780387332543>.
- Cruz-Reyes, L., Fernandez, E., Sanchez, P., Coello Coello, C. A., & Gomez, C. (2017). Incorporation of implicit decision-maker preferences in multi-objective evolutionary optimization using a multi-criteria classification method. *Applied Soft Computing*, 50, 48-57. <https://doi.org/10.1016/j.asoc.2016.10.037>
- Deb, K., & Agrawal, R. B. (1995). Simulated binary crossover for continuous search space. *Complex Systems*, 9, 115-148.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester-New York-Weinheim-Brisbane-Singapore-Toronto: John Wiley & Sons. Retrieved from <http://dl.acm.org/citation.cfm?id=559152>
- Deb, K. (2007). Current trends in evolutionary multi-objective optimization. *International Journal for Simulation and Multidisciplinary Design Optimisation*, 1(1), 1-8. <https://doi.org/10.1051/ijsmdo:2007001>
- Deb, K., & Tiwari, S. (2008). Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185, 1062-1087. <https://doi.org/10.1016/j.ejor.2006.06.042>
- Deb, K., Sinha, A., Korhonen, P., & Wallenius, J. (2010). An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE T Evolut Comput*, 14(5), 723-739. <https://doi.org/10.1109/TEVC.2010.2064323>
- Dias, L., & Mousseau, V. (2006). Inferring electre's veto-related parameters from outranking examples. *European Journal of Operational Research*, 170, 172-191. <https://doi.org/10.1016/j.ejor.2004.07.044>
- Doumpos, M., Marinakis, Y., Marimaki, M., & Zopounidis, C. (2009). An evolutionary approach to construction of outranking models for multicriteria classification: The case of ELECTRE TRI method. *European Journal of Operational Research*, 199, 496-505. <https://doi.org/10.1016/j.ejor.2008.11.035>
- Fernandez, E., Navarro, J., & Bernal, S. (2009). Multicriteria sorting using a valued indifference relation under a preference disaggregation paradigm. *European Journal of Operational Research*, 192, 602-609. <https://doi.org/10.1016/j.ejor.2008.09.020>
- Fernandez, E., Lopez, E., Lopez, F., & Coello Coello, C. A. (2011). Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. *Information Science*, 181, 44-56. <https://doi.org/10.1016/j.ins.2010.09.007>
- Fernandez, E., Navarro, J., & Mazcorro, G. (2012). Evolutionary multi-objective optimization for inferring outranking model's parameters under scarce reference information and effects of reinforced preference. *Foundations of Computing and Decision Sciences*, 37, 163-197. <https://doi.org/10.2478/v10209-011-0010-0>
- Fernandez, E., Lopez, E., Mazcorro, G., Olmedo, R., & Coello, C. (2013). Application of the non-outranked sorting genetic algorithm to public project portfolio selection. *Information Sciences*, 228, 131-149. <https://doi.org/10.1016/j.ins.2012.11.018>

- Fernandez, E., Gomez, C., Rivera, G., & Cruz-Reyes, L. (2015). Hybrid Metaheuristic approach for handling many objectives and decisions on partial support in project portfolio optimization. *Information Sciences*, 315, 102-122. <https://doi.org/10.1016/j.ins.2015.03.064>
- Gastelum Chavira, D. A., Leyva Lopez, J. C., Solano Noriega, J. J., Ahumada Valenzuela, O., & Alvarez Carrillo, P. A. (2017). A credit ranking model for a parafinancial company based on the ELECTRE-III method and a multiobjective evolutionary algorithm. *Applied Soft Computing*, 60(1), 190-201. <https://doi.org/10.1016/j.asoc.2017.06.021>
- Greco, S., Mousseau, V., & Slowinski, R. (2008). Ordinal regression revisited: Multiple criteria ranking with a set of additive value functions. *European Journal of Operational Research*, 191, 415-435. <https://doi.org/10.1016/j.ejor.2007.08.013>
- Greco, S., Slowinski, R., Figueira, J. R., & Mousseau, V. (2010). Robust ordinal regression. *Trends in Multiple Criteria Decision Analysis*, 142, 241-283. [https://doi.org/10.1007/978-1-4419-5904-1\\_9](https://doi.org/10.1007/978-1-4419-5904-1_9)
- Kahraman, C., Engin, O., Kaya, İ., & Kerim Yilmaz, M. (2012). An application of effective genetic algorithms for solving hybrid flow shop scheduling problems. *International Journal of Computational Intelligence Systems*, 1(2), 134-147. <https://doi.org/10.1080/18756891.2008.9727611>
- Meghwani, S. S., & Thakur, M. (2017). Multi-criteria algorithms for portfolio optimization under practical constraints. *Swarm and Evolutionary Computation*, 37(1), 104-125. <https://doi.org/10.1016/j.swevo.2017.06.005>
- Miller, G. A. (1956). The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychological Review*, 63(2), 81-97. <https://doi.org/10.1037/h0043158>
- Molina, J., Santana-Quintero, L. V., Hernández-Díaz, A. G., Coello Coello, C. A., & Caballero, R. (2009). g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, 197(2), 685-692. <https://doi.org/10.1016/j.ejor.2008.07.015>
- Mousseau, V., & Slowinski, R. (1998). Inferring an electre-tri model from assignment examples. *Journal of Global Optimization*, 12, 157-174. <https://doi.org/10.1023/A:1008210427517>
- Mousseau, V., Figueira, J., & Naux, J. P. (2001). Using assignment examples to infer weights for ELECTRE TRI method: some experimental results. *European Journal of Operational Research*, 130(2), 263-275. [https://doi.org/10.1016/S0377-2217\(00\)00041-2](https://doi.org/10.1016/S0377-2217(00)00041-2)
- Mousseau, V., & Dias, L. C. (2004). Valued outranking relations in ELECTRE providing manageable disaggregation procedures. *European Journal of Operational Research*, 156(2), 467-482. [https://doi.org/10.1016/S0377-2217\(03\)00120-6](https://doi.org/10.1016/S0377-2217(03)00120-6)
- Phelps, S., & Koksalan, M. (2003). An interactive evolutionary metaheuristic for multiobjective combinatorial optimization. *Management Science*, 49(12), 1617-1770. <https://doi.org/10.1287/mnsc.49.12.1726.25117>
- Roy, B. (1991). The outranking approach and the foundations of electre methods. *Theory and Decisions*, 31(1), 49-73. <https://doi.org/10.1007/BF00134132>
- Roy, B. (1996). *Multicriteria methodology for decision aiding*. Boston, MA: Springer. <https://doi.org/10.1007/978-1-4757-2500-1>
- Wagner, T., & Trautmann, H. (2010). Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE Transactions on Evolutionary Computation*, 14, 688-701. <https://doi.org/10.1109/TEVC.2010.2058119>
- Xiaohan, Y., Suojan, Z., Xianglin, L., & Xiuli, Q. (2018). ELECTRE methods in prioritized MCDM environment. *Information Sciences*, 424(1), 301-316. <https://doi.org/10.1016/j.ins.2017.09.061>
- Xiong, N., Molina, D., Leon Ortiz, M., & Herrera, F. (2015). A walk into metaheuristics for engineering optimization: principles, methods and recent trends. *International Journal of Computational Intelligence Systems*, 8(4), 606-636. <https://doi.org/10.1080/18756891.2015.1046324>

Zhu, Y., & Luo, Y. (2016). Multi-objective optimisation and decision-making of space station logistics strategies. *International Journal of Systems Science*, 47(13), 3132-3148. <https://doi.org/10.1080/00207721.2015.1091898>

Zolfani, S. H., Maknoon, R., & Zavadskas, E. K. (2016). An introduction to prospective multiple attribute decision making (PMADM). *Technological and Economic Development of Economy*, 22(2), 309-326. <https://doi.org/10.3846/20294913.2016.1150363>

## APPENDIX

### Calculation of the credibility index of outranking

Let  $G$  be a family of  $N$  consistent criteria which are used as basis for the DM's judgments about actions. The ELECTRE methods introduced a binary relation  $S$  that is a model of the preference relation "at least as good as". Proposition  $xSy$  (' $x$  outranks  $y$ ') holds if, and only if, the  $DM$  has sufficient arguments in favour of ' $x$  is at least as good as  $y$ ' and there are no strong arguments against it. In more formal way, the coalition of the criteria in agreement with that proposition is strong enough, and there is no important coalition discordant with it. This can be expressed by the following logical equivalence:

$$xSy \leftrightarrow C(x, y) \wedge \sim D(x, y),$$

where:

- $C(x,y)$  is the predicate about the strength of the concordance coalition; it is composed by the criteria subsets  $C(xSy) = \{g_i \in G \mid g_i(x) - g_i(y) \geq -q_i\}$  and  $C(yQx) = \{g_i \in G \mid g_i(y) - p_i \leq g_i(x) < g_i(y) - q_i\}$ . Observe that  $Q$  denotes weak preference, and  $p_i$  and  $q_i$  denote the preference and indifference thresholds for criterion  $i$ ,  $p_i \geq q_i \geq 0$ .
- $D(x,y)$  is the predicate about the strength of the discordance coalition; it is formed by the criteria subset  $C(yPx) = \{g_i \in G \mid g_i(y) - g_i(x) \geq p_i\}$ , here  $P$  denotes *strict preference*.
- $\wedge$  and  $\sim$  are the logical connectives for fuzzy conjunction and fuzzy negation, respectively.

Let  $c(x, y)$  denote the truth degree of predicate  $C(x, y)$ . Using the *product* operator for conjunction, in ELECTRE III the degree of credibility  $\sigma(x,y)$  of  $xSy$  is calculated by:

$$\sigma(x, y) = c(x, y) \cdot Nd(x, y),$$

where  $Nd(x, y)$  denotes the truth degree of the non-discordance predicate. The concordance index  $c(x,y)$  is defined as follows:

$$c(x, y) = \sum_{g_i \in G} w_i \cdot c_i(x, y),$$

where  $w_i$  is the weight of the  $i$ -th criterion ( $w_1 + w_2 + \dots + w_N = 1$ ), and  $c_i(x, y)$  is the marginal (partial) concordance index for the  $i$ -th criterion. This index is calculated by:

$$c_i(x, y) = \begin{cases} 0 & \text{if } g_i \in C(yPx) \\ \frac{g_i(x) - g_i(y) + p_i}{p_i - q_i} & \text{if } g_i \in C(yQx) \\ 1 & \text{if } g_i \in C(xSy) \end{cases}$$

The partial discordance is measured in comparison with a veto threshold  $v_i$ , which is the maximum difference obtained from  $g_i(y) - g_i(x)$  that is compatible with  $\sigma(x, y) > 0$ . Following Mousseau and Dias (2004), we shall use a simplification of the original formulation of the discordance indices in the ELECTRE III method given by:

$$Nd(x, y) = \min_{g_i \in C(yPx)} \{1 - d_i(x, y)\}.$$

The value of  $d_i(x, y)$  is calculated by:

$$d_i(x, y) = \begin{cases} 1 & \text{if } f \nabla_i \geq v_i \\ \frac{\nabla_i - u_i}{v_i - u_i} & \text{if } f u_i < \nabla_i < v_i, \\ 0 & \text{if } f \nabla_i \leq u_i \end{cases}$$

where  $\nabla_i = g_i(y) - g_i(x)$  and  $u_i$  is a discordance (pre-veto) threshold. Finally, and given that the degree of credibility  $\sigma(x, y)$  is calculated by combining all the previous equations, the following parameters must be specified:

1. the vector of weights  $w$ ;
2. the vector of indifference thresholds  $q$ ;
3. the vector of preference thresholds  $p$ ;
4. the vector of veto thresholds  $v$ ;
5. the vector of discordance thresholds  $u$ .

Additionally, if a crisp outranking relation is built on the universe, then a cutting level  $\lambda^*$  should be specified.