

A metaheuristic optimization-based indirect elicitation of preference parameters for solving many-objective problems

Laura Cruz-Reyes¹, Eduardo Fernandez², Nelson Rangel-Valdez³

¹ Postgraduate and Research Division,
National Mexican Institute of Technology/Madero Institute of Technology,
Cd. Madero, Tamaulipas 89440, México
E-mail: lauracruzreyes@itcm.edu.mx

² Faculty of Civil Engineering,
Autonomous University of Sinaloa,
Culiacan, Sinaloa 80040, México
E-mail: eddyf171051@gmail.com

³ CONACYT Research Fellow, Postgraduate and Research Division,
National Mexican Institute of Technology/Madero Institute of Technology,
Cd. Madero, Tamaulipas 89440, México
E-mail: nrangelva@conacyt.mx

Received 17 May 2016

Accepted 1 August 2016

Abstract

A priori incorporation of the decision maker's preferences is a crucial issue in many-objective evolutionary optimization. Some approaches characterize the best compromise solution of this problem through fuzzy outranking relations; however, they require the elicitation of a large number of parameters (weights and different thresholds). This paper proposes a novel metaheuristic-based optimization method to infer the model's parameters of a fuzzy relational system of preferences, based on a small number of judgments given by the decision maker. The results show a satisfactory rate of error when predicting new outcomes with the parameter values obtained by using small size reference sets.

Keywords: metaheuristic, decision aid, parameter inference, indirect approach, preference analysis disaggregation.

1. Introduction

A wide variety of problems in the real world often involve multiple objectives to be minimized or maximized simultaneously^{1,2}. As a consequence of the conflicting nature of the criteria, it is not possible to obtain a single optimum, and consequently, the ideal solution to a multi-objective optimization problem (MOP) cannot be reached. Unlike single-objective optimization, the best solution of a MOP is not well-defined from a purely mathematical point of view. To solve a MOP means to find the best compromise solution according to the decision maker's (DM's) particular system of preferences. Since all the

compromise solutions are mathematically equivalent, the DM should provide some additional information for choosing the most preferred one³.

Multi-Objective Evolutionary Algorithms (MOEAs) are particularly attractive to solve MOPs because they deal simultaneously with a set of possible solutions (the MOEA's population), which allows to obtain an approximation of the Pareto frontier in a single run of the algorithm. Thus, by using MOEAs the DM and/or the decision analyst does not need to perform a set of separate single-objective optimizations (as is usually required when using conventional multi-objective programming methods) in order to generate compromise

solutions. Additionally, MOEAs are more robust regarding the mathematical characteristics of objective functions and constraints, whereas these are real concerns for mathematical programming methods (cf. Ref. 4).

Most current approaches in evolutionary multi-objective optimization concentrate on generating an approximation of the Pareto optimal set. The DM still has to choose the best compromise solution out of that set. Nevertheless, as was stated in Ref. 5, there has been an increasing interest in incorporating the DM's preference information in the search process. Following Ref. 5, this interest is motivated by the following issues:

1. The DM's cognitive effort would be greatly reduced if the MOEA were able to identify the Region of Interest (RoI), the privileged zone of the Pareto frontier that best matches the DM's preferences. This is particularly important in problems with five or more objectives because i) the capacity of the human mind is restricted to handling a small amount of information at the same time⁶; and ii) the cardinal of a representative portion of the known Pareto frontier grows exponentially with the number of objective functions.
2. With an increasing number of objectives, almost all solutions in a current MOEA's population become non-dominated (in a local sense); so, the algorithm's selective pressure toward the true Pareto frontier may be very weak; the DM's preference information allows reinforcing the necessary selective pressure.

Most approaches for incorporating preferences in MOEAs assume a particular model of DM preferences. This model may be based on value functions (cf. Refs. 7, 8, 9), outranking relations (see Ref. 3), reference points (cf. Ref. 10), reservation points¹¹, desirability thresholds¹², decision rules¹³, and other ways of aggregating multi-criteria preferences. Interesting surveys of this topic are found in Refs. 14 and 15.

The models based on outranking relations exhibit some desirable features: i) the outranking approach can handle imprecise and qualitative information; ii) the preference relation derived from the outranking

information can be used as complement of dominance, increasing the selective pressure toward the RoI; iii) fuzzy outranking relations are more flexible than value functions, handling phenomena as intransitivity and incomparability.

An interesting outranking approach for incorporating preferences in MOEAs was proposed by Fernandez et al. in Ref. 3. Based on the index of credibility of the outranking, Fernandez et al. modeled the relational system of preferences proposed by Roy in Ref. 16. They derived a model for the strict, weak and K-preference. An original many-objective problem is mapped in a surrogate three-objective problem, as proposed in Ref. 3. The best compromise solution is a non-dominated solution of the surrogate problem.

The method of Fernandez et al. in Ref. 3 was successfully applied in solving portfolio problems with many objectives^{17,18} and project partial support¹⁹. In order to apply the method, the outranking model parameters (weights and thresholds required by the index of credibility of the outranking) must be elicited. Besides, some additional symmetric and asymmetric parameters (that are specific of the preference model proposed by Fernandez et al. in Ref. 3 are required too. Usually, the DMs reveal difficulties when they are inquired to assign values to parameters whose meanings are barely clear for them²⁰. On the other hand, indirect procedures, which compose the so-called preference-disaggregation analysis (*PDA*), use regression-like methods for inferring a set of parameters from a battery of decision examples²¹. According to Greco et al. in Ref. 22, *MCDA* approaches based on disaggregation paradigms are of increasing interest because they imply relatively less cognitive effort from the DM. The direct eliciting settlement has been criticized in Refs. 23 and 24. Marchant argues²³ that the only valid preferential input-information is that arising from DM's preferential judgments in pairwise comparisons.

The heavy cognitive effort required from the DM in order to make a direct parameter setting is probably the main objection against the approach of Ref. 3. As a consequence, a method for an indirect parameter elicitation would be welcome to concern many objective problems. However, as stated by Fernandez²⁰, in outranking-based models, inferring all the parameters

simultaneously requires solving a non-linear programming problem with non-convex constraints, which is usually difficult for mathematical programming techniques^{25,26}. According to Doumpos²¹, the relational form of these models and the veto conditions may make it impossible to infer the model parameters in real-size data sets by using mathematical programming. An alternative is the use of metaheuristics, which can find, in relatively small time, approximated solutions convenient for real purposes (see, e.g. Refs. 27 and 28).

The research work in this paper focuses on the development of an optimization approach for the PDA method. The approach is based on metaheuristics and used to infer the parameters of the preference model proposed by Fernandez³. According to the consulted scientific literature, our work is the only one so far that involves the considered outranking parameters. The resulting method was tested using different algorithms. The conducted experimental design evaluated the quality of the estimated parameters in both, its proximity to the real parameters, and its capacity to predict new preference relations. The remaining of this document is organized as follows. Section 2 defines the preference relations, and the surrogate model defined in Ref. 3. Section 3 shows the particular set of parameters studied, and the inference approach defined in this paper to estimate them. Section 4 shows the developed method to estimate parameters for the studied preference model, and presents the experimental design conducted in order to test the method. Finally, Section 5 presents some concluding remarks derived from our research.

2. The method of Fernandez et al. (2011) for finding the best compromise

Consider a *MOP* of the form

$$\begin{aligned} \text{Maximize } F &= (f_1(z), f_2(z), \dots, f_n(z)) \\ z &\in R_F \end{aligned} \quad (1)$$

in which z denotes a vector of decision variables and R_F is determined by a set of constraints.

In the following O denotes the image of R_F in the objective space mapped by the vector function F . An element $x \in O$ is a vector (x_1, \dots, x_n) , where x_i is the

i -th objective function value. $S(x, y)$ denotes the predicate “the *DM* considers that option x is at least as good as y ” defined on $O \times O$.

Roy described¹⁶ situations concerning this non-ideal behavior from real decision actors by using a relational system of preferences composed of several binary relations. The definitions from Roy¹⁶ are given below:

1. **Indifference:** It corresponds to the existence of clear and positive reasons that justifies equivalence between the two actions. Notation: x/y .
2. **Strict preference:** It corresponds to the existence of clear and positive reasons that justify significant preference in favor of one (identified) of the two actions. The statement x is strictly preferred to y is denoted by xPy . P is asymmetric and non-reflexive.
3. **Weak preference:** It corresponds to the existence of clear and positive reasons in favor of x over y , but that are not sufficient to justify strict preference. Indifference and strict preference cannot be distinguished appropriately. This is denoted by xQy . Q is asymmetric and non-reflexive.
4. **Incomparability:** None of the preceding situations predominates. That is, the absence of clear and positive reasons that justify any of the above relations. Notation: xRy . R is symmetric.
5. **K-preference:** It corresponds to the existence of clear and positive reasons that justify strict preference in favor of one (identified) of the two objects or incomparability between the two objects, but with no significant division established between the situations of strict preference and incomparability. Notation: xKy . K is asymmetric.
6. **Non-preference:** It corresponds to situations in which indifference and incomparability are both possible, without being able to differentiate between them. This is denoted by $x \sim y$. \sim is symmetric.

Fernandez et al. assumed³ that there is a method for assigning a degree of truth $\sigma(x, y)$ in $[0, 1]$ to the predicate $S(x, y)$; outranking methods such as ELECTRE-III^{29,30} and PROMETHEE³¹ may be used.

This work uses ELECTRE-III, as defined in Appendix A. Once $\sigma(x,y)$ has been calculated, it can be useful for modeling the crisp preference relations defined b for each pair (x, y) :

Definition 1: x is (λ, β) -strictly preferred to y if at least one of the following conditions holds y Roy (1996). Using real numbers $\lambda, \beta, \varepsilon$, ($\varepsilon < \beta < \lambda < 1$), the proposals in Refs. 3 and 17 define the following relations:

- i. x dominates y
- ii. $\sigma(x,y) \geq \lambda \wedge \sigma(y,x) < 0.5$
- iii. $\sigma(x,y) \geq \lambda \wedge (0.5 \leq \sigma(y,x) < \lambda) \wedge (\sigma(x,y) - \sigma(y,x)) \geq \beta$

Notation: $xP(\lambda, \beta)y$. The value λ is the outranking credibility threshold, greater than 0.5; β is an asymmetry parameter.

Definition 2: x and y are (λ, ε) -indifferent if $\sigma(x,y) \geq \lambda$, $\sigma(y,x) \geq \lambda$ and $|\sigma(x,y) - \sigma(y,x)| < \varepsilon$. Notation: $xI(\lambda, \varepsilon)y$. The value ε is a symmetry parameter.

Definition 3: x is $(\lambda, \beta, \varepsilon)$ -weakly preferred to y if the following conditions are satisfied.

- i. $\sigma(x,y) \geq \lambda \wedge \sigma(x,y) > \sigma(y,x)$
- ii. $x \cdot \text{not } P(\lambda, \beta)y$
- iii. $x \cdot \text{not } I(\lambda, \varepsilon)y$

Notation: $xQ(\lambda, \beta, \varepsilon)y$

Definition 4: x is (λ, ε) - K -preferred to y if the following conditions are satisfied:

- i. $0.5 \leq \sigma(x,y) < \lambda$
- ii. $\sigma(y,x) < 0.5$
- iii. $(\sigma(x,y) - \sigma(y,x)) > \varepsilon$

Notation: $xK(\lambda, \varepsilon)y$

Definition 5: The incomparability relation xRy is defined by $\sigma(x,y) < 0.5 \wedge \sigma(y,x) < 0.5$.

Definition 6: There is a $(\lambda, \beta, \varepsilon)$ -non-preference between x and y if the following conditions are satisfied.

- i. $\text{not. } xP(\lambda, \beta)y \wedge \text{not. } yP(\lambda, \beta)x$;
- ii. $\text{not. } xQ(\lambda, \beta, \varepsilon)y \wedge \text{not. } yQ(\lambda, \beta, \varepsilon)x$;
- iii. $\text{not. } xK(\lambda, \varepsilon)y \wedge \text{not. } yK(\lambda, \varepsilon)x$;
- iv. $\text{not. } xI(\lambda, \varepsilon)y$;
- v. $\text{not. } xRy$.

Notation: $x \sim (\lambda, \beta, \varepsilon)y$

Fernandez et al. used³ the above relations P, Q and K to define the best compromise solution to Problem 1 as follows:

- Given $x \in O$, the set of $(\lambda, \beta, \varepsilon)$ -strictly outranking solutions is defined as $(S_O)_x = \{y \in O \text{ such that } yP(\lambda, \beta)x\}$;
- The $(\lambda, \beta, \varepsilon)$ -non strictly outranked frontier is defined as $N_S = \{x \in O \text{ such that } \text{card}(S_O)_x = 0\}$;
- If x^* is the best compromise of Problem 1, then there is no $y \in O$ such that yPx^* . Then, if $P \subseteq P(\lambda, \beta)$, in order to be the best compromise x^* must belong to N_S (a necessary condition).

Once N_S has been identified, Fernandez et al. suggest³ to choose the best compromise considering two additional criteria: a) $\text{card}(W_{NS})_x = \text{card}\{y \in N_S \text{ such that } yQ(\lambda, \beta, \varepsilon)x \text{ or } yK(\lambda, \beta)x\}$; and, b) $\text{card}(F_{NS}) = \text{card}\{y \in N_S \text{ such that } F_n(y) > F_n(x)\}$, where F_n denotes the net flow score in N_S , that is $F_n(a) = \sum_{c \in N_S - \{a\}} [\sigma(a, c) - \sigma(c, a)]$. The best solution to Problem 1 is chosen from the non-dominated set of

$$\text{Minimize } (\text{card}(S_O)_x, \text{card}(W_{NS})_x, \text{card}(F_{NS})_x) \quad x \in O \quad (2)$$

with preemptive priority favoring $\text{card}(S_O)$.

3. A method to infer the model parameters by using preference relations

This section presents the inference approach used to estimate the parameter values of the fuzzy relational system.

3.1. Some assumptions

Assumption 1: The DM is willing to make judgments using P, Q, I, K, R and \sim on a reference set T of pairs of alternatives (x, y) .

Assumption 2: Given a pair of alternatives (x, y) the DM can identify one and only one of the following statements as more acceptable: $xPy, yPx, xIy, xQy, yQx, xKy, yKx, xRy, x \sim y$. So, each pair (x, y) in T is associated with a relation A of the set $\{P, I, Q, K, R, \sim, P^{-1}, Q^{-1}, K^{-1}\}$, where $(xA^{-1}y \Leftrightarrow yAx)$.

3.2. The model parameters

If credibility index $\sigma(x,y)$ is calculated as in ELECTRE III, the following parameters must be specified in the model of Fernandez et al. in Ref. 3:

- i) the vector of weights $w=\{w_1, w_2, \dots, w_n\}$;
- ii) the vector of indifference thresholds $q=\{q_1, q_2, \dots, q_n\}$;
- iii) the vector of preference thresholds $p=\{p_1, p_2, \dots, p_n\}$;
- iv) the vector of veto thresholds $u=\{u_1, u_2, \dots, u_n\}$;
- v) the vector of discordance thresholds (only when the simplification suggested by Mousseau and Dias is used³²) $v=\{v_1, v_2, \dots, v_n\}$;
- vi) λ , β , and ε .

Given the alternatives (x,y) , the value of the credibility index $\sigma(x,y)$ is a function of the parameters $\eta=\{w,q,p,u,v\}$, and will be denoted in the future as $\sigma(\eta,x,y)$. The function $\sigma(\eta,x,y)$, in combination with the remaining outranking (λ), symmetry (β), and asymmetry (ε) thresholds, determine the preference relations according to Definitions 1-6.

3.3. The inference approach

Let P_{fr} be the set of feasible parameter vectors $(\eta,\lambda,\beta,\varepsilon)$. 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 $(x,y) \in T$:

$$xPy \Leftrightarrow xP(\eta_0,\lambda_0,\beta_0)y \quad (3.a)$$

$$xIy \Leftrightarrow xI(\eta_0,\lambda_0,\varepsilon_0)y \quad (3.b)$$

$$xQy \Leftrightarrow xQ(\eta_0,\lambda_0,\beta_0,\varepsilon_0)y \quad (3.c)$$

$$xKy \Leftrightarrow xK(\eta_0,\lambda_0,\varepsilon_0)y \quad (3.d)$$

$$x \sim y \Leftrightarrow x \sim (\eta_0,\lambda_0,\beta_0,\varepsilon_0)y \quad (3.e)$$

$$xRy \Leftrightarrow xR(\eta_0)y \quad (3.f)$$

The best parameter setting should be the closest solution to the ideal one in the sense of certain acceptable metric. Following Refs. 3 and 20 we propose a metric based on the so-called *inconsistencies* with Eqs. 3. Shortly, an inconsistency arises when an equivalence in Eqs. 3 is not fulfilled. For example, given $(x,y) \in T$ and $(\eta,\lambda,\beta,\varepsilon) \in P_{fr}$ we call inconsistency with (3.a) the fact that x is strictly preferred to y by the DM and however *not* $(xP(\eta,\lambda,\beta)y)$, or vice versa, $xP(\eta,\lambda,\beta)y$ but x is not

strictly preferred to y by the DM. Let N_1, N_2, \dots, N_6 be the number of inconsistencies in T with Eqs. 3.a, 3.b, 3.c, 3.d, 3.e and 3.f, respectively.

Since the method by Fernandez et al. in Ref. 3 (see Problem 2) gives priority to find the non-strictly outranked set, N_I is far the most important measure. The information coming from Q and K is also considered in Problem 2, being thus more important than that provided by I , R , and \sim . So, we propose to find the best $(\eta,\lambda,\beta,\varepsilon)$ from the solution of the optimization problem:

$$\begin{aligned} &\text{Minimize } (E_1, E_2, E_3) \\ &(\eta, \lambda, \beta, \varepsilon) \in P_{fr} \end{aligned} \quad (4)$$

where P_{fr} is the feasible region of the parameter values, $E_1=N_1$, $E_2=N_3+N_4$, and $E_3=N_2+N_5+N_6$, with preemptive priority favoring E_1 over E_2 , and E_2 over E_3 .

4. Optimization approach to solve the inference approach

This section presents the proposed approach based in metaheuristic to estimate the parameter values of the model of preferences studied.

4.1. Some metaheuristic approaches

We propose here a method based on mono-objective optimization algorithms to solve Problem (4). The method exploits the preemptive priority established for the problem; it finds the best value for the first objective, then this value is used as a bound when the second objective is minimized. Finally, E_3 is minimized keeping the minimum values previously obtained for (E_1, E_2) .

Figure 1 depicts the general procedure followed by the approach. Each metaheuristic algorithm solves $P=(DM_{sim},T)$, i.e. an instance of Problem (4) formed by the parameter values that define the model of preferences for a simulated DM, denoted as DM_{sim} , and its reference set T , trying to minimize only the objective E_I . The best value $Best_{E_I}$ is used as bound in the following step of the method; in this step the same algorithm searches the minimization of objective E_2 while it validates that the solutions achieved yield a value on E_I equal or smaller than $Best_{E_I}$. Finally, in the last step, the algorithm now minimizes the third objective of Problem (4), i.e. E_3 , while keeping the

values for E_1 and E_2 smaller than $Best_{E_1}$ and $Best_{E_2}$, respectively. The best solution $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ will arise from the parameter vectors obtained from the last step of this procedure.

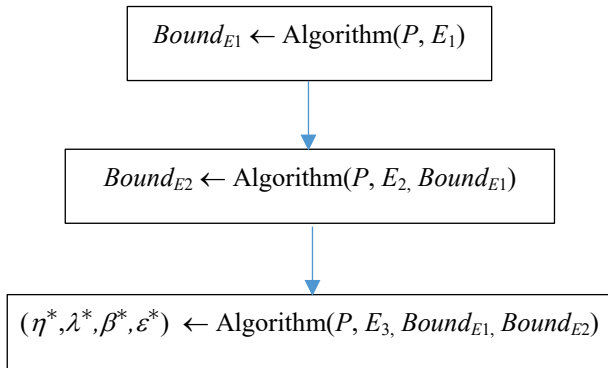


Figure 1. Strategy followed to solve Problem (4) using mono-objective algorithms.

The algorithms that were analyzed in this work are: a) Genetic Algorithm; b) Particle Swarm Optimization; c) Tabu Search; and, d) Simulated Annealing. These strategies have two main components in common: the problem encoding and the evaluation function. The evaluation function on each stage of the general procedure is related to a particular objective E_i of Problem (4), and is described in Section 3.3. The encoding, the algorithms, and the fine-tuning process of the algorithms' parameters are detailed in the remaining of this section.

4.1.1. Problem encoding

All the strategies developed in this work use the same encoding to solve Problem (4). This encoding is based on a vector ρ of real numbers which represents the parameter vector $(\eta, \lambda, \beta, \varepsilon)$ to be estimated. The size of ρ is $5n+3$, depending on the number n of criteria involved.

In order to provide an example of the proposed encoding, let's suppose that the decision process is made over actions composed by two criteria. Then, the vector ρ will be set up as in Figure 2.

0	1	3	4	5	6	7	8	9	10	11	12	13
w_1	w_2	q_1	q_2	p_1	p_2	u_1	u_2	v_1	v_2	λ	β	ε

Figure 2. Vector ρ representing a solution for the Problem (4).

4.1.2. Genetic Algorithm

The Genetic Algorithm metaheuristic (GA) is inspired by natural selection, it uses data structures that represent chromosomes, and it moves from one population of them to another new one through recombination, mutation, crossover, or selection operators, in order to preserve critical information (see Refs. 33 and 34).

The algorithm shown in Figure 3 is based on a GA paradigm, and it is used to solve Problem (4) under the lexicographic method previously described.

The chromosome of the GA is derived from the encoding in Section 4.1.1. The algorithm randomly creates an initial population of individuals P_0 . After that, and at each generation, it creates from the actual P_t a new offspring population Q_t using the chosen operators. Basically, in this phase the algorithm uses *Binary Tournament* to choose two parents, and breed two child using *Simulated Binary Crossover* (or *SBXCrossover*); finally, the pair of resulting offsprings is subject to the *Polynomial Mutation* process.

The next step in the algorithm is the combination of parents and offspring into a set R_t from which will be extracted the individuals with the best fitness; these will form the next generation of parents P_{t+1} , and their fitness is evaluated from the objectives E_i in Problem (4) taking into account the constraints. The number of individuals per generation is defined in the parameter *Population Size*.

Note that this implementation of the GA defines the stop criterion in terms of the maximum number of evaluations allowed; on this base the number of generations can be computed through the quotient of the *Evaluations* and the *Population Size*.

The GA will return the set of parameter vectors that yield the best fitness value for the objectives E_i satisfying the *Bounds*.


```

GeneticAlgorithm( $P, E_i, Bounds$ )
 $P_0 \leftarrow \text{CreateInitialPopulation}()$ 
while(  $e < \text{Evaluations}$  ) {
     $Q_t \leftarrow \text{CreateOffspring}(P_t, Sel, Cross, Mut)$ 
     $R_t \leftarrow P_t \cup Q_t$ 
     $P_{t+1} \leftarrow \text{BestFitness}(R_t, E_i, Bounds)$ 
     $t \leftarrow t + 1$ 
     $e \leftarrow e + \text{PopulationSize}$ 
}
return {  $(\rho, E_i(\rho)) \mid \text{Fitness}(\rho) = \text{Best}_{E_i, Bounds}$  }

```

a) Genetic Algorithm (GA)

```

LexicographicProcess ( )
 $(\rho_1, Bound_{E1}) \leftarrow \text{GA}(P, E_1, \{none\})$ 
 $(\rho_2, Bound_{E2}) \leftarrow \text{GA}(P, E_2, \{Bound_{E1}\})$ 
 $(\rho_3, Bound_{E3}) \leftarrow \text{GA}(P, E_3, \{Bound_{E1}, Bound_{E2}\})$ 

```

b) Lexicographic process with GA

Figure 3. Method for Problem (4) based on GA.

The key elements required in the implementation of the proposed GA are: a) for the polynomial mutation, its probability and index of distribution; b) for *SBXCrossover* also its probability and index of distribution; c) the population size; and d) the stop criterion (or *Evaluations*).

The parameters of the GA were subject of a fine-tuning process based on Covering Arrays (CA) (see Ref. 35), in order to identify the best configuration of them for the GA.

The fine-tuning process employs commonly used parameter values (in this case are the ones in Table 1a), and with the CA it constructs a set of valid configurations (see Table 1b); finally, using a random generated instance, it tests all the configurations to identify the one with the best performance, which in our case is shown in Table 1c. The best one is used to lead the main experimental design proposed in this paper.

Table 1. Parameters of GA.

a) Tested Parameter Values						
No.	Parameter	Possible Values				
		V_1	V_2	V_3		
A	Mut.	0.02	0.05	0.10		
B	Mut.	5	10	20		
C	Cross.	0.85	0.90	0.95		
D	Cross.	5	10	20		
E	Population	50	100	200		
F	Evaluations	100000	50000	25000		

b) Valid configurations using a CA(15;2,6,3) ³⁵						
Config.	Parameter Values					
	A	B	C	D	E	F
1	0.02	5	0.90	20	200	50000
2	0.02	10	0.90	5	100	100000
3	0.02	20	0.85	5	50	25000
4	0.05	5	0.95	5	50	100000
5	0.05	10	0.85	10	200	100000
6	0.05	20	0.90	20	100	25000
7	0.10	5	0.85	10	100	25000
8	0.10	10	0.95	20	50	50000
9	0.10	20	0.90	10	50	100000
10	0.02	20	0.95	10	200	50000
11	0.10	10	0.95	5	200	25000
12	0.05	10	0.85	20	100	50000
13	0.05	20	0.95	5	100	25000
14	0.02	20	0.90	20	100	100000
15	0.05	10	0.95	5	50	50000

c) Best Configuration, Config. 14		
No.	Parameter	Value
A	Mut. Probability	0.02
B	Mut. Distribution Index	20
C	Cross. Probability	0.90
D	Cross. Distribution Index	20
E	Population Size	100
F	Evaluations	100000

4.1.3. Particle Swarm Optimization

The Particle Swarm Optimization metaheuristic (PSO) is inspired in social models where physical movements, or cognitive experiences are adjusted to optimize environmental parameters of a given population³⁶. The adjustments of the movements are controlled in this algorithm mainly by the parameters ω , ϕ_p , and ϕ_g , which must be selected carefully to control the behavior and efficacy of the method.

The algorithm shown in Figure 4 is based on a PSO paradigm, and it is used to solve Problem (4) under the

lexicographic method previously described³⁷. The definition of particle used by the algorithm is as in Section 4.1.1. The number of dimensions of these particles is set to $5n+3$, where n is the number of criteria that define the actions. Firstly, the algorithm defines the initial position x_p , the best position b_p , and the velocity v_p of each particle, and obtain the best known position ρ . Note that the fitness of a position is determined by the objective E_i that is analyzed, and the fulfillment of the constraints (given by Bounds in the pseudocode below). After that, the algorithm updates the velocity v_p of the particle and the new value is added to its actual position x_p ; the velocity v_p is updated using the parameters (ω, ϕ_p, ϕ_g), the values r_p, r_g generated uniformly at random using $U(l_{p,d}, u_{p,d})$, the best position of the particle b_p , and the best known position of the swarm ρ . Finally, the best position b_p and the best swarm position ρ are updated accordingly to the new position x_p achieved by the particle and its fitness. The algorithm returns ρ .

PSO($P, E_i, Bounds$)

```

P0 ← CreateInitialPopulation()
for each particle  $p$  in  $P_0$ 
  for each dimension  $d$  in  $p$ 
     $x_{p,d} \leftarrow U(l_{p,d}, u_{p,d})$ 
     $b_{p,d} \leftarrow x_{p,d}$ 
     $v_{p,d} \leftarrow U(l_{p,d}, u_{p,d}) - l_{p,d}$ 
    if  $E_i(x_p) < E_i(\rho)$  and  $Bounds(x_p)$ 
       $\rho \leftarrow x_p$ 
while(  $e < Evaluations$  ) {
  for each particle  $p$  in  $P_t$ 
    for each dimension  $d$  in  $p$ 
       $r_p, r_g \leftarrow U(l_{p,d}, u_{p,d})$ 
       $v_{p,d} \leftarrow \omega v_{p,d} + \phi_p r_p (b_{p,d} - x_{p,d}) + \phi_g r_g (\rho_{p,d} - x_{p,d})$ 
    )
     $x_p \leftarrow v_p$ 
    if  $E_i(x_p) < E_i(b_p)$ 
       $b_p \leftarrow x_p$ 
    if  $E_i(x_p) < E_i(\rho)$ 
       $\rho \leftarrow x_p$ 
     $e \leftarrow e + SwarmSize$ 
}
return ( $\rho, E_i(\rho)$ )

```

a) Algorithm PSO

LexicographicProcess ()

```

( $\rho_1, Bound_{E1}$ ) ← PSO( $P, E_1, \{none\}$ )
( $\rho_2, Bound_{E2}$ ) ← PSO( $P, E_2, \{Bound_{E1}\}$ )
( $\rho_3, Bound_{E3}$ ) ← PSO( $P, E_3, \{Bound_{E1}, Bound_{E2}\}$ )

```

b) Lexicographic process with PSO

The parameters of the PSO implementation that require a previous adjustment of their values are ω, ϕ_p, ϕ_g , the swarm size, and the stop criterion (or *Evaluations*).

The adjustments in the PSO parameters followed the same CA based method described in Section 4.1.2. Table 2a presents the values considered per parameter; Table 2b shows the set of valid configurations tested; and, Table 2c contains the winner configuration through the method, using a random instance. The latter configuration was used to lead the main experimental design proposed in this paper.

Table 2. Parameters of PSO.

a) Tested Parameter Values					
No.	Parameter	Possible Values			
		V ₁	V ₂	V ₃	
A	ϕ_p	1.00	1.05	1.10	
B	ϕ_g	1.00	1.05	1.10	
C	ω	0.10	0.20	0.30	
D	Swarm Size	50	100	200	
E	Evaluations	100000	50000	25000	

b) Valid configurations using a CA(13;2,5,3) ³⁵					
Config.	Parameter Values				
	A	B	C	D	E
1	1.00	1.00	0.20	200	100000
2	1.00	1.05	0.20	50	50000
3	1.00	1.10	0.10	50	25000
4	1.05	1.00	0.30	50	25000
5	1.05	1.05	0.10	100	100000
6	1.05	1.10	0.20	200	50000
7	1.10	1.00	0.10	100	50000
8	1.10	1.05	0.30	200	25000
9	1.10	1.10	0.20	100	25000
10	1.00	1.10	0.30	100	100000
11	1.10	1.00	0.30	50	100000
12	1.05	1.00	0.10	200	50000
13	1.10	1.00	0.30	100	50000

c) Best Configuration, Config. 5		
No.	Parameter	Value
A	ϕ_p	1.05
B	ϕ_g	1.05
C	ω	0.10
D	Swarm Size	100
E	Evaluations	100000

Figure 4. Method for Problem (4) based on PSO.

4.1.4. Tabu Search

The Tabu Search metaheuristic (TS) was proposed by Glover and Laguna³⁸. The strategy uses a Tabu list as a *memory* that stores information of forbidden movements, so that the algorithm can avoid been stuck in a local optimal solution. The strategy uses the *expiration time* to remove movements from the Tabu list.

The algorithm shown in Figure 5 is based on a TS paradigm, and it is used to solve Problem (4) under the lexicographic method described in Section 4.1.

```

TabuSearch( $P, E_i, Bounds$ )
 $\rho_{act} \leftarrow \text{CreateRandomInitialSolution}()$ 
 $\rho^* \leftarrow \rho_{act}$ 
while(  $e < \text{Evaluations}$  ) {
     $\rho_{best} \leftarrow \rho_{act}$ 
    for  $p$  in  $(\eta, \lambda, \beta, \varepsilon)$ 
         $\rho_{new} \leftarrow N(\rho_{act}, p)$ 
        if  $Bounds(\rho_{new})$ 
            if  $\text{Fitness}(\rho_{new}) < \text{Fitness}(\rho_{best})$ 
                if NotInTabuList( $p$ )
                     $\rho_{best} \leftarrow \rho_{new}$ 
     $\rho_{act} \leftarrow \rho_{best}$ 
    if  $\text{Fitness}(\rho_{best}) < \text{Fitness}(\rho^*)$ 
         $\rho^* \leftarrow \rho_{best}$ 
    UpdateTabuList( $p$ )
     $e \leftarrow e + V$ 
}
return ( $\rho^*, E_i(\rho^*)$ )

```

a) Algorithm TS

```

LexicographicProcess ( )
( $\rho_1, Bound_{E1}$ )  $\leftarrow \text{TS}(P, E_1, \{none\})$ 
( $\rho_2, Bound_{E2}$ )  $\leftarrow \text{TS}(P, E_2, \{Bound_{E1}\})$ 
( $\rho_3, Bound_{E3}$ )  $\leftarrow \text{TS}(P, E_3, \{Bound_{E1}, Bound_{E2}\})$ 

```

b) Lexicographic process with TS

Figure 5. Method for Problem (4) based on TS.

The representation of a solution in the TS of Figure 5 follows the encoding in Section 4.1.1. The neighborhood function used (denoted by N) randomly varies a value of a single chosen parameter in the bounds allowed by the problem; the local search tests every of the $V=5n+3$ parameters being estimated. The movements in the Tabu list correspond to parameters that are modified to create a new solution from the actual solution ρ_{act} . The *Fitness* of a solution is

evaluated using E_i , and the *Bounds* according to the phase being solved by the method. Firstly, the algorithm tries the change of every parameter of ρ_{act} to create a new solution ρ_{new} , and selects the best one ρ_{best} , which has the best fitness and does not belong to the tabu list. After that, the solution ρ_{best} is used to update ρ_{act} and the global best solution ρ^* . Finally, the tabu list is updated so that the movement p used to produce ρ^* is stored, and the expiration time of the remaining is checked. The algorithm returns the global best solution ρ^* found.

The CA based fine-tuning process, described in Section 4.1.2, applied to the proposed TS implementation works on the stop criterion (i.e. the number of *Evaluations*), and on the Tabu list size, and the expiration time used in the *UpdateTabuList* method. Table 3a presents the values considered per parameter; Table 3b shows the set of valid configurations tested; and, Table 3c contains the winner configuration through the method, using a random instance. The latter configuration was used to lead the main experimental design proposed in this paper.

Table 3. Parameters of TS.

a) Tested Parameter Values				
No.	Parameter	Possible Values		
		V_1	V_2	V_3
A	Expiration	$(5n+3)$	$2(5n+3)$	$5(5n+3)$
B	Tabu List	$(5n+3)$	$2(5n+3)$	$5(5n+3)$
C	Evaluations	25000	50000	100000

b) Valid configurations using a CA(10;2,3,3) ³⁵			
Config.	Parameter Values		
	A	B	C
1	$(5n+3)$	$2(5n+3)$	25000
2	$(5n+3)$	$2(5n+3)$	50000
3	$(5n+3)$	$(5n+3)$	100000
4	$2(5n+3)$	$5(5n+3)$	25000
5	$2(5n+3)$	$(5n+3)$	50000
6	$2(5n+3)$	$2(5n+3)$	100000
7	$5(5n+3)$	$(5n+3)$	25000
8	$5(5n+3)$	$5(5n+3)$	50000
9	$5(5n+3)$	$2(5n+3)$	100000
10	$(5n+3)$	$5(5n+3)$	100000

c) Best Configuration, Config. 6		
No.	Parameter	Value
A	Expiration Time	$2(5n+3)$
B	Tabu List Size	$2(5n+3)$
C	Evaluations	100000

4.1.5. Simulated Annealing

The Simulated Annealing approach (SA)³⁹ is based on the analogy between the simulation of the annealing of solids and the problem of solving large combinatorial optimization problems. In condensed matter physics, annealing denotes a physical process in which: firstly, a solid is heated up by increasing its temperature to a maximum value; and secondly the solid is cooled so that its particles arrange in the low energy ground state of a corresponding lattice. This low energy state is achieved once that the temperature of the solid in its liquid phase is slowly lowered. The analogy between the optimization of a problem and the process already described can be seen in this way: the solid becomes the problem, the energy is represented by the objective function being optimized, and the low energy state is achieved through a cooling process defined by an initial and final temperatures and a cooling factor. The main features of the SA are an initialization function; a neighborhood function that randomly changes the value of a parameter chosen at random, and a cooling schedule.

```

SimulatedAnnealing( $P, E_i, Bounds$ )
 $\rho_{act} \leftarrow \text{CreateRandomInitialSolution}()$ 
 $\rho^* \leftarrow \rho_{act}$ 
while(  $e < \text{Evaluations}$  and  $T_0 > T_f$  ){
  for  $i=0$  to  $L$ 
     $\rho_{new} \leftarrow N(\rho_{act})$ 
     $\Delta E \leftarrow C(\rho_{act}) - C(\rho_{new})$ 
    if  $\Delta E > 0 \parallel \exp(-\Delta E) > U(0,1)$ 
       $\rho_{act} \leftarrow \rho_{new}$ 
      if  $C(\rho_{new}) < C(\rho^*)$ 
         $\rho^* \leftarrow \rho_{new}$ 
     $T_0 \leftarrow \alpha T_0$ 
     $e \leftarrow e + L$ 
}
return ( $\rho^*, E_i(\rho^*)$ )

```

a) Algorithm SA

```

LexicographicProcess ( )
( $\rho_1, Bound_{E1}$ )  $\leftarrow$  SA( $P, E_1, \{none\}$ )
( $\rho_2, Bound_{E2}$ )  $\leftarrow$  SA( $P, E_2, \{Bound_{E1}\}$ )
( $\rho_3, Bound_{E3}$ )  $\leftarrow$  SA( $P, E_3, \{Bound_{E1}, Bound_{E2}\}$ )

```

b) Lexicographic process with SA

The algorithm shown in Figure 6 is based on a SA paradigm, and it is used to solve Problem (4) under the lexicographic method described in Section 4.1. Solutions are represented by using the encoding of Section 4.1.1, and the first one is constructed at random, and it is assigned to ρ_{act} . Then, in each iteration the algorithm tries to improve ρ_{act} and create ρ_{new} using its neighborhood function N , which arbitrarily changes the value of a parameter chosen at random. Using the Boltzmann criterion and the fitness C of ρ_{new} , it is decided whether or not ρ_{new} must substitute ρ_{act} . The fitness C is computed as a linear combination of E_i and $Bounds$ with priority in $Bounds$. The algorithm returns the global best solution ρ^* found.

Table 4. Parameters of SA.

a) Tested Parameter Values				
No.	Parameter	Possible Values		
		V_1	V_2	V_3
A	L	50	100	200
B	α	0.85	0.95	0.98
C	T_0	1	10	100
D	T_f	1×10^{-3}	1×10^{-6}	1×10^{-9}
E	Evaluations	25000	50000	100000

b) Valid configurations using a CA(13;2,5,3)³⁵

Parameter Values					
Config.	A	B	C	D	E
1	100	0.98	100	1×10^{-6}	100000
2	100	0.85	10	1×10^{-6}	25000
3	50	0.85	1	1×10^{-9}	25000
4	50	0.85	100	1×10^{-3}	50000
5	200	0.95	1	1×10^{-6}	50000
6	100	0.98	10	1×10^{-9}	50000
7	100	0.95	1	1×10^{-3}	100000
8	50	0.98	100	1×10^{-6}	100000
9	50	0.95	10	1×10^{-9}	100000
10	200	0.95	100	1×10^{-9}	25000
11	200	0.85	10	1×10^{-6}	100000
12	100	0.98	1	1×10^{-9}	25000
13	200	0.98	10	1×10^{-3}	25000

c) Best Configuration, Config. 1

No.	Parameter	Value
A	Markov Chain Length, L	100
B	Cooling Factor, α	0.98
C	Initial Temperature, T_0	100
D	Final Temperature, T_f	0.000001
E	Evaluations	100000

Figure 6. Method for Problem (4) based on SA.

The parameters of the SA implementation that required a previous adjustment of their values are: the Markov Chain length L , the cooling factor α , the initial temperature T_0 , the final temperature T_f , and the stop criterion (as the number of *Evaluations*). The fine-tuning process followed is the one presented in Section 4.1.2. Table 3a presents the values considered per parameter; Table 3b shows the set of valid configurations tested; and, Table 3c contains the winner configuration through the method, using a random instance. The latter configuration was used to lead the main experimental design proposed in this paper.

4.2. Experimental Design

This section details the experiment conducted to test the performance of the PDA related to the fuzzy outranking relational system proposed in Refs. 3 and 17. The description is divided into two parts. The first part describes the generation of instances to analyze the Problem (4) formulated to infer model parameters. The second part shows the validation process followed to measure the quality of the estimated parameter values, mainly in its capacity for properly predicting preferences in new sets of instances.

4.2.1 Generating instances

In order to test the model, it was necessary to simulate DM's responses, and to generate a set of instances as cases of study. For this purpose, we simulate the DM's preferences through the random generation of the parameter vector $\eta = \{w, q, p, u, v\}$, for a given number of criteria, and its combination with one of the considered tuples from $(\lambda, \beta, \varepsilon) = \{(0.51, 0.15, 0.07), (0.67, 0.15, 0.07), (0.70, 0.20, 0.10), (0.75, 0.20, 0.10)\}$. Then, the generated set $DM_{sim-parameters} = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$ is considered as the "true" preference parameters of the simulated DM that should be approximated by our inference approach.

Each instance is characterized by: i) a reference set T formed by a finite number of alternatives; and, ii) a set of statements made over pairs $(x, y) \in T \times T$, which are given by the DM as its preference judgments. The reference set T is generated using a random function; the number of alternatives, the number of criteria describing each action, and the range of criterion values are defined in advanced for T . In addition, for a particular action, the value of each criterion is generated

at random in the specified range. The set $A = \{P, I, Q, K, R, \sim, P^{-1}, Q^{-1}, K^{-1}\}$ is the set of preference statements that the real DM establishes on $(x, y) \in T \times T$ according to Assumption 2. In order to simulate the DM's responses, we assume that $x A_i y$ if $x A_i(\eta_0, \lambda_0, \beta_0, \varepsilon_0) y$ where $A_i \in A$ and $A_i(\eta_0, \lambda_0, \beta_0, \varepsilon_0)$ is one of the relations given by Definition 1. That is, $x A_i(\eta_0, \lambda_0, \beta_0, \varepsilon_0) y$ is considered as the response from the DM when (s)he is questioned about his/her preference on a pair (x, y) .

Given that the work presented in this document searches the identification of algorithms that properly characterizes to a DM_{sim} , it proposes an analysis of their performance on two sets of instances which differ in the number of involved actions. Let's point out that the use of a different amount of alternatives implies a greater involvement of a DM, reason why these should be taken as small as possible. Due to this reason, the first set (denoted as S_{10}) contains 40 instances where each has a reference set composed by 10 alternatives, and 45 preference statements established using a particular set of preference parameter $DM_{sim-parameters} = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$. The second set, referred as S_{20} , is formed by 40 instances each composed by 20 different alternatives, and 190 preference statements established using the same set of preference parameters $DM_{sim-parameters} = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$ used in the set S_{10} . The change in the number of alternatives from S_{10} to S_{20} allows to study the impact of the number of actions taken into account on the performance of the algorithms.

Each alternative of the sets S_{10} and S_{20} is constituted by 10 objectives in this work; and the universe from which these alternatives are formed considers values in the range $[1, 10]$ per objective. Hence, an alternative is generated by randomly assigning a value to each of its 10 objectives.

4.2.2 Validation

In order to validate properly the performance of the considered algorithms GA, PSO, TS, and SA, it is proposed to evaluate and compare the prediction capacity of the best estimation $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ obtained by each them. That prediction capacity can be measured by computing the error achieved on a new set of instances, called S_{100} . For each setting $DM_{sim-parameters} = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$ used in generating S_{10} , a validation set V with 100

objects generated at random is created. Each pair (x, y) of $V \times V$ is associated with preference statements $xA_i(\eta_0, \lambda_0, \beta_0, \varepsilon_0)y$ and $xA_j(\eta^*, \lambda^*, \beta^*, \varepsilon^*)y$. Then, in order to evaluate the error existing between a solution to Problem (4) $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ and $(\eta_0, \lambda_0, \beta_0, \varepsilon_0)$, we propose the use of the indicator defined as in Eq. (5).

$$I_P = \sum_{(x,y) \in T \times T} f_1(x, y),$$

where,

$$f_1(x, y) = \begin{cases} 1, & \text{if } xP(\eta_0, \lambda_0, \beta_0, \varepsilon_0)y \Leftrightarrow xP(\eta^*, \lambda^*, \beta^*, \varepsilon^*)y \text{ is FALSE} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Since the objectives in Problem (4) obey a lexicographical order, I_P is defined according to the objective E_1 , which counts inconsistencies derived from the equivalence $xP(\eta_0, \lambda_0, \beta_0, \varepsilon_0)y \Leftrightarrow xP(\eta^*, \lambda^*, \beta^*, \varepsilon^*)y$, for all $(x, y) \in V \times V$. Whenever a tie is produced using the indicator I_P , the objectives E_2 and E_3 of Problem (4) are used to break the tie. Note that the indicator I_P is defined as a way of measuring the capacity of prediction, in new instances, of the best estimation $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ obtained for the parameter values.

The new set S_{100} has 40 instances, each composed by 100 different alternatives, and 4450 preference statements. The best obtained estimations $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ using the set S_{10} were used to predict the preference relations given in S_{100} , and the resulting preferences were compared against the expected ones, i.e. $DM_{sim-parameters} = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$. The previous process is repeated using the set S_{20} .

Finally, the validation process concludes with a summary of the errors obtained per instance and algorithm, in contrast with the $DM_{sim-parameters} = (\eta_0, \lambda_0, \beta_0, \varepsilon_0)$.

4.3. Results

The results in this section are derived from solving Problem (4), using the proposed method of Section 4.1, under the experimental design presented in Section 4.2. The conducted experiment involves the prediction of the preference statements in a new set of instances S_{100} . It concludes with the results derived from the prediction

capacity of the estimated parameter values over the new set of instances. During this experiment, some constraints used in the literature, as in Ref. 20, were considered over the values of the parameters; these constraints are depicted as follows:

- $w_i > 0$;
- $\lambda > 0.51$;
- $0 \leq q_i \leq p_i < u_i < v_i (i = 1, \dots, n)$;
- $|(v_j + p_j)/2 - u_j| \leq 0.1(v_j - p_j)$;

Also, this work considers that the DM can provide lower and upper bounds on the values of the parameters. Based on this idea, we include a constraint that establishes the bounds for the parameters values in $\pm 30\%$ of the simulated values.

The content of this section is organized into three parts. The Section 4.3.1 presents a summary of the estimated parameter values obtained using the proposed method, contrasting them with the expected real parameter values. The Section 4.3.2 shows the details of the results concerning the use of the centroids obtained from S_{10} . Finally, the Section 4.3.3 details of the results concerning the use of the centroids obtained from S_{20} .

4.3.1 Estimated parameter values

Given that the algorithms used are nondeterministic, we obtain diverse solutions in the parameter space that hold optimal values. So, it is necessary a strategy that allows an appropriate handling of that diversity, choosing a representative solution.

For this purpose, the centroids $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ are taken as the best estimated values that are returned by the method as a solution for a given instance of Problem (4). In order to obtain a centroid, the method solves 30 times the instance. After that, the sets of solutions reported by the strategy used (i.e. the algorithms GA, PSO, TS, or SA) are put together to form the whole set of solutions.

Finally, the estimated parameter values $(\eta_i, \lambda_i, \beta_i, \varepsilon_i)$ of each solution i , from the whole set of solutions, are summed up individually, to obtain its average. The resulting value of each parameter involved in this process constitutes what is called the centroid.

Table 5. Comparison of average estimated parameter values of each algorithm against the average parameter values of a DM_{sim} per set of instances S_{10} and S_{20} .

	Set S_{10}										Set S_{20}									
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
DM_{sim}	0.09	0.12	0.10	0.08	0.09	0.11	0.09	0.09	0.10	0.10	0.09	0.12	0.10	0.08	0.09	0.11	0.09	0.09	0.10	0.10
GA	0.10	0.12	0.10	0.09	0.10	0.11	0.09	0.09	0.10	0.10	0.09	0.12	0.11	0.09	0.10	0.11	0.09	0.09	0.10	0.10
PSO	0.10	0.12	0.11	0.09	0.10	0.11	0.09	0.09	0.10	0.09	0.10	0.13	0.11	0.09	0.10	0.11	0.09	0.09	0.10	0.09
SA	0.10	0.12	0.11	0.08	0.10	0.11	0.09	0.10	0.10	0.10	0.10	0.12	0.10	0.09	0.10	0.11	0.09	0.10	0.10	0.10
TS	0.10	0.12	0.10	0.09	0.10	0.11	0.09	0.10	0.10	0.10	0.10	0.12	0.10	0.09	0.10	0.11	0.09	0.10	0.10	0.10

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}
DM_{sim}	0.15	0.17	0.15	0.13	0.15	0.13	0.13	0.16	0.13	0.12	0.15	0.17	0.15	0.13	0.15	0.13	0.13	0.16	0.13	0.12
GA	0.16	0.17	0.15	0.14	0.16	0.12	0.13	0.17	0.13	0.12	0.16	0.17	0.15	0.13	0.16	0.13	0.13	0.16	0.13	0.13
PSO	0.16	0.17	0.16	0.14	0.17	0.14	0.14	0.16	0.13	0.13	0.16	0.18	0.17	0.13	0.16	0.13	0.13	0.17	0.14	0.12
SA	0.17	0.19	0.17	0.16	0.18	0.15	0.14	0.18	0.14	0.14	0.18	0.17	0.16	0.14	0.17	0.13	0.14	0.18	0.14	0.13
TS	0.17	0.18	0.17	0.14	0.17	0.14	0.14	0.18	0.14	0.13	0.19	0.19	0.19	0.16	0.18	0.15	0.16	0.19	0.15	0.14

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
DM_{sim}	0.70	0.65	0.62	0.66	0.71	0.61	0.66	0.73	0.74	0.70	0.70	0.65	0.62	0.66	0.71	0.61	0.66	0.73	0.74	0.70
GA	0.65	0.63	0.53	0.60	0.67	0.54	0.57	0.68	0.68	0.62	0.62	0.68	0.56	0.62	0.65	0.53	0.55	0.68	0.67	0.65
PSO	0.71	0.65	0.61	0.65	0.70	0.62	0.64	0.75	0.71	0.71	0.69	0.65	0.64	0.65	0.71	0.61	0.67	0.71	0.74	0.71
SA	0.82	0.72	0.69	0.71	0.79	0.63	0.72	0.84	0.82	0.72	0.75	0.67	0.67	0.71	0.74	0.63	0.68	0.78	0.81	0.76
TS	0.78	0.70	0.68	0.74	0.80	0.68	0.74	0.82	0.82	0.79	0.82	0.77	0.70	0.74	0.79	0.68	0.75	0.84	0.82	0.81

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
DM_{sim}	1.66	1.89	1.67	1.63	1.56	1.65	1.52	1.69	1.55	1.76	1.66	1.89	1.67	1.63	1.56	1.65	1.52	1.69	1.55	1.76
GA	1.92	1.99	2.01	1.88	1.87	1.99	1.92	1.94	1.79	1.99	1.84	2.08	1.97	1.87	1.84	1.94	1.90	1.91	1.77	2.04
PSO	1.82	2.09	1.93	1.83	1.77	1.91	1.76	1.86	1.75	2.00	1.85	2.01	1.92	1.80	1.81	1.91	1.80	1.85	1.73	1.99
SA	2.31	2.22	2.61	2.23	2.33	2.39	2.56	2.24	2.10	2.45	2.10	2.11	2.34	2.13	2.11	2.31	2.28	2.07	2.08	2.41
TS	2.18	2.11	2.38	2.15	2.16	2.37	2.44	2.16	2.03	2.39	2.10	2.18	2.31	2.12	2.10	2.25	2.26	2.13	1.93	2.30

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
DM_{sim}	3.98	3.67	4.69	3.97	4.13	4.67	4.75	3.86	3.66	4.41	3.98	3.67	4.69	3.97	4.13	4.67	4.75	3.86	3.66	4.41
GA	3.28	3.41	3.75	3.25	3.25	3.59	3.53	3.30	2.95	3.56	3.22	3.48	3.66	3.22	3.21	3.55	3.54	3.21	2.97	3.69
PSO	3.47	3.65	3.93	3.49	3.53	3.92	3.83	3.41	3.24	3.92	3.65	3.53	3.98	3.55	3.54	4.00	3.82	3.44	3.25	3.84
SA	4.14	3.98	4.94	4.09	4.27	4.53	4.86	3.95	3.67	4.54	3.69	3.72	4.29	3.80	3.79	4.29	4.22	3.52	3.73	4.35
TS	3.87	3.68	4.43	3.88	3.90	4.42	4.64	3.79	3.57	4.28	3.56	3.74	4.27	3.71	3.69	4.06	4.09	3.64	3.25	4.04

	λ	β	ε	λ	β	ε
DM_{sim}	0.66	0.18	0.09	0.66	0.18	0.09
GA	0.69	0.18	0.09	0.68	0.17	0.09
PSO	0.66	0.18	0.09	0.67	0.18	0.09
SA	0.74	0.20	0.09	0.74	0.19	0.09
TS	0.70	0.19	0.09	0.72	0.21	0.10

Table 6. Estimated parameter values for instances I_{39} of set S_{10} and I_{38} of set S_{20} .

Instance 39 of set S_{10}										
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
DM_{sim}	0.12	0.12	0.12	0.09	0.10	0.09	0.09	0.08	0.09	0.06
GA	0.12	0.11	0.13	0.11	0.11	0.11	0.11	0.06	0.08	0.06
PSO	0.15	0.16	0.12	0.09	0.09	0.09	0.11	0.09	0.11	0.01
SA	0.12	0.13	0.13	0.09	0.10	0.10	0.12	0.08	0.09	0.05
TS	0.12	0.13	0.11	0.08	0.11	0.09	0.10	0.09	0.09	0.08

Instance 38 of set S_{20}										
	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
DM_{sim}	0.07	0.11	0.11	0.07	0.08	0.09	0.07	0.10	0.11	0.15
GA	0.05	0.10	0.08	0.08	0.11	0.10	0.08	0.10	0.11	0.19
PSO	0.09	0.13	0.11	0.08	0.10	0.12	0.09	0.08	0.13	0.08
SA	0.08	0.13	0.08	0.07	0.07	0.10	0.08	0.11	0.12	0.17
TS	0.07	0.13	0.13	0.08	0.07	0.11	0.07	0.10	0.09	0.16

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}
DM_{sim}	0.19	0.18	0.13	0.10	0.12	0.10	0.13	0.18	0.11	0.10
GA	0.19	0.20	0.17	0.12	0.10	0.12	0.15	0.23	0.08	0.08
PSO	0.16	0.21	0.13	0.12	0.13	0.12	0.10	0.17	0.11	0.10
SA	0.23	0.23	0.17	0.14	0.16	0.11	0.17	0.24	0.12	0.11
TS	0.21	0.18	0.14	0.13	0.13	0.12	0.12	0.20	0.14	0.11

	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	q_9	q_{10}
DM_{sim}	0.14	0.16	0.18	0.12	0.18	0.12	0.12	0.16	0.11	0.16
GA	0.15	0.19	0.17	0.09	0.17	0.09	0.13	0.18	0.11	0.11
PSO	0.13	0.21	0.16	0.13	0.18	0.11	0.13	0.17	0.09	0.18
SA	0.16	0.22	0.23	0.15	0.21	0.12	0.12	0.20	0.13	0.20
TS	0.16	0.20	0.23	0.15	0.15	0.09	0.15	0.15	0.15	0.20

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
DM_{sim}	0.76	0.87	0.56	0.74	0.55	0.49	0.66	0.83	0.62	0.75
GA	0.96	0.74	0.52	0.56	0.55	0.42	0.47	0.79	0.52	0.54
PSO	0.74	0.97	0.57	0.85	0.54	0.50	0.76	0.93	0.51	0.64
SA	0.62	0.67	0.50	0.79	0.69	0.42	0.80	1.08	0.60	0.93
TS	0.93	0.95	0.60	0.92	0.58	0.61	0.71	0.90	0.67	0.80

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
DM_{sim}	0.88	0.50	0.56	0.68	0.86	0.69	0.55	0.49	0.81	0.68
GA	0.62	0.47	0.40	0.53	0.68	0.74	0.39	0.55	0.70	0.69
PSO	1.00	0.43	0.73	0.66	0.66	0.48	0.50	0.61	0.81	0.53
SA	0.98	0.47	0.52	0.87	1.01	0.85	0.69	0.39	1.05	0.84
TS	1.11	0.60	0.71	0.88	1.13	0.89	0.64	0.63	1.04	0.52

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
DM_{sim}	1.47	1.71	1.59	1.25	1.58	1.63	1.28	1.64	1.26	2.03
GA	1.61	2.06	2	1.61	2.02	1.65	1.67	1.85	1.59	2.11
PSO	1.78	2.01	2.02	1.35	1.88	1.9	1.67	1.49	1.62	1.88
SA	2.22	2.09	2.77	2.6	2.72	2.12	1.93	1.89	2.13	2.34
TS	2.13	2.06	2.32	2.35	2.5	2.05	2.24	1.87	1.85	2.43

	u_1	u_2	u_3	u_4	u_5	u_6	u_7	u_8	u_9	u_{10}
DM_{sim}	1.63	2.00	1.39	1.69	1.26	1.74	1.58	1.44	1.96	2.00
GA	1.94	2.15	1.81	1.81	1.47	2.00	1.92	1.61	1.97	1.94
PSO	2.07	2.38	1.80	1.89	1.54	2.03	1.83	1.62	2.55	2.34
SA	3.15	2.04	2.90	2.72	1.80	2.29	2.55	1.87	2.43	2.66
TS	2.88	2.58	2.30	2.55	1.95	2.64	2.06	1.88	1.90	2.60

	v_1	v_2	v_3	v_4	v_5	v_6	v_7	v_8	v_9	v_{10}
DM_{sim}	3.53	3.12	5.29	3.95	5.15	3.63	4.91	3.04	3.28	3.88
GA	2.55	3.6	3.84	2.88	3.87	3.03	3.44	3.15	2.97	3.85
PSO	3.41	3.45	3.97	2.6	3.6	4.64	3.41	2.27	2.88	3.56
SA	4.23	3.98	5.65	4.99	5.34	4.29	3.49	2.47	4.12	3.32
TS	3.77	3.13	4.57	4.3	4.98	3.5	4.26	2.92	3.39	4.34

	λ	β	ϵ
DM_{sim}	0.70	0.20	0.10
GA	0.71	0.17	0.11
PSO	0.80	0.19	0.10
SA	0.74	0.21	0.13
TS	0.76	0.21	0.11

	λ	β	ϵ
DM_{sim}	0.67	0.15	0.07
GA	0.67	0.15	0.07
PSO	0.62	0.14	0.06
SA	0.78	0.14	0.08
TS	0.85	0.19	0.07

For space limitation we only show in Table 5 the results of two instances. This table compares the centroids obtained by the algorithms GA, PSO, TS, and SA, against $DM_{sim-parameters}=(\eta_0, \lambda_0, \beta_0, \varepsilon_0)$, the simulated parameter values. The average percentage of deviation in the instance of S_{10} is 16% for GA, PSO and TS, and 23% for SA. Besides, these percentages are 16%, 17%, 23% and 22% respectively in the instance of S_{20} . These performances are similar in the other instances.

Table 6 shows the average of the estimated parameter values for all the instances in the sets S_{10} and S_{20} . This table provides the average of the simulated parameter values, $DM_{sim-parameters}=(\eta_0, \lambda_0, \beta_0, \varepsilon_0)$, and the average of the centroids $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ obtained using the algorithms GA, PSO, TS, and SA.

4.3.2 Performance of prediction of centroids from S_{10} over the new set S_{100}

In this part of the experiment, the centroids $(\eta^*, \lambda^*, \beta^*, \varepsilon^*)$ were computed under the method proposed in Section 4.1, and having as input the set of instances S_{10} . The centroids are used to predict the preference statements $x A(\eta_0, \lambda_0, \beta_0, \varepsilon_0) y$ and $y A(\eta_0, \lambda_0, \beta_0, \varepsilon_0) x$ for every pair $(x, y) \in V \times V$, where V is the test set formed in each instance of S_{100} . The maximum number of inconsistencies that can be found in an instance is 9900, because each of the 4450 judgments of the DM_{sim} is evaluated as $x A y$ and $y A x$.

Table 7 summarizes the results from the validation process described in Section 4.2.3. An instance in this table is represented by a row, and a set of columns named after one of the tuples $(\lambda, \beta, \varepsilon) = \{(0.51, 0.15, 0.07), (0.67, 0.15, 0.07), (0.70, 0.20, 0.10), (0.75, 0.20, 0.10)\}$. Each instance has associated four cells in the table, which contain the value of I_P , i.e. the number of inconsistencies obtained by the algorithms GA, PSO, TS, and SA. Note that the only preference relation under study was the strict preference P , and it was enough to rank the algorithms properly. These results were statistically validated. Firstly, using the Friedman test with a significance level of 0.05 it was found that significant differences exist among the algorithms;

secondly, the Finner test indicated the superiority of GA over SA and TS.

Table 7. Summary of the indicator I_P from results derived from set S_{100} using centroids derived from set S_{10} .

10 Actions								
	(0.51,0.15,0.07)				(0.67,0.15,0.07)			
	GA	PSO	SA	TS	GA	PSO	SA	TS
1	86	71	152	111	43	69	88	89
2	114	119	185	119	48	72	151	155
3	71	76	123	162	36	80	134	60
4	95	78	186	110	71	88	138	79
5	91	111	240	110	100	87	157	147
6	114	72	160	220	48	69	141	99
7	86	98	208	138	28	102	87	143
8	55	88	193	83	66	164	153	93
9	97	44	72	60	31	27	374	70
10	90	52	182	80	67	190	134	158
	899	809	1701	1193	538	948	1557	1093

a) Inconsistencies of part 1 of instances with 10 actions.

	(0.70,0.20,0.10)				(0.75,0.20,0.10)			
	GA	PSO	SA	TS	GA	PSO	SA	TS
1	101	58	168	72	24	113	108	81
2	52	189	195	142	83	127	269	124
3	96	32	89	169	59	53	68	60
4	62	29	326	73	80	85	180	146
5	73	86	121	77	68	46	82	104
6	41	62	152	102	90	58	202	91
7	22	21	114	38	60	47	107	56
8	34	93	94	100	27	71	70	54
9	41	71	222	89	73	34	256	37
10	32	47	105	75	131	53	216	98
	554	688	1586	937	695	687	1558	851

b) Inconsistencies of part 2 of instances with 10 actions.

Table 8 shows the rank of the algorithms GA, PSO, SA, TS, according to the value of indicator I_P shown in Table 7, which evaluates the most important preference relation for this study. Each row of Table 8 is combined with a tuple from $(\lambda, \beta, \varepsilon)$, i.e. a group of four columns with such title, to represent an instance of the set S_{10} . The algorithms are ranked per instance, the one with the

minimum number of inconsistencies (or incorrect judgments) is ranked 1, the next one 2, and so on.

Table 8. Ranking of the performance of the algorithms per instance, when the centroids obtained from S_{10} are validated using S_{100} .

10 Actions									
	(0.51,0.15,0.07)				(0.67,0.15,0.07)				
	GA	PSO	SA	TS	GA	PSO	SA	TS	
1	2.0	1.0	4.0	3.0	1.0	2.0	3.0	4.0	
2	1.0	2.0	4.0	2.0	1.0	2.0	3.0	4.0	
3	1.0	2.0	3.0	4.0	1.0	3.0	4.0	2.0	
4	2.0	1.0	4.0	3.0	1.0	3.0	4.0	2.0	
5	1.0	3.0	4.0	2.0	2.0	1.0	4.0	3.0	
6	2.0	1.0	3.0	4.0	1.0	2.0	4.0	3.0	
7	1.0	2.0	4.0	3.0	1.0	3.0	2.0	4.0	
8	1.0	3.0	4.0	2.0	1.0	4.0	3.0	2.0	
9	4.0	1.0	3.0	2.0	2.0	1.0	4.0	3.0	
10	3.0	1.0	4.0	2.0	1.0	4.0	2.0	3.0	
	18.0	17.0	37.0	27.0	12.0	25.0	33.0	30.0	

a) Ranking of part 1 of instances with 10 actions.

	(0.70,0.20,0.10)				(0.75,0.20,0.10)				
	GA	PSO	SA	TS	GA	PSO	SA	TS	
1	3.0	1.0	4.0	2.0	1.0	4.0	3.0	2.0	
2	1.0	3.0	4.0	2.0	1.0	3.0	4.0	2.0	
3	3.0	1.0	2.0	4.0	2.0	1.0	4.0	3.0	
4	2.0	1.0	4.0	3.0	1.0	2.0	4.0	3.0	
5	1.0	3.0	4.0	2.0	2.0	1.0	3.0	4.0	
6	1.0	2.0	4.0	3.0	2.0	1.0	4.0	3.0	
7	2.0	1.0	4.0	3.0	3.0	1.0	4.0	2.0	
8	1.0	2.0	3.0	4.0	1.0	4.0	3.0	2.0	
9	1.0	2.0	4.0	3.0	3.0	1.0	4.0	2.0	
10	1.0	2.0	4.0	3.0	3.0	1.0	4.0	2.0	
	16.0	18.0	37.0	29.0	19.0	19.0	37.0	25.0	

b) Ranking of part 2 of instances with 10 actions.

Table 9 shows the Borda score of the ranks obtained by each algorithm in Table 8; Figure 7 compares the ranks graphically (grouped by tuples). Using each of the four groups of configurations presented in Table 7, several statistical tests were developed to validate the performance of the algorithms.

Table 9. Borda score from Table 8.

Algorithm	Total
GA	65.0
PSO	79.0
TS	111.0
SA	144.0

Summing up the information presented in this section, the centroids obtained using the GA statistically obtained a performance at least similar to the remaining algorithms, but they minimized the total number of inconsistencies found among all the instances (it achieved 2686 according to Table 7). Also, the GA results yielded the minimum number of inconsistencies in 23 of the 40 instances (see Table 8) reaching the best Borda score. Finally, if we take into account that the centroids resulting from this metaheuristic had also previously shown a good performance in estimating the parameter values of the preference model, then we can conclude that it has the best performance using only 10 alternatives.

4.3.3 Performance of prediction of centroids from S_{20} over the new set S_{100}

In this part of the experiment, the centroids were computed under the method proposed in Section 4.1, and having as input the set of instances S_{20} .

Similarly to Table 8, Table 10 summarizes the results from the validation process described in Section 4.2.3. Again, when validating these results with the Friedman test (and a significance level of 0.05), there were significant differences among the algorithms. Also, with the use of the Finner test it was found a superiority of GA over SA and TS. Besides, the Wilcoxon test, with a p -value of 0.008, finds significant differences between GA and PSO.

Table 12 shows a summary of the ranks obtained by each algorithm in Table 11. Figure 8 groups the ranks per tuple, and graphically compares them, supporting the conclusion that the best algorithm is indeed GA.

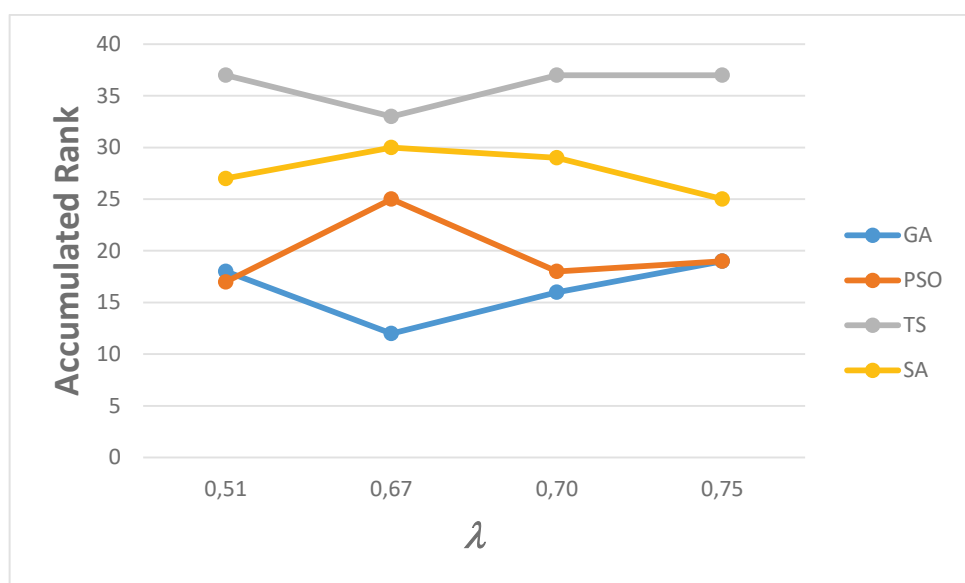


Figure 7. Performance of prediction of the centroids obtained per algorithm using the set S_{10} over the set S_{100} using: a) an accumulated sum of ranks; and, b) a graph of ranks per value λ . Note that each value λ is associated with particular values of (β, ϵ) , as indicated in Table 7.

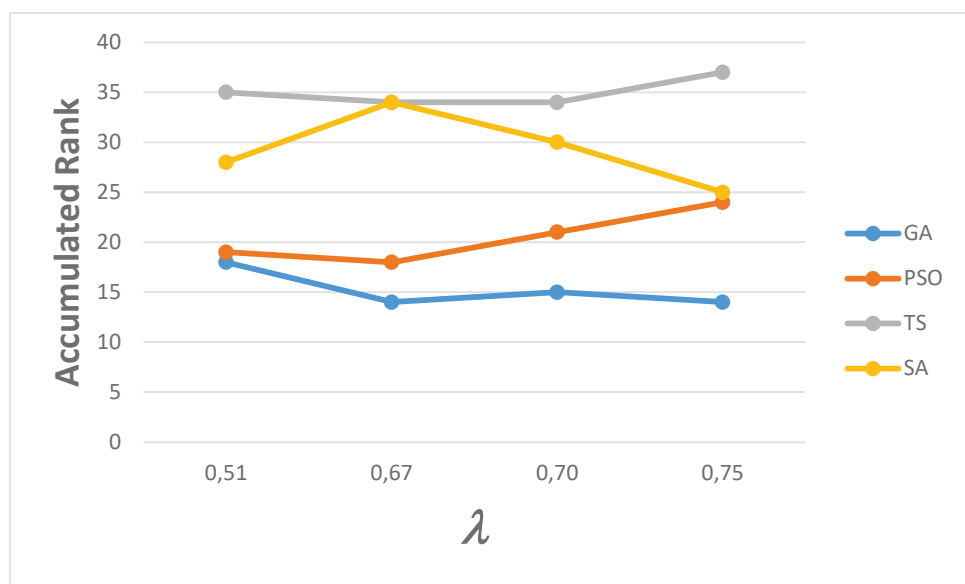


Figure 8. Performance of prediction of the centroids obtained by each algorithm using S_{20} over the set S_{100} using: a) an accumulated sum of ranks; and, b) a graph of ranks per value λ . Note that each value λ is associated with particular values of (β, ϵ) , as indicated in Table 7.

Table 10. Summary of the indicator I_p with results derived from the set S_{100} using centroids derived from the set S_{20} .

20 Actions								
	(0.51,0.15,0.07)				(0.67,0.15,0.07)			
	GA	PSO	SA	TS	GA	PSO	SA	TS
1	75	56	129	114	48	35	64	145
2	111	95	225	86	55	42	113	80
3	75	90	138	87	34	42	158	85
4	69	65	151	91	64	76	175	131
5	77	90	74	97	100	128	67	129
6	81	67	194	116	74	39	143	77
7	80	72	176	107	31	34	86	63
8	37	161	133	140	62	69	104	109
9	35	50	118	79	22	89	90	98
10	58	43	119	70	37	44	111	59
	698	789	1457	987	527	598	1111	976

a) Inconsistencies of part 1 of instances with 20 actions.

	(0.70,0.20,0.10)				(0.75,0.20,0.10)			
	GA	PSO	SA	TS	GA	PSO	SA	TS
1	37	49	72	55	41	64	96	48
2	46	73	224	118	23	51	52	42
3	52	37	200	98	51	40	54	62
4	51	118	46	59	47	56	139	79
5	42	77	84	66	87	35	119	110
6	28	37	45	56	97	200	108	129
7	39	50	56	54	21	63	136	36
8	50	85	78	108	32	91	115	63
9	111	65	146	72	64	60	119	56
10	34	33	87	55	21	43	84	60
	490	624	1038	741	484	703	1022	685

b) Inconsistencies of part 2 of instances with 20 actions.

Table 11. Ranking of the performance of the algorithms per instance, when the centroids obtained from S_{20} are validated using S_{100} .

20 Actions								
	(0.51,0.15,0.07)				(0.67,0.15,0.07)			
	GA	PSO	SA	TS	GA	PSO	SA	TS
1	2.0	1.0	4.0	3.0	2.0	1.0	3.0	4.0
2	3.0	2.0	4.0	1.0	2.0	1.0	4.0	3.0
3	1.0	3.0	4.0	2.0	1.0	2.0	4.0	3.0
4	2.0	1.0	4.0	3.0	1.0	2.0	4.0	3.0
5	2.0	3.0	1.0	4.0	2.0	3.0	1.0	4.0
6	2.0	1.0	4.0	3.0	2.0	1.0	4.0	3.0
7	2.0	1.0	4.0	3.0	1.0	2.0	4.0	3.0
8	1.0	4.0	2.0	3.0	1.0	2.0	3.0	4.0
9	1.0	2.0	4.0	3.0	1.0	2.0	3.0	4.0
10	2.0	1.0	4.0	3.0	1.0	2.0	4.0	3.0
	18.0	19.0	35.0	28.0	14.0	18.0	34.0	34.0

a) Ranking of part 1 of instances with 20 actions.

	(0.70,0.20,0.10)				(0.75,0.20,0.10)			
	GA	PSO	SA	TS	GA	PSO	SA	TS
1	1.0	2.0	4.0	3.0	1.0	3.0	4.0	2.0
2	1.0	2.0	4.0	3.0	1.0	3.0	4.0	2.0
3	2.0	1.0	4.0	3.0	2.0	1.0	3.0	4.0
4	2.0	4.0	1.0	3.0	1.0	2.0	4.0	3.0
5	1.0	3.0	4.0	2.0	2.0	1.0	4.0	3.0
6	1.0	2.0	3.0	4.0	1.0	4.0	2.0	3.0
7	1.0	2.0	4.0	3.0	1.0	3.0	4.0	2.0
8	1.0	3.0	2.0	4.0	1.0	3.0	4.0	2.0
9	3.0	1.0	4.0	2.0	3.0	2.0	4.0	1.0
10	2.0	1.0	4.0	3.0	1.0	2.0	4.0	3.0
	15.0	21.0	34.0	30.0	14.0	24.0	37.0	25.0

b) Ranking of part 2 of instances with 20 actions.

Table 12. Borda score from Table 11.

Algorithm	Total
GA	61.0
PSO	82.0
TS	117.0
SA	140.0

4.4. Discussion

The model proposed was tested under an experimental design involving two different sets of instances. The set S_{10} involved 40 instances, each having 10 actions and all the possible preferences established among them, which means a total of 45 judgments that should be made by the DM that is being modeled. The set S_{20} also involved 10 instances, but in this case each having 20 actions and a total of 190 judgments that should be made by the DM. On the set S_{10} , using GA, we obtained only 2686 inconsistencies on the 396,000 strict preference simulated statements counted over all the set of instances from the test set S_{100} . It means a 0.68% of error. On the other hand, the use of the set S_{20} combined with GA reduced the error to 0.55%. Although the results were improved by using a greater set of actions, the cognitive effort that it implies for the DM may be not justified; i.e., the increment of more than 400% in the amount of judgments that the DM must provide is probably not justified by a relatively weak improvement in the reduction of the number of inconsistencies. So, the DM-decision analyst couple should find a compromise between the model accuracy and the required cognitive effort.

In addition, and according to all the results already presented, it is considered that the best algorithm is GA, because it has the best rank value, i.e. it minimizes the inconsistencies in most instances. Further, the performance on the estimation of parameter values through the use of GA on the instances can be considered adequate because: a) it uses a small number of DM's judgments; b) it ranks better among all the algorithms; and, c) its superiority was proved through statistical tests.

5. Concluding Remarks

This work proposes an approach based on a Preference Disaggregation Analysis procedure to infer parameter settings for the model of preferences proposed by Fernandez³. The preference model is based on outranking relations which require the use of weights and thresholds parameters, denoted by (η, λ) , and some additional symmetric and asymmetric parameters (β, ε) . The relations involved are of strict preference, indifference, weak preference, incomparability, K-preference, and non-preference. The model of preferences under study has not been addressed by a

PDA methodology before. The proposed PDA strategy involves an optimization model for the indirect parameter setting, and its solution using a method based on meta-heuristics.

The optimization model, called Problem (4), is an optimization problem that approximates the values of the parameters $(\eta, \lambda, \beta, \varepsilon)$ through the minimization of the inconsistencies found between the preference statements produced by estimated parameter values, and preference judgments established by a Decision Maker (DM) through a reference set. This problem is based in three lexicographically ordered objectives. The most important objective relates the inconsistencies from the strict preference relation. The second objective in importance involves the combined inconsistencies from the indifference and weak preference relations. The third objective incorporates the inconsistencies found in the remaining relations.

The performance of the proposed PDA procedure was tested using four different algorithms: Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Tabu Search (TS), and Simulated Annealing (SA). The experimental design used to test their performance involves the estimation of the parameters $(\eta, \lambda, \beta, \varepsilon)$ in two different set of instances formed by small size reference sets. The quality of the best estimated parameters, called centroids, was evaluated by determining both: a) the percentage of deviation of the estimated parameters, and the expected ones; and, b) the number of inconsistencies obtained when using the estimated parameters in the prediction of preference relations on new outcomes, i.e. a new set of instances.

The results reveal that the algorithms GA and PSO had the best performance in relation with the percentage of deviation, because of their reductions in the gap between estimated and simulated parameter values of up to a 16% of deviation in the instances considered. However, the algorithm GA had the best quality in the prediction of preferences in new outcomes, or instances, having the minimum number of inconsistencies in more than 50% of the instances, in comparison with the other approaches. Also, it was revealed that our proposal requires a relatively small number of judgments defined by a DM to properly estimate parameter values that yield a minimal number of inconsistencies in new outcomes.

Acknowledgements

This work has been partially supported by PRODEP and the following projects: a) CONACYT Project 236154, b) Project 3058 from the program *Cátedras CONACyT*, and c) Project 269890 from CONACYT networks.

References

1. Deb, K. (2001). *Multi-Objective Optimization using Evolutionary Algorithms*, John Wiley & Sons, Chichester-New York-Weinheim-Brisbane-Singapore-Toronto. <http://dl.acm.org/citation.cfm?id=559152>.
2. Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*, Second ed., Springer, New York. <http://www.springer.com/us/book/9780387332543>.
3. Fernandez, E., Lopez, E., Lopez, F., Coello, C. (2011). *Increasing selective pressure toward the best compromise in Evolutionary Multiobjective Optimization: the extended NOSGA method*, Information Sciences 181:44–56. DOI:10.1016/j.ins.2010.09.007.
4. Coello, C.A. (1999). *A comprehensive survey of evolutionary-based multiobjective optimization techniques*. Knowledge and Information Systems, an International Journal, 1(3):269–308. DOI:10.1007/BF03325101.
5. Branke, J., Salvatore, C., Greco, S., Slowiński, Zielniewicz, P. (2016). *Using Choquet integral as preference model in interactive evolutionary multiobjective optimization*. European Journal of Operational Research, 250:884–901. DOI:10.1016/j.ejor.2015.10.027.
6. Miller G A. The magical number seven, plus or minus two: some limits on our capacity for processing information. Psychological Review, 63(2):81–97, 1956. DOI:10.1037/h0043158
7. Phelps S., Koksalan M. (2003) *An interactive evolutionary metaheuristic for multiobjective combinatorial optimization*. Management Science 49(12):1726 – 17. DOI:10.1287/mnsc.49.12.1726.25117
8. Deb K., Sinha A., Korhonen P., and Wallenius J. (2010). *An interactive evolutionary multiobjective optimization method based on progressively approximated value functions*. IEEE T Evolut Comput, 14(5):723–739. DOI: 10.1109/TEVC.2010.2064323
9. Branke, J., Greco, S., Slowiński, R., & Zielniewicz, P. (2015). *Learning value functions in interactive evolutionary multiobjective optimization*. IEEE Transactions on Evolutionary Computation, 19(1), 88–102. DOI:10.1109/TEVC.2014.2303783
10. Molina, J., Santana-Quintero L.V., Hernández-Díaz A.G., Coello Coello, C.A., and Caballero, R. (2009). *g-dominance: Reference point based dominance for multiobjective metaheuristics*. Eur J Oper Res, 197(2):685–692, 2009. DOI:10.1016/j.ejor.2008.07.015
11. Deb, K. (2007). *Current trends in evolutionary multi-objective optimization*, International Journal for Simulation and Multidisciplinary Design Optimisation 1(1):1–8. DOI:10.1051/ijsmdo:2007001.
12. Wagner, T., Trautmann, H. (2010). *Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions*. IEEE T Evolut Comput, 14:688–701. DOI: 10.1109/TEVC.2010.2058119
13. Greco, S., Slowinski R., Figueira J.R., Mousseau V. (2010). *Robust ordinal regression*. International Series in Operations Research & Management Science: In Trends in Multiple Criteria Decision Analysis, 142, 241–283. DOI:10.1007/978-1-4419-5904-1_9.
14. 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* (2nd ed.). Springer. DOI:10.1007/978-1-4939-3094-4_23.
15. Bechikh, S. (2013). *Incorporating Decision Maker's Preference Information in Evolutionary Multi-objective Optimization*, Diss. PhD thesis, High Institute of Management of Tunis, University of Tunis, Tunisia. <http://delta.cs.cinvestav.mx/~ccoello/EMOO/thesis-bechikh.pdf.gz>.
16. Roy, B. (1996). *Multicriteria Methodology for Decision Aiding*, Kluwer. DOI:10.1007/978-1-4757-2500-1.
17. 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. DOI: 10.1016/j.ins.2012.11.018
18. Cruz, L., Fernandez, E., Gomez, C., Rivera, G., Perez, F. (2014). *Many-objective portfolio optimization of interdependent projects with a priori incorporation of decision maker preferences*. Appl. Math & Inf. Sci, 8(4):1517–1531. DOI:10.12785/amis/080405.
19. 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. DOI:10.1016/j.ins.2015.03.064.
20. 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. DOI:10.2478/v10209-011-0010-0.
21. 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** (2009) 496–505. DOI: 10.1016/j.ejor.2008.11.035.
22. Greco, S., Mousseau, V., Slowinski, R. (2008). *Ordinal regression revisited: Multiple criteria ranking with a set of additive value functions*. EJOR **191**:415–435. DOI: 10.1016/j.ejor.2007.08.013
 23. Marchant, T. (2010). *Debate on How to assign numerical values to different parameters that aim at differentiating the role that the criteria have to play in a comprehensive preference model?*, 71 Meeting of the Euro Working Group Multiple Criteria Decision Aiding, Turin, Italy.
 24. Pirlot, M. (2010). *Debate on How to assign numerical values to different parameters that aim at differentiating the role that the criteria have to play in a comprehensive preference model?*, 71 Meeting of the Euro Working Group Multiple Criteria Decision Aiding, Turin, Italy.
 25. Mousseau, V., Slowinski, R. (1998). *Inferring an electre tri model from assignment examples*. Journal of Global Optimization **12**:157–174. DOI:10.1023/A:1008210427517.
 26. Dias, L., Mousseau, V. (2006). *Inferring electre's veto-related parameters from outranking examples*. European Journal of Operational Research, **170**:172–191. DOI: 10.1016/j.ejor.2004.07.044.
 27. 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. DOI:10.1080/18756891.2008.9727611.
 28. 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. DOI:10.1080/18756891.2015.1046324.
 29. Roy, B. (1990). *The outranking approach and the foundations of ELECTRE methods*, In: Bana and Costa, C.A., (ed.) Reading in Multiple Criteria Decision Aid, Springer-Verlag, Berlin, 1990, pp.155-183. DOI:10.1007/BF00134132.
 30. Figueira, J., Mousseau, V., Roy, B. (2005). *ELECTRE Methods*, In: Figueira, J., Greco, S. and Erghott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, Springer Science + Business Media, New York, 2005, pp.133-162. DOI:10.1007/0-387-23081-5_4.
 31. Brans, J.P., Mareschal, B. (2005). *PROMETHEE Methods*, In: Figueira, J., Greco, S. and Erghott, M. (eds.) Multiple Criteria Decision Analysis: State of the Art Surveys, Springer Science + Business Media, New York, 2005, pp. 163-190. DOI:10.1007/0-387-23081-5_5.
 32. Mousseau, V., Dias, L.C. (2004). *Valued outranking relations in ELECTRE providing manageable disaggregation procedures*. European Journal of Operational Research, **156**(2):467-482. DOI:10.1016/S0377-2217(03)00120-6
 33. Melanie, M. (1999). *An introduction to Genetic Algorithms*. Cambridge, Massachusetts: A Bradford Book the MIT Press. <https://mitpress.mit.edu/books/introduction-genetic-algorithms>.
 34. Arellano-Verdejo, J., Guzmán-Arenas, A., Godoy-Calderon, S., Barrón-Fernández, R. (2014). *Efficiently Finding the Optimum Number of Clusters in a Dataset with a New Hybrid Cellular Evolutionary Algorithm*. Computación y Sistemas, vol. **18**(2):313–327. DOI: 10.13053/CyS-18-2-2014-034
 35. Rangel-Valdez, N., Torres-Jimenez, J., Bracho-Rios, J., Quiz-Ramos, P. (2009, Oct). *Problem and algorithm fine-tuning: a case of study using Bridge Club and Simulated Annealing*. In: Proceedings of International Joint Conference on Computational Intelligence (IJCCI 2009), Madeira, Portugal. INSTICC Press, pp. 302–305. URL: <http://www.scitepress.org/DigitalLibrary/ProceedingsDetails.aspx?ID=BKrT7ZO14Ew=&t=1>.
 36. Kennedy, J., Eberhart, R. (1995, Nov/Dec). *Particle Swarm Optimization*. In Neural Networks, 1995. Proceedings, IEEE International Conference on., 4:942–1948. DOI:10.1109/ICNN.1995.488968.
 37. Clerc, M. (2012). *Standard Particle Swarm Optimization* (PDF). HAL open access archive. URL: https://hal.archives-ouvertes.fr/file/index/docid/764996/filename/SPSO_descriptions.pdf.
 38. Glover, F., Laguna, M. (1997). *Tabu Search*. Norwell, MA, USA: Kluwer Academic Publishers. <http://dl.acm.org/citation.cfm?id=549765>.
 39. Kirkpatrick, S., Gelatt Jr., C.D., Vecchi, M.P. (1983). *Optimization by simulated annealing*. Science. **220** (4598): 671–680. DOI:10.1126/science.220.4598.671.
 40. Perny, P. (1998). *Multicriteria filtering methods based on concordance and non-discordance principles*. Annals of Operations Research **80**:137-165. DOI:10.1023/A:1018907729570.

Appendix A. Calculus of the degree of credibility of outranking $\sigma(x,y)$

The ELECTRE methods introduced a binary relation S based on xSy . Proposition xSy (x outranks y) holds if, and only if, the DM has sufficient arguments in favor 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 G in agreement with that proposition is strong enough, and there is no important coalition discordant with it, see Ref. 16. This can be expressed by the following logical equivalence⁴⁰:

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

where:

- $C(x,y)$ is the predicate about the strength of the concordance coalition; this coalition is composed of two criterion subsets :
 - $C(xSy) = \{g_j \in G \text{ such that } g_j(x) - g_j(y) \geq -q_j\};$
 - $C(yQx) = \{g_j \in G \text{ such that } g_j(y) - p_j \leq g_j(x) < g_j(y) - q_j\}$ (Q denotes ‘weak preference’);
 - p_j and q_j denote the preference and indifference thresholds for criterion j ($p_j \geq q_j \geq 0$).
- $D(x,y)$ is the predicate about the strength of the discordance coalition $C(yPx) = \{j \in G \text{ such that } g_j(y) - g_j(x) \geq p_j\}$ (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 of xSy is calculated by:

$$\sigma(x,y) = c(x,y) \cdot Nd(x,y) \quad (A.1)$$

where $Nd(x,y)$ denotes the truth degree of the non-discordance predicate.

The concordance index $c(x,y)$ is defined as follows:

$$c(x,y) = \sum_G w_j c_j(x,y) \quad (A.2)$$

where:

w_j is the weight of the j -th criterion ($w_1 + w_2 + \dots + w_N = 1$); $c_j(x,y)$ is the marginal (partial) concordance index for the j -th criterion. This index is calculated by:

$$c_j(x,y) = \begin{cases} 0 & \text{if } j \in C(yPx) \\ (g_j(x) - g_j(y) + p_j) / (p_j - q_j) & \text{if } j \in C(yQx) \\ 1 & \text{if } j \in C(xSy) \end{cases} \quad (A.3)$$

The partial discordance is measured in comparison with a veto threshold v_j , which is the maximum difference $g_j(y) - g_j(x)$ compatible with $\sigma(x,y) > 0$. Following Mousseau and Dias in Ref. 32, we shall use a simplification of the original formulation of the discordance indices in the ELECTRE III method given by:

$$Nd(x,y) = \min_{j \in C(yPx)} [1 - d_j(x,y)] \quad (A.4)$$

$$d_j(x,y) = \begin{cases} 1 & \text{iff } \nabla_j \geq v_j \\ (\nabla_j - u_j) / (v_j - u_j) & \text{iff } u_j < \nabla_j < v_j \\ 0 & \text{iff } \nabla_j \leq u_j \end{cases} \quad (A.5)$$

where $\nabla_j = g_j(y) - g_j(x)$ and u_j is a discordance (pre-veto) threshold.

$\sigma(x,y)$ is calculated by combining Eqs. A.1–A.5 with Eq. (5).