

## Incorporation of decision-maker preferences in an interactive evolutionary multi-objective algorithm using a multi-criteria sorting

Laura Cruz-Reyes<sup>1</sup>, Eduardo Fernández<sup>2</sup>, Patricia Sánchez<sup>3\*</sup>, Claudia Gómez<sup>1</sup>  
<sup>1</sup>*Tecnológico Nacional de México: Instituto Tecnológico de Ciudad Madero, México,*  
*Universidad Autónoma de Sinaloa, México,* <sup>3</sup>*Tecnológico Nacional de México: Instituto*  
*Tecnológico de Tijuana, México.*

*lauracruzreyes@itcm.edu.mx, eddyf@uas.edu.mx, jpatricia.sanchez@gmail.com,*  
*claudia.gomez@itcm.edu.mx*

**Abstract.** Here, an interactive method is proposed to incorporate the preferences of the Decision Maker (DM) into the optimization process and lead the search towards the Region of Interest (ROI). The DM's preferences are expressed in a reference set and are reflected by an outranking model. This information is used by a multi-criteria sorting method to create selective pressure towards solutions that are satisfactory to the DM. Our method obtains a better characterization of the ROI when compared with the well known NSGA-II and A<sup>2</sup>-NSGA-III in simple and complex project portfolio problems.

**Keywords:** Multi-Objective Evolutionary Algorithms, multi-criteria sorting, incorporation of preferences, interactive method.

### 1. Introduction

A wide variety of problems in the real world involves several conflicting objectives to be optimized simultaneously under certain constraints [1]. As a consequence of the conflicting nature of the criteria, there is no single solution that simultaneously optimizes each objective, so most approaches look for a set of trade-off solutions. This type of problem is known in the literature as the Multi-objective Optimization Problem (MOP). Solving MOPs means finding the compromise solution that best satisfies the preferences of the Decision Maker (DM) [2].

Multi-objective Evolutionary Algorithms (MOEAs) have been widely used for solving MOPs (e.g. [3, 4]) because they are able to provide, as output, an approximation of the Pareto frontier (the non-dominated solutions) in a single run of the algorithm. However, as stated by Deb [5] and Fernandez et al. [2], one aspect that is often disregarded in the literature on MOEAs is the fact that the solution of a problem involves not only the search but also the decision-making process. That is, finding the Pareto frontier does not completely solve the problem; the DM still has to choose the best compromise solution out of that set. This is not a difficult task when dealing with problems having two or three objectives. However, as the number of criteria increases, the size of the Pareto frontier increases exponentially. Thus, it becomes harder or even impossible for the DM to establish valid judgments in order to compare many solutions with several conflicting criteria. Besides, the methods from the field of multi-criteria decision analysis do not perform well on such large decision problems, making it difficult to obtain a single solution [2].

As was stated by Miller in [6], the capacity of the human mind is restricted to handling a small amount of information at one time. This is a very serious obstacle when the DM has to compare a subset of non-dominated solutions to identify the best compromise solution in problems with many objectives. Hence, 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. The ROI is defined by Deb et al. [7] and Adra et al. [8] as the set of non-dominated solutions that are preferred by the DM over the other solutions. In similar terms, the ROI may be defined as a subset of the Pareto front whose elements are considered *satisfactory* by the DM. In order to guide the search towards the ROI, the DM must agree to incorporate his/her multi-criteria preferences into the search process.

In this paper we propose a method, called the *Interactive Multi-Criteria Sorting Genetic Algorithm* (I-MCSGA), for incorporating the DM's preferences interactively during the optimization process; the main contributions are summarized below. We propose to integrate the DM's preferences with an MOEA in such a way that they can be incorporated implicitly and interactively in the evolutionary optimization algorithm. The preferences are modelled by an outranking paradigm, and also embedded in a reference set. A multi-criteria sorting method is used to identify new *satisfactory* solutions among those generated by the evolutionary search. These solutions are used i) to create selective pressure towards the ROI; and ii) to enhance the reference set in order to increase its assignment capacity. From time to time, the current *satisfactory* solutions are shown to the DM, who updates the reference set with them. In this way, the DM 'learns' progressively about the optimization problem, adjusts his/her (initially poorly defined) preferences, and then drives the search towards the ROI.

The rest of the paper is organized as follows. In Section 2 we give an overview of preference-based evolutionary methods, and we also describe a multi-criteria sorting method (the THESEUS method). With this background, an interactive approach for incorporating preference information, and the implementation of this approach, called I-MCSGA, are detailed in Section 3. In Section 4, we present computer experiments used to confirm the advantages of the proposed approach. Finally, some conclusions are discussed in Section 5.

## 2. Background

### 2.1 A brief outline of preference-based evolutionary approaches

Multi-objective metaheuristic approaches have by now demonstrated their ability to approximate the whole Pareto front; however, the number of efficient solutions found is often too large. The selection of one solution (the final preferred alternative) from a huge set is evidently a difficult task for the DM, especially when the number of objectives increases [9]. As was mentioned above, because of humans' cognitive limitations, as described by Miller [6], it is essential to provide the DM with a reduced number of satisfactory alternatives. The DM is only interested in discovering the zone of the Pareto front corresponding to his/her preferences (the ROI), rather than the whole Pareto front. Therefore, information about the DM's preferences should be reflected by a representative model. The modelling of the preferences plays a key role in decision-making [10], since it will define the nature and organization of the information. The information about the DM's preferences can be expressed in diverse ways. Bechikh [9] states that the preference information structures most commonly used are the following: *weights* (e.g. [11, 12]), *ranking solutions* (e.g. [13, 14]), *ranking objectives* (e.g. [15, 16]), *reference point* (e.g. [7, 17]), *reservation point* (e.g. [18]), *trade-off between objectives* (e.g. [19]), *desirability thresholds* (e.g. [20]) and *outranking parameters* (e.g. [2, 21]).

The method in the recent paper of Oliveira et al. [22] lies in the last of these classes, because the preferences are expressed by an outranking relation and preference parameters. Oliveira et al. [22] use the ELECTRE TRI multi-criteria sorting method combined with an evolutionary algorithm. In ELECTRE TRI a reference profile is introduced to establish the boundary between two consecutive ordered categories. A critical aspect of ELECTRE TRI is defining the reference profiles because frequently is a very hard task for the DM, particularly when (s)he has just a vague conception regarding the boundary between adjacent categories. The existence of such boundaries is doubtful in many real-world problems (cf. [23, 24]). Besides, one can question whether a reference profile is sufficient for an acceptable characterization of the category related to it. If the object to be sorted were to be incomparable with several reference profiles, ELECTRE TRI would suggest inappropriate assignments.

The above approaches in evolutionary computation are grouped according to the different information structures used to incorporate the DM's preferences. Another way of classifying these approaches depends on the stage at which the preference information is articulated by the DM. According to Hwang and Masud [25], preferences can be requested in several ways: *a priori*, *a posteriori* and *interactively*. A brief description of these is given below.

In an *a priori* approach, the DM's preferences are articulated before the start of the method; the optimization process is then carried out by following the preference information. Afterwards, the optimization method finds the most-preferred point without additional interaction with the DM. However, the procedure is highly prone to error, since, unfortunately, the DM does not know how good is the best possible solution for the problem and how practical his/her aspirations are. Therefore, the DM may be dissatisfied with the outcome obtained. Some efforts in this direction can be found in [11, 26].

In an *a posteriori* method, the decision-making phase is initially ignored, and a representative portion of the Pareto-optimal frontier is generated before incorporating the DM's preferences. The idea is to define a subset of the complete set of non-dominated solutions and present it to the DM so that (s)he selects the most *satisfactory* solution as the final one. This approach

performs well in problems with few objectives, but the results deteriorate with an increase in the number of objective functions, as was discussed in the introduction to this paper. Some examples of *a posteriori* approaches are given in [27, 28].

The *interactive* approach depends on the progressive definition of the DM's preferences, together with the exploration of the objective space. Articulation of preferences is performed during the optimization process, so that progress towards a particular region of the Pareto-optimal frontier is made. The DM must be willing to participate in the solution process and direct it according to his/her preferences. As the interactive process advances in identifying better solutions, the DM not only specifies his/her preferences, but also learns about the problem and can thus adjust his/her level of aspiration. Considering that the DM is part of the solution method, any solution in the set obtained has a high possibility of being accepted as the final solution. As stated by Hwang and Masud [25], some disadvantages of interactive methods are: (1) the solutions rely on the precision of the local preference that can be shown by the DM; (2) for several approaches there is no assurance that the most preferred solution can be achieved within a finite number of interactive steps; and (3) more effort is required of the DM than with the other methods presented above. There is a large variety of works in evolutionary multi-objective optimization that address the *interactive* approach (e.g. [29, 30]).

According to Miettinen et al. [31], *interactive* methods lessen the disadvantages of *a priori* and *a posteriori* methods because the DM can progressively refine his/her initial preferences, and only solutions that are interesting to the DM are generated. For this reason, we propose an interactive approach for obtaining a reduced set of efficient solutions adapted to the DM's preferences.

In addition, the approach of Fernandez et al. [2], the so-called NOSGA2, is an important precedent for our work. The *a priori* way of incorporating preferences in NOSGA2 has recently been used by Cruz et al. [32] to optimize interdependent project portfolios with many objectives. In that paper, the authors proposed the *Non Outranked Ant Colony Optimization* (NO-ACO) method that is briefly described below.

## 2.2 Description of the NO-ACO model

The NO-ACO algorithm uses a set of agents called ants, and a local search, to perform its optimization process. This approach incorporates the DM's preferences following the model of preferences by Fernandez et al. [2] that is based on outranking relations proposed by Roy in [33]. It uses the degree of truth of the statement 'x is at least as good as y', which is represented by  $\sigma(x, y)$ , and can be calculated using outranking methods such as ELECTRE [34] and PROMETHEE [35]. Let us consider a threshold of acceptable credibility  $\lambda$ , an asymmetry parameter  $\beta$  to ensure the strict preference or *k*-preference, and a symmetry parameter  $\varepsilon$  for the indifference relation. For each pair of solutions  $(x, y)$ , the model identifies one of the preference relations given in Table 1. Given a set of feasible solutions  $O$ , the preferential system of NO-ACO establishes the sets shown in Table 2.

**Table 1.** The preference relations between each pair of solutions.

Preference relation	Definition	Conditions <sup>a</sup>
Strict preference ( $xPy$ )	The DM has clear and well-defined reasons justifying the choice of $x$ over $y$	$x$ dominates $y \vee (\sigma(x, y) \geq \lambda \wedge \sigma(y, x) < 0.5) \vee (\sigma(x, y) \geq \lambda \wedge [0.5 \leq \sigma(y, x) < \lambda] \wedge [\sigma(x, y) - \sigma(y, x)] \geq \beta)$ . (1)
Indifference ( $xIy$ )	The DM has clear and positive reasons that justify an equivalence between the two options	$\sigma(x, y) \geq \lambda \wedge \sigma(y, x) \geq \lambda \wedge  \sigma(x, y) - \sigma(y, x)  \leq \varepsilon$ . (2)
Weak preference ( $xQy$ )	The DM hesitates between $xPy$ and $xIy$	$\sigma(x, y) \geq \lambda \wedge \sigma(x, y) > \sigma(y, x) \wedge \neg xPy \wedge \neg xIy$ . (3)
Incomparability ( $xRy$ )	The DM perceives a high degree of heterogeneity between the two alternatives, so (s)he cannot express a preference	$\sigma(x, y) < 0.5 \wedge \sigma(y, x) < 0.5$ . (4)
<i>k</i> -preference ( $xKy$ )	The DM hesitates between $xPy$ and $xRy$	$0.5 \leq \sigma(x, y) < \lambda \wedge \sigma(y, x) < 0.5 \wedge \sigma(x, y) - \sigma(y, x) > \beta/2$ . (5)

<sup>a</sup>Considering  $(0 \leq \varepsilon \leq \beta \leq \lambda$  and  $\lambda > 0.5)$ .

**Table 2.** Sets defined by NO-ACO.

Set / Measure	Definition	Conditions
$S$	This is composed of the alternatives that strictly outrank $x$	$S(O, x) = \{y \in O \mid yPx\}.$ (6)
$NS$	This is known as the non-strictly-outranked frontier	$NS(O) = \{x \in O \mid S(O, x) = \emptyset\}.$ (7)
$W$	This is composed of the non-strictly-outranked solutions that are weakly preferred to $x$	$W(O, x) = \{y \in NS(O) \mid yQx \vee yKx\}.$ (8)
$NW$	This is known as the non-weakly-outranked frontier.	$NW(O) = \{x \in NS(O) \mid W(O, x) = \emptyset\}.$ (9)
$Net Flow$	This is the measure used to identify the preferred solutions by the DM on the non-strictly-outranked frontier	$F_n(x) = \sum_{y \in NS(O) \setminus \{x\}} [\sigma(x, y) - (y, x)].$ (10) where $F_n(x) > F_n(y)$ denotes a certain preference of $x$ over $y$
$F$	This is composed of non-strictly-outranked solutions that are greater in net flow than $x$	$F(O, x) = \{y \in NS(O) \mid F_n(y) > F_n(x)\}.$ (11)
$NF$	This is known as the net-flow non-outranked frontier.	$NF(O) = \{x \in NS(O) \mid F(O, x) = \emptyset\}.$ (12)

The problem that NO-ACO solves is

$$\min_{x \in O} \left\{ \left| S(O, x) \right|, \left| W(O, x) \right|, \left| F(O, x) \right| \right\}. \quad (13)$$

The best compromise solution is found through a lexicographic search, with pre-emptive priority favouring  $|S(O, x)|$ .

### 2.3 An outline of the THESEUS method

Let us recall, following [36], some basic aspects of the THESEUS method. THESEUS employs outranking relations to solve multi-criteria sorting problems, where sorting refers to problems in which the categories have been defined in an ordinal way [37]. The aim of the THESEUS method is to assign multi-criteria objects to preference-ordered categories. THESEUS rests on the following premises (cf. [36, 38]):

- i. There is a finite set of ordered categories  $Ct = \{C_1, \dots, C_M\}$ , ( $M \geq 2$ );  $C_M$  is assumed to be the preferred category.
- ii.  $U$  is the universe of objects  $x$  described by a coherent set of  $N$  real-valued criteria, denoted  $G = \{g_1, g_2, \dots, g_j, \dots, g_N\}$ , with  $N \geq 3$ .
- iii. There is a set of reference objects  $T$  (also called a *reference set* or *training set*), which is composed of elements  $b_{kh} \in U$  assigned to category  $C_k$ , ( $k = 1, \dots, M$ ).
- iv. The DM agrees with a fuzzy outranking relation  $\sigma(x, y)$  defined on  $U \times U$  (see Section 2.2). Its value models the degree of credibility of the statement ‘ $x$  is at least as good as  $y$ ’ from the DM’s perspective.

The THESEUS method is based on comparing a new object to be assigned with reference objects through models of preference and indifference relations (cf. [36]). The assignment is not a consequence of the object’s intrinsic properties: rather, it is the result of comparisons with other objects whose assignments are known. In the following,  $C(x)$  denotes a potential category for the assignment of object  $x$ . According to THESEUS,  $C(x)$  should satisfy some consistency rules:

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

$$\begin{aligned} xPb_{kh} &\Rightarrow C(x) \cdot C_k, \\ b_{kh}Px &\Rightarrow C_k \cdot C(x). \end{aligned} \quad (14.a)$$

$$\begin{aligned} xQb_{kh} &\Rightarrow C(x) \cdot C_k, \\ b_{kh}Qx &\Rightarrow C_k \cdot C(x). \end{aligned} \quad (14.b)$$

$$\begin{aligned} xIb_{kh} &\Rightarrow (C(x) \cdot C_k) \wedge (C_k \cdot C(x)) \\ &\Leftrightarrow C(x) = C_k. \end{aligned} \quad (14.c)$$

The relations  $P$ ,  $I$ , and  $Q$  were defined in Eqs. (1–3). The symbol  $\succeq$  denotes the statement ‘is at least as good as’ on the set of categories, which is related to the decision-making context. THESEUS uses the inconsistencies with Eqs. (14.a–c) to compare the possible assignments of  $x$ ; more specifically:

- The set of  $P$ -inconsistencies for  $x$  and  $C(x)$  is defined as  $D_P = \{(x, b_{kh}), (b_{kh}, x), b_{kh} \in T \text{ such that (14.a) is FALSE}\}$ ;
- The set of  $Q$ -inconsistencies for  $x$  and  $C(x)$  is defined as  $D_Q = \{(x, b_{kh}), (b_{kh}, x), b_{kh} \in T \text{ such that (14.b) is FALSE}\}$ ;
- The set of  $I$ -inconsistencies for  $x$  and  $C(x)$  is defined as  $D_I = \{(x, b_{kh}), (b_{kh}, x), b_{kh} \in T \text{ such that (14.c) is FALSE}\}$ .

Suppose that  $C(x) = C_k$  and consider  $b_{jh} \in T$ . Some cases in which  $x$  and  $b_{jh}$  belong to adjacent categories and nevertheless  $xIb_{jh}$  may be explained by ‘discontinuity’ of the description;  $x$  may be close to the upper (lower) boundary of  $C_k$  and  $b_{jh}$  may be close to the lower (upper) boundary of  $C_j$ . These are called second-order  $I$ -inconsistencies and are grouped in the set  $D_{2I}$ . The set  $D_{1I} = D_I - D_{2I}$  contains the so-called first-order  $I$ -inconsistencies, which are not consequences of the discontinuity effect described above.  $n_P, n_Q, n_{1I}$ , and  $n_{2I}$  denote the cardinalities of the inconsistency sets defined above. Let  $N_1 = n_P + n_Q + n_{1I}$ , and  $N_2 = n_{2I}$ .

THESEUS suggests an assignment that minimizes the above inconsistencies with lexicographic priority favouring  $N_1$ , which is considered the most important criterion [36]. The basic assignment rule is:

For each  $x \in U$  and given a minimum credibility level  $\lambda > 0.5$

- i. Starting with  $k=1$  ( $k=1, \dots, M$ ) and considering each  $b_{kh} \in T$ , calculate  $N_1(C_k)$ ;
- ii. Identify the set  $\{C_j\}$  whose elements hold  $C_j = \operatorname{argmin} N_1(C_k)$
- iii. Select  $C_{k^*} = \operatorname{argmin} N_2(C_i)$ ;  
 $\{C_j\}$
- iv. Assign  $x$  to  $C_{k^*}$ .

The suggestion may be a single category or a sequence of categories. The first case is called a ‘precise assignment’. Otherwise, the multi-category solution obtained highlights the highest category ( $C_H$ ) and the lowest category ( $C_L$ ); each category in this interval may be acceptable for the assignment of the object, but THESEUS fails to determine the most appropriate. A solution of this type is called an ‘imprecise assignment’.

According to recent studies by Fernandez et al. [38, 39], the capacity of THESEUS for suggesting appropriate assignments increases with the cardinality of the reference set. Fernandez et al. [38] proved that this capacity is improved with an automatic enhancement of the reference set, that is, when new objects are assigned by THESEUS and incorporated into the reference set without requiring acceptance by the DM.

### 3. Our proposal

In this section, an interactive approach to search the ROI, and a possible implementation of thereof will be presented. In our work, the DM’s preferences are expressed in a reference set and an outranking model. The preference information is exploited by a multi-criteria sorting method to detect new satisfactory solutions generated by an evolutionary approach, and that way it drives the search towards the ROI. The DM’s preferences will be updated at some stage during the optimization process. A general vision of this approach is shown in Fig. 1.

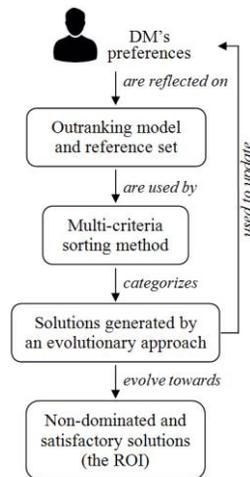


Fig. 1. The general scheme of the proposed approach to search the ROI.

### 3.1 An interactive approach to search the ROI based on a multi-criteria sorting method

The proposal suggests using two reference sets, each one is divided into two categories: *satisfactory* and *unsatisfactory*. The first set, called  $T_{upd}$ , reflects the DM's preferences that were obtained in the last interaction with him/her; the second one, called  $T_{enh}$ , accumulates new solutions through an automatic process. The two sets are identical at the beginning of the approach. The use of these sets and the description of the processes of our interactive approach are detailed below; Figure 2 illustrates these processes.

- i. **Interact with the DM to update preference information.** This process aims to model the current DM's preference information, from his/her local knowledge. This information is represented by a set of solutions that the DM considers *satisfactory* or *unsatisfactory*, that is, a reference set ( $T_{upd}$ ) that implicitly reflects his/her multi-criteria preferences. The interaction with the DM is performed at two points in time: *a*) at the beginning of the process, when  $T_{upd}$  is created with the first definition of the DM's preferences; and *b*) during the search process, when  $T_{upd}$  is updated by the DM. The number of interactions will be limited by a given input parameter. The update can take place of the second interaction onwards and it is performed as set out below. First, the set  $T_{enh}$  is shown to the DM;  $T_{enh}$  contains new solutions categorized as *satisfactory* that is the result of an automatic enhancement process, which is described in (iii). The DM selects the solutions in that set that (s)he really considers *satisfactory*. Once the DM has performed this validation, the solutions in  $T_{upd}$  and  $T_{enh}$  must be replaced by the chosen solutions by the DM. The interaction allows the DM to learn, as new solutions are shown to him/her, and in this way (s)he adjusts his/her notion of what a *satisfactory* solution is. The update also allows the DM to correct possible errors of assignment that could be generated during the automatic enhancement procedure of  $T_{enh}$ . After each interaction, the reference set  $T_{upd}$  will contain the latest expression of the DM's preferences. The validation of the preference information can be performed by a real DM or by an algorithm able to behave in a way that is similar to a real DM. It is important to highlight that the updating process is the essential part of this proposal, because by incorporating the DM into the solution approach, any of the solutions obtained at the end of the optimization process can be the solution to be implemented.
- ii. **Progressively explore the solution space.** This process aims to create selective pressure towards the ROI, through an exploration method that identifies, at least locally, new non-dominated solutions that would probably be considered as *satisfactory* by the DM. To perform this process two things are required: *a*) a procedure to guide the search towards efficient solutions in the sense of Pareto optimality; and *b*) an approach to capture the notion of a *satisfactory solution*. For the second of these, we consider it ideal to use a multi-criteria sorting method. This method will assign to predefined categories the new non-dominated solutions derived from the search process, using the preference information embedded in  $T_{enh}$ . This evaluation will help to direct selective pressure towards solutions assigned as *satisfactory*.
- iii. **Improve the capacity of the sorting method for making category assignments.** This procedure aims to incorporate automatically new solutions into the reference set  $T_{enh}$  as the search progresses.  $T_{enh}$  will accumulate new solutions that have been assigned as *satisfactory* by the sorting method that uses  $T_{upd}$  as the reference set. Although the incorporated solutions have not been validated by the DM, the use of a more populated reference set will help the sorting method to make more appropriate assignments [38].

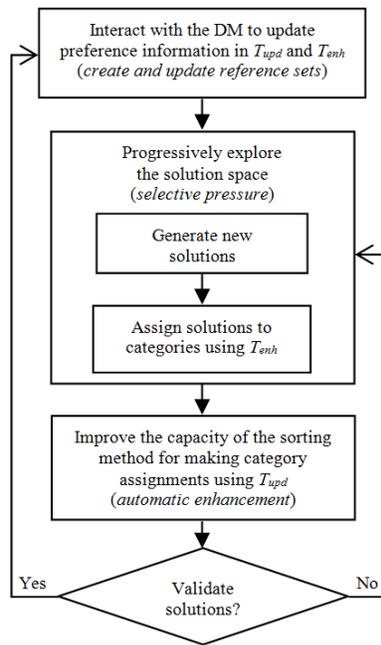


Fig. 2. An interactive approach to search the ROI.

### 3.2 One way to specify the proposed approach: The Interactive Multi-Criteria Sorting Genetic Algorithm

Here, we present a way to implement the approach proposed in Section 3.1. This method, called the *Interactive Multi-Criteria Sorting Genetic Algorithm* (I-MCSGA), is based on the following points:

- The DM's preferences are reflected by an outranking model  $\sigma(x,y)$  built using ELECTRE III; this model contains a set of preference parameters (weights and thresholds of indifference, preference and veto). The DM's preferences are also embedded in the reference set  $T_{upd}$  in which a set of solutions (actual or potential ones) are assigned by the DM as *satisfactory* and *unsatisfactory*. The set  $T_{upd}$  therefore contains the updated information about the DM's preferences. The preference parameters in  $\sigma$  should be compatible with the preference information in  $T_{upd}$ . Such compatibility may be guaranteed (although this is not mandatory) by an indirect parameter elicitation method (e.g. [40]).
- Since the ROI may be defined as a subset of the Pareto front whose elements are considered *satisfactory* by the DM, our method privileges (in a current population) solutions that are non-dominated and are considered *satisfactory* by a multi-criteria sorting method; in this work we use the THESEUS method. The way of creating selective pressure towards non-dominated solutions is inspired by the Non-dominated Sorting Genetic Algorithm-II (NSGA-II) [27], and is strengthened with the sorting of solutions.
- The THESEUS method is used to make decisions on whether a particular solution in a current population is *satisfactory* or not. For making assignments, THESEUS uses the enhanced reference set  $T_{enh}$ . Before any enhancement  $T_{enh} \equiv T_{upd}$ .  $T_{enh}$  is automatically enhanced with new generated solutions that have been assigned to the *satisfactory* category by THESEUS using  $T_{upd}$  as reference set.
- After a certain number of iterations, the *satisfactory* category of  $T_{enh}$  is presented to the DM, who updates  $T_{upd}$  with those solutions. The preference parameters may be updated from the current  $T_{upd}$ .

Remarks:

- As stated above,  $T_{upd}$  contains the most updated preferences from the real DM. Such an updating process is necessary for two main reasons: i) to correct the imprecise assignments made by the automatic enhancement of  $T_{enh}$ ; and ii) because the DM 'learns' as (s)he 'discovers' new solutions and (s)he adjust his/her notion of what a *satisfactory* solution is.
- The cardinality enhancement of  $T_{enh}$  should improve its capacity to make appropriate assignments [38, 39], thus improving the selective pressure towards the ROI. Besides, cardinality  $(T_{enh}) \geq$  cardinality  $(T_{upd})$  should be ensured.

- At the end of the search process, the *satisfactory* solutions of  $T_{enh}$  will be presented to the DM so that (s)he chooses the final solution from that set.

The general method of the I-MCSGA is shown in Algorithm 1. Note that it is composed of two main phases that will be described in the following subsections.

---

**Algorithm 1.** Procedure I-MCSGA

---

**Input:**  $L, num\_iterations, enh\_number, enh\_interval, upd\_number, upd\_interval$

**Output:** *satisfactory* category of  $T_{enh}$

**Phase 1:**

1. Construct an initial reference set  $T_{upd}$  //Section 3.2.1
2. Initialize  $T_{enh}$  with  $T_{upd}$

**Phase 2:**

3. Set  $\sigma$ -parameters agreeing with  $T_{upd}$  (using, for example, an indirect parameter elicitation method)
  4. Initialize parent population  $P$  with  $T_{upd}$  and complement it with random individuals until a size  $L$  is achieved
  5. Generate non-dominated fronts on  $P$  (based on objective function values)
  6. Give to these fronts a rank (level)  $F_i$  according to non-domination level
  7. Categorize the solutions in  $F_1$  using the THESEUS sorting method ( $\sigma$ ) //Section 3.2.2
  8. Generate a child population  $Q$  of size  $L$  by applying selection, recombination, and mutation operators on  $P$
  9. FOR I=1 to  $num\_iterations$  DO
  10.  $P' = P \cup Q$
  11. Generate non-dominated fronts on  $P'$  (based on objective function values)
  12. Give to these fronts a rank (level)  $F_i$  according to non-domination level
  13. Categorize the solutions in  $F_1$  using the THESEUS sorting method ( $\sigma$ ) //Section 3.2.2
  14. Create from  $P'$  a new parent population  $P$  of size  $L$  by using a diversity operator //Section 3.2.3
  15. Generate a child population  $Q$  of size  $L$  by applying selection, recombination, and mutation operators on  $P$
  16. Enhance  $T_{enh}$  a certain number of times ( $enh\_number$ ) every certain number of iterations ( $enh\_interval$ ) //Section 3.2.4
  17. Update  $T_{upd}$  (by interacting with the DM) a certain number of times ( $upd\_number$ ) every certain number of iterations ( $upd\_interval$ ) //Section 3.2.5
  18. End FOR
  19. Repeat steps 10-13 and 16
  20. Give the DM the satisfactory solutions of  $T_{enh}$
  21. End PROCEDURE
- 

### 3.2.1 A method to construct an initial reference set

In the first phase of the I-MCSGA, the DM is prompted to provide a reference set, which we call  $T_{upd}$ . The categories in that set are *satisfactory* and *unsatisfactory*. In order to create the reference set, the DM has the following two choices:

- Provide a set of solutions in the objective space and, according to their objective values, indicate which of them are considered *satisfactory*, by using his/her current (limited) knowledge about the problem. These solutions do not necessarily have to be feasible, since they will only be used as a reference to start the search process. The DM may refine his/her aspiration levels during his/her interaction with the optimization process.
- Run a multi-objective metaheuristic that matches the characteristics of the problem to be addressed. This approach will be used to obtain an approximation to the Pareto frontier. These solutions will be sorted by the DM into the set of categories to form the initial reference set. When this option is used to create the reference set, the method becomes a hybrid approach.

Whichever option is used, the DM's preferences will be reflected in the reference set  $T_{upd}$ . In this phase, the reference set  $T_{enh}$  is initialized with  $T_{upd}$ , that is, at the beginning of the optimization process the two sets are identical. On the other hand, according

to Fernandez et al. [38], reference sets of large size should supply more knowledge about the assignment policy, leading to more appropriate assignments. Hence, in cases where the *satisfactory* category of the reference set is poorly populated, we recommend increasing the cardinality of this category. Therefore, we propose an approach for adding fictitious solutions (derived from an existing solution) to extend and intensify this category. The procedure to generate these solutions is as follows:

- i. Identify a pair of objectives with nearly equal weights, called *similar weight objectives*;
- ii. Create a replica of an existing solution; and
- iii. Modify the *similar weight objectives* of the replicated solution, adding to one of them and subtracting from the other one the same predefined value.

We do this in order to make a slight variation in the objectives of the existing solution, in the sense of improvement and compensation. An example of this procedure is given in Table 3.

**Table 3.** Real and fictitious solutions.

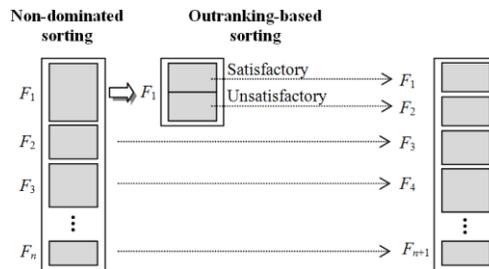
Reference element	Objective values/weights				Category
	$N_1/0.27$	$N_2/0.15$	$N_3/0.26$	$N_4/0.32$	
2	67655	53740	3145	3580	Satisfactory (real)
3	68155	53740	2645	3580	<i>Satisfactory</i> (fictitious)
Absolute difference	500	0	500	0	

### 3.2.2 A way to create selective pressure towards the ROI

The second phase of the I-MCSGA is a variant of the NSGA-II [27] where the main change is the incorporation of the THESEUS method (Section 2.3) which uses the current reference set  $T_{enh}$  to identify *satisfactory* solutions in the search process. THESEUS is used in the present paper as an assignment tool in order to characterize the ROI. We chose THESEUS because it has given good results with artificial and real-world data (cf. [36]). In comparison with other multi-criteria sorting methods, THESEUS can handle more general and larger reference sets, thus providing more suitable assignments (cf. [41]). Our approach works like the NSGA-II but with the following additional steps:

- i. Calculate  $\sigma$  on  $F_1 \times T_{enh}$ , where  $F_1$  is the non-dominated front of NSGA-II (the first front);
- ii. Each solution  $x \in F_1$  is assigned by THESEUS to one category  $C_k$  of the set  $\{satisfactory, unsatisfactory\}$  by using  $T_{enh}$  as reference set;
- iii.  $F_1$  is divided into two sub-fronts; the first ranked sub-front contains the solutions that were assigned to the most preferred category (*satisfactory*);
- iv. The remaining fronts of the current NSGA-II population are re-ordered by considering each sub-front of the original  $F_1$  as a new front; the NSGA-II's elitism involves the new  $F_1$ , which is now constituted by the non-dominated solutions belonging to the most preferred category.

The above steps are illustrated in Fig. 3. In a MOP, the ROI should be composed of solutions that belong to the most preferred category. Hence, the solutions in the ROI are characterized by the fact that they are *i)* non-dominated, and *ii)* considered *satisfactory* solutions by the DM. Therefore, our approach creates a selective pressure towards solutions that have both features. In addition to incorporating the THESEUS method to search the ROI, our approach: a) uses diversity metrics according to the problem size; b) automatically enhances the reference set  $T_{enh}$ ; and c) allows interaction with the DM to update his/her preferences. These processes are described in the following sections.



**Fig. 3.** Multi-criteria sorting by I-MCSGA.

### 3.2.3 Diversity-preservation operators

As with any other genetic procedure, our method uses a diversity-preservation operator that is adapted according to the number of objective functions of the problem. In problems with three and four objectives, it is considered acceptable to use the well-known *crowding distance based operator*, where the density of solutions in the neighbourhood is measured [27]. However, as was stated in [42], the crowded-comparison operator is not suitable for many-objective optimization problems. For this reason, when dealing with many-objective problems (problems with five or more objectives), we use the *reference point based operator* of the recent proposal called Adaptive NSGA-III (A<sup>2</sup>-NSGA-III) [43], where the maintenance of diversity between population members is aided by generating and adaptively updating a set of well-distributed reference points. This new operator incorporates the following operations: i) normalization of the values of the objective functions and the reference points in order to keep the ranges similar; ii) association of every population member with a specific reference point; iii) a niching procedure in order to ensure a diverse set of solutions; and iv) an update process that identifies those reference points that are not associated with a population member, allowing those points to be relocated by using addition and deletion strategies. These same actions are implemented in our method, with a slight adjustment in the normalization operation as recommended in [42].

### 3.2.4 An automatic procedure for enhancing the reference set

The reference set  $T_{enh}$  is automatically enhanced by the addition of some new solutions created during the search process. Each solution  $x \in F_1$ , where  $F_1$  is the first sub-divided front (see Section 3.2.2 step iv), is considered in order to perform the automatic enhancement of  $T_{enh}$ . A solution  $x$  belongs to the *satisfactory* category of  $T_{enh}$  if and only if the following two conditions are fulfilled:

- i.  $x$  is assigned to the *satisfactory* category by THESEUS using  $T_{upd}$  as the reference set;
- ii. There is no solution  $b \in T_{enh}$  such that  $b$  dominates  $x$ .

These two conditions (called *valid\_assignment*) are used in the enhancement process of  $T_{enh}$  that is described in Algorithm 2.

---

**Algorithm 2.** Automatic enhancement procedure

---

**Input:**  $F_1, T_{enh}$

**Output:**  $T_{enh}$  enriched

```
1. FOR each  $x \in F_1$  DO
2.   IF  $x$  fulfils the conditions valid_assignment THEN
3.     FOR each  $y \in satisfactory$  DO
4.       IF  $x$  dominates  $y$  THEN
5.         Remove  $y$  from satisfactory and incorporate it into
           unsatisfactory
6.     End FOR
7.     Add  $x$  to satisfactory
8.   ELSE
9.     Add  $x$  to unsatisfactory
10. End FOR
11. Return  $T_{enh}$  enriched
12. End PROCEDURE
```

---

Thus, the procedure itself decides which new solutions should be incorporated into the reference set  $T_{enh}$ . According to Fernandez et al. [38], THESEUS suggests more appropriate assignments when the cardinality of its reference set is increased, that is, when it works with larger reference sets. Therefore, the capacity of THESEUS to suggest appropriate assignments increases when  $T_{enh}$  is automatically enhanced; and consequently the selective pressure towards the ROI is improved. Another important aspect stated by Fernandez et al. [39] is the fact that THESEUS tends to perform better with a uniform distribution of reference objects per category. Based on this observation, we try to maintain approximately the same number of elements in the two categories of  $T_{enh}$ . In the absence of balance between the categories, the oldest solutions are removed from the category with more elements until both categories have nearly the same cardinality. Likewise, oldest solutions are also removed when one or both of the categories exceed an upper limit of elements ( $\approx 100$ ). The enhancement process can be carried out once in every certain number of iterations; this interval is set as input at the start of the method.

### 3.2.5 An interactive method for updating preferences

Our method takes advantage of the fact that the DM can progressively learn about the problem, thereby adjusting his/her preferences interactively, to drive the search towards the preferred zone of the Pareto optimal frontier (the ROI). For this, it is necessary to reflect the DM's modified preferences by updating the current reference set. As was stated before, the DM's preferences are reflected in  $T_{upd}$ , and therefore it is necessary to update this set. The updating is carried out as follows:

- i. The *satisfactory* category of  $T_{enh}$  is presented to the DM who selects, from that set, the solutions that (s)he considers *satisfactory* according to his/her preferences.
- ii. The solutions selected as *satisfactory* by the DM now form the new *satisfactory* category of  $T_{upd}$ .
- iii. The remaining solutions of  $T_{enh}$  now form the new *unsatisfactory* category of  $T_{upd}$ .
- iv. After updating the DM's preferences, do  $T_{enh} \equiv T_{upd}$ .

The updating of  $T_{upd}$  can be performed once in every certain number of iterations; this parameter is introduced as input at the beginning of the method. The number of updates depends on the time and effort that the DM is willing to spend. On the other hand, updating the DM's preferences will help to eliminate possible inappropriate assignments suggested by THESEUS, since possible inconsistencies between the reference set and the preference model parameters can cause wrong assignments, as demonstrated in the study by Fernandez et al. [39]. Once the DM updates his/her preferences in  $T_{upd}$ , the same procedure as is described in Section 3.2.4 is applied to maintain the balance between the cardinality of the categories, if necessary.

## 4. Some computer experiments

Let us consider two experiments that use the I-MCSGA to address the *Public Project Portfolio Problem*, which will be described in Section 4.3. In both experiments, we want to verify that our method leads to a good characterization of the ROI. We conduct the first experiment to determine whether the I-MCSGA is capable of obtaining better solutions than those obtained by the NSGA-II method in problems with three and four objectives; the second experiment explores whether our method outperforms A<sup>2</sup>-NSGA-III in many-objective problems (nine and sixteen objectives).

The configurations used in both experiments are outlined below.

- In the first phase of the I-MCSGA, the NO-ACO algorithm (Section 2.2) is used to obtain an approximation to the Pareto frontier. This set of solutions is employed to construct the initial reference set  $T_{upd}$ . The NO-ACO parameters are the same as those reported in [32].
- The parameters of the outranking model are set to the values suggested by Fernandez et al. in [2].
- To ensure relatively fair conditions, the parameters for NSGA-II, A<sup>2</sup>-NSGA-III and the second phase of the I-MCSGA are the same: crossover probability = 1; mutation probability = 0.01; the number of iterations = 500.
- The data for the projects (e.g., cost, area, and region) are different for each instance.
- The algorithms are programmed in the Java language, using the JDK 1.7 compiler, and NetBeans 7.1 as IDE, and they are run 30 times for each instance on a Mac Pro with an Intel Quad-Core 2.8 GHz processor and 3 GB of RAM.
- In the absence of a real DM, we simulate him/her using the relational system of preferences proposed by Fernandez et al. [2] (Section 2.2). As a result, the creation of the initial reference set  $T_{upd}$  and its interactive updating is done by the preference model. These procedures are described below.

### 4.1 Constructing an initial reference set without a real DM

To construct the initial reference set  $T_{upd}$ , the solutions obtained by NO-ACO are categorized using the preference model. The categories considered to create the reference set are *satisfactory* and *unsatisfactory*. The reference set is constructed according to the following steps:

- i. Run NO-ACO to find a set  $A$  of solutions containing a subset of the approximate Pareto frontier.
- ii. Create the *satisfactory* category with those solutions belonging to the known Pareto frontier that satisfy  $|S(A, x)| = 0$  (Eq. (6)), that is, that belong to the non-strictly-outranked frontier, and that also each of them fulfil one of the following conditions:
  - a. Satisfies  $|W(A, x)| = 0$  (Eq. (8)) and  $|F(A, x)| = 0$  (Eq. (11)), that is, the solution belongs to the non-weakly and net-flow non-outranked frontier in  $A$ . The set of these solutions is known as the *Best Solutions* (BS).

- b. Is solution indifferent (Eq. (2)) to any solution in BS.
  - c. Is a non-dominated solution (minimization) with respect to the objectives  $|W(A, x)|$  and  $|F(A, x)|$  in  $A$ .
  - d. Has positive net flow score (Eq. (10)).
- iii. Create the *unsatisfactory* category with the remaining solutions generated by NO-ACO in step i.

#### 4.2 Updating the reference set without a real DM

To perform the updating of  $T_{upd}$ , the following steps are performed automatically:

- i. Re-assign the solutions of the *satisfactory* category of  $T_{enh}$  using the simulated DM (Section 4.1 step ii).
- ii. Create the new *satisfactory* category of  $T_{upd}$  with the solutions re-assigned by the simulated DM as *satisfactory* in  $T_{enh}$ .
- iii. Create the new *unsatisfactory* category of  $T_{upd}$  with the remaining solutions of  $T_{enh}$ .
- iv. Apply the procedure described in Section 3.2.4 to maintain approximately the same number of reference elements in the two categories of  $T_{upd}$ .
- v. Do  $T_{enh} \equiv T_{upd}$ .

#### 4.3 Case study: A public project portfolio problem

Let us consider a decision-making situation in which the DM is in charge of selecting a group of projects (a portfolio) that will be implemented by his/her organization. The aim of this decision problem is to choose the ‘best’ portfolio that satisfies some budget constraints. Formalizing these concepts, let us consider a set of  $N$  projects, where the  $i$ th project is represented by a  $p$ -dimensional vector  $f(i) = \langle f_1(i), f_2(i), f_3(i), \dots, f_p(i) \rangle$ , where each  $f_j(i)$  indicates the contribution of project  $i$  to the  $j$ th objective. Each objective denotes the benefit target; that is, people belonging to a social category (e.g., Extreme Poverty, Poverty, Middle), who receive a benefit level (e.g., High Impact, Middle Impact, Low Impact) from the  $i$ th project.

On the other hand, a portfolio  $x$  is a subset of these projects, which is usually modelled as a binary vector  $x = \langle x_1, x_2, \dots, x_N \rangle$ . In this vector,  $x_i$  is a binary variable where  $x_i = 1$  if the  $i$ th project is supported and  $x_i = 0$  otherwise.

There is a total budget that the organization is willing to invest, which is denoted as  $B$ ; each project has an associated cost  $c_i$ . Portfolios are subject to the budget constraint:

$$\left( \sum_{i=1}^N x_i c_i \right) \leq B. \quad (15)$$

The  $i$ th project corresponds to an area (e.g., health, education) denoted by  $a_i$ . Each area has budgetary limits defined by the DM or another competent authority. Let us consider, for each area  $k$ , a lower and an upper limit,  $L_k$  and  $U_k$  respectively. Based on this, the constraint for each area  $k$  is

$$L_k \leq \sum_{i=1}^N x_i g_i(k) c_i \leq U_k. \quad (16)$$

where  $g$  is defined as

$$g_i(k) = \begin{cases} 1 & \text{if } a_i = k, \\ 0 & \text{otherwise} \end{cases}. \quad (17)$$

Besides, each project corresponds to a geographical region that will benefit from the project. In the same way as for the areas, each region has lower and upper limits as another constraint that must be fulfilled by a feasible portfolio.

The quality of a portfolio  $x$  is determined by the union of the benefits of each of the projects that compose it. This can be expressed as

$$z(x) = \langle z_1(x), z_2(x), z_3(x), \dots, z_p(x) \rangle. \quad (18)$$

where  $z_j(x)$ , in its simplest form, is calculated as

$$z_j(x) = \sum_{i=1}^N x_i f_j(i). \quad (19)$$

If we denote by  $R_F$  the region of feasible portfolios, the problem of the project portfolio is to identify one or more portfolios that solve

$$\max_{x \in R_F} \{z(x)\}. \quad (20)$$

In this problem, the only accepted solutions are those portfolios that satisfy the following constraints: the total budget constraint (Eq. (15)), the area constraints (Eq. (16)), and the region constraints (similar to Eq. (16)).

#### 4.4 First experiment: NSGA-II vs I-MCSGA

The first experiment consists of comparing the quality of the solutions provided by NSGA-II against those obtained from I-MCSGA in addressing problems with three and four objectives. We test six random instances whose basic information is shown in Table 4. In both algorithms, the population size is 100. The number of iterations to carry out the automatic enhancement is set to one. This means that the automatic enhancement process is performed at each iteration, namely 500 times. The number of iterations to execute the updating of the reference set  $T_{upd}$  is set to 200. That is to say, the updating is carried out only twice throughout the whole optimization process.

**Table 4.** Information about instances used in the first experiment.

Instance	Instance Description	
	Objectives	Projects
1	3	100
2	3	100
3	3	100
4	4	25
5	4	25
6	4	25

In Table 5, we present the average results of the 30 runs carried out for each instance. Instances 1–3 correspond to problems with three objectives. Although in this dimension NSGA-II is very competitive, the results reveal that on average our approach dominates between 1% and 7% of the solutions suggested by NSGA-II (Column 5). Conversely, in only one instance was there a solution of I-MCSGA that was dominated by a solution found by NSGA-II. No NSGA-II solution belongs to the *satisfactory* category, whereas almost all I-MCSGA solutions were *satisfactory* (Column 7). Instances 4–6 correspond to problems with four objectives. We can see that the solutions from our method dominate, on average, between 1% and 9% of the solutions suggested by NSGA-II, whereas no I-MCSGA solution is dominated by any solution generated by NSGA-II (Column 5). Besides, NSGA-II has very few solutions belonging to the *satisfactory* category, while all solutions of our method belong to the *satisfactory* category (Column 7).

Analysing the information contained in Table 5, we can observe that even though NSGA-II generates a larger number of solutions than I-MCSGA (Column 3), it fails to characterize the ROI, while our method always finds solutions belonging to the ROI.

**Table 5.** Comparative results between NSGA-II and I-MCSGA.

Instance	Algorithm	Average from 30 runs				
		Size of the solution set	Non-dominated solutions in $(A \cup B)^a$	% Non-dominated solutions in $(A \cup B)^a$	Solutions in the ROI*	% Solutions in the ROI*
1	NSGA-II	103	96	93%	0	0%
	I-MCSGA	50	50	100%	50	100%
2	NSGA-II	103	102	99%	0	0%
	I-MCSGA	10	10	100%	10	100%
3	NSGA-II	103	96	93%	0	0%
	I-MCSGA	54	53	98%	53	98%
4	NSGA-II	105	102	97%	1	1%
	I-MCSGA	5	5	100%	5	100%
5	NSGA-II	102	93	91%	5	5%
	I-MCSGA	10	10	100%	10	100%
6	NSGA-II	103	102	99%	2	2%
	I-MCSGA	4	4	100%	4	100%

<sup>a</sup>A and B are the solution sets generated by NSGA-II and I-MCSGA, respectively.

\*Solutions belonging to the approximated ROI (non-dominated and *satisfactory*)

#### 4.5 Second experiment: A<sup>2</sup>-NSGA-III vs I-MCSGA

The second experiment compares the quality of the solutions of A<sup>2</sup>-NSGA-III against those obtained from I-MCSGA, for problems with nine and sixteen objectives. We test for four random instances, three of which are problems with nine objectives and the last of which is a problem with sixteen objectives. The basic information of these instances is shown in Table 6. The number of reference points in A<sup>2</sup>-NSGA-III is taken from the recommendations of Deb and Jain in [44]. The population size and number of reference points used in both algorithms are shown in Table 7. The parameters used for automatic enhancement and for updating the reference set  $T_{upd}$  are set as described in Section 4.4.

**Table 6.** Information about instances used in the second experiment.

Instance	Instance Description	
	Objectives	Projects
1	9	100
2	9	100
3	9	100
4	16	500

**Table 7.** Population size and number of reference points used by both algorithms.

No. of objectives	Population size	No. of ref. points
9	174	174
16	136	136

The comparative results are summarized in Table 8. Instances 1–3 correspond to problems with nine objectives. It can be seen that our method on average dominates between 7% and 37% of the solutions suggested by A<sup>2</sup>-NSGA-III, while the solutions from I-MCSGA always remain as non-dominated (Column 5). There are no A<sup>2</sup>-NSGA-III solutions belonging to the *satisfactory* category, while our method always finds non-dominated solutions belonging to the *satisfactory* category (Column 7). Instance 4 concerns a problem with sixteen objective functions. We can see that the solutions from our method dominate all of the solutions suggested by A<sup>2</sup>-NSGA-III, whereas no I-MCSGA solution is dominated by any solution generated by A<sup>2</sup>-NSGA-III (Column 5). It is not surprising that no A<sup>2</sup>-NSGA-III solution belongs to the ROI, while our approach always finds *satisfactory* solutions (Column 7). These results give some evidence that our approach is effective in many-objective problems, providing a good characterization of the ROI.

**Table 8.** Comparative results between A<sup>2</sup>-NSGA-III and I-MCSGA.

Instance	Algorithm	Average from 30 runs			
		Size of the solution set	Non-dominated solutions in $(A \cup B)^a$	% Non-dominated solutions in $(A \cup B)^a$	Solutions in the ROI* % Solutions in the ROI*
1	A <sup>2</sup> -NSGA-III	198	124	63%	0 0%
	I-MCSGA	111	111	100%	111 100%
2	A <sup>2</sup> -NSGA-III	200	186	93%	0 0%
	I-MCSGA	11	11	100%	11 100%
3	A <sup>2</sup> -NSGA-III	199	154	77%	0 0%
	I-MCSGA	12	12	100%	12 100%
4	A <sup>2</sup> -NSGA-III	164	0	0%	0 0%
	I-MCSGA	192	192	100%	192 100%

<sup>a</sup>A and B are the solution sets generated by A<sup>2</sup>-NSGA-III and I-MCSGA, respectively.

\*Solutions belonging to the approximated ROI (non-dominated and *satisfactory*)

## 5. Conclusions and future work

We have presented an approach for incorporating the DM's preferences in an implicit and interactive way in solving optimization problems with few and many objectives. The interactive approach allows the DM to learn progressively about the problem and refines his/her preferences, leading the search towards a specific region of the solution space, known as the Region of Interest (ROI), instead of the whole Pareto front. Here, the solutions in the ROI are non-dominated and are considered *satisfactory* by the DM.

The approach was verified through a method that we named I-MCSGA. We simulated the DM by using a preference model, with the parameters for the model remaining unchanged during the optimization process. The DM's preferences are reflected by an outranking model and a reference set that is used by a multi-criteria sorting method (THESEUS) to assign new solutions to ordered categories. At a certain time, the solutions that were assigned as *satisfactory* by THESEUS during the search process are validated by a simulated DM to carry out the updating process of the preferences.

I-MCSGA was evaluated on project portfolio optimization problems, and we compared it with two methods from the literature. It was compared in relatively simple problems (three and four objectives) with NSGA-II and in many-objective problems (nine and sixteen objectives) with A<sup>2</sup>-NSGA-III. Our method showed better results than these methods with respect to Pareto dominance and also with respect to its capacity to reach the ROI.

The experimental results reveal that the elitism based on Pareto dominance combined with the category assignments performed by THESEUS help I-MCSGA to create selective pressure towards the ROI. The proposed automatic enhancement process is effective to incorporate new solutions into the reference set, helping THESEUS to suggest more appropriate assignments. Moreover, the suggested process of the interactive updating of preferences is effective in validating the solutions of the enhanced reference set, even when the real DM was replaced by a preference model. Therefore, the I-MCSGA, based on the proposed method, has shown its ability to characterize the ROI and to handle optimization problems with a few and many objectives efficiently.

Future research directions include extending the experimentation to problems where the true Pareto frontier is known, with the aim of validating the outcomes of our approach with greater certainty.

## Acknowledgments

This work has been partially supported by PRODEP and the following projects: a) CONACYT Project 236154 and b) Project 269890 from CONACYT networks.

## References

1. Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary algorithms for solving multi-objective problems. 2nd edn, Springer, New York (2007).
2. Fernandez, E., Lopez, E., Lopez, F., Coello, C.A.: Increasing selective pressure towards the best compromise in evolutionary multiobjective optimization: The extended NOSGA method. *Inform. Sciences*, 181(1), pp. 44–56 (2011).
3. Machin, M., Nebro, A.J.: Multiobjective Adaptive Metaheuristics. *Computación y Sistemas*, 17(1), 53-62 (2013).
4. Lagunas, J.R., Moo, V.M., Ortíz, B.: Two-Degrees-of-Freedom Robust PID Controllers Tuning Via a Multiobjective Genetic Algorithm. *Computación y Sistemas*, 18(2), 259-273 (2014).
5. Deb, K.: Current trends in evolutionary multi-objective optimization. *Int. J. Simul. Multi. Des. Optim.* 1(1), pp. 1–8 (2007).
6. Miller, G.A.: The magical number seven, plus or minus two: some limits on our capacity for processing information. *Psychol. Rev.* 63(2), pp. 81–97 (1956).
7. Deb, K., Sundar, J., Udaya Bhaskara Rao, N., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithms. *International Journal of Computational Intelligence Research*, 2(3), pp. 273–286 (2006).
8. Adra, S.F., Griffin, I., Fleming, P.J.: A comparative study of progressive preference articulation techniques for multiobjective optimisation. *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science 4403*, Springer-Verlag, pp. 908–921 (2007).
9. Bechikh, S.: Incorporating decision maker's preference information in evolutionary multi-objective optimization. PhD thesis, University of Tunis, Tunisia (2013).
10. Sanchez, P.J.: Modelos para la combinación de preferencias en toma de decisiones: herramientas y aplicaciones. PhD thesis, University of Granada, Spain (2007).
11. Branke, J., Deb, K.: Integrating user preferences into evolutionary multi-objective optimization. *Knowledge Incorporation in Evolutionary Computation, Studies in Fuzziness and Soft Computing*, Springer-Verlag, pp. 461–477 (2005).
12. Zitzler, E., Brockhoff, D., Thiele, L.: The hypervolume indicator revisited: On the design of Pareto-compliant indicators via weighted integration. *Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science*, Springer-Verlag, pp. 862–876 (2007).
13. Fowler, J.W., Gel, E.S., Köksalan, M.M., Korhonen, P., Marquis, J.L., Wallenius, J.: Interactive evolutionary multi-objective optimization for quasi-concave preference functions. *Eur. J. Oper. Res.* 206(2), pp. 417–425 (2010).
14. Deb, K., Sinha, A., Korhonen, P.J., Wallenius, J.: An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE T. Evol. Comput.* 14(5), pp. 723–739 (2010).
15. Cvetkovic, D., Parmee, I.C.: Preferences and their application in evolutionary multiobjective optimization. *IEEE T. Evol. Comput.* 6(1), pp. 42–57 (2002).

16. Rachmawati, L., Srinivasan, D.: Incorporating the notion of relative importance of objectives in evolutionary multiobjective optimization. *IEEE T. Evol. Comput.* 14(4), pp. 530–546 (2010).
17. Bechikh, S., Ben, S.L., Ghédira, K.: The r-dominance: a new dominance relation for preference-based evolutionary multi-objective optimization. Technical Report BS-2010-001, SOIE Research Unit, High Institute of Management of Tunis, Tunisia (2010).
18. Deb, K., Kumar, A.: Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Proceedings of the 9th annual conference on genetic and evolutionary computation*, ACM, USA, pp. 781–788 (2007).
19. Branke, J., Kaubler, T., Schmeck, H.: Guidance in evolutionary multi-objective optimization. *Adv. Eng. Softw.* 32(6), pp. 499–507 (2001).
20. Wagner, T., Trautmann, H.: Integration of preferences in hypervolume-based multiobjective evolutionary algorithms by means of desirability functions. *IEEE T. Evol. Comput.* 14(5), pp. 688–701 (2010).
21. Fernandez, E., Lopez, E., Bernal, S., Coello, C.A., Navarro, J.: Evolutionary multiobjective optimization using an outranking-based dominance generalization. *Comput. Oper. Res.* 37(2), pp. 390–395 (2010).
22. Oliveira, E., Antunes, C.H., Gomes, A.: A comparative study of different approaches using an outranking relation in a multi-objective evolutionary algorithm. *Comput. Oper. Res.* 40(6), pp. 1602–1615 (2013).
23. Köksalan, M., Mousseau, V., Özpeynirci, Ö., Özpeynirci, S.B.: A new outranking-based approach for assigning alternatives to ordered classes. *Nav. Res. Log.* 56(1), pp. 74–85 (2009).
24. Almeida-Dias, J., Figueira, J., Roy, B.: Electre Tri-C: a multiple criteria sorting method based on characteristic reference actions. *Eur. J. Oper. Res.* 204(3), pp. 565–580 (2010).
25. Hwang, C.L., Masud, A.S.M.: *Multiple Objective Decision Making-Methods and Applications*. Lecture Notes in Economics and Mathematical Systems 164, Springer-Verlag, (1979).
26. Deb, K., Kumar, A.: Light beam search based multi-objective optimization using evolutionary algorithms. *IEEE Congress on evolutionary computation (CEC-07)*, Singapore, pp. 2125–2132, (2007).
27. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE T. Evol. Comput.* 6(2), pp. 182–197 (2002).
28. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization. In *Proc. Evolutionary methods for design, optimization and control with applications to industrial problems*, pp. 95–100 (2002).
29. Ben Said, L., Bechikh, S., Ghédira, K.: The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. *IEEE T. Evolut. Comput.* 14(5), pp. 801–818 (2010).
30. Pedro, L.R., Takahashi, R.H.: INSPM: An interactive evolutionary multi-objective algorithm with preference model. *Inform. Sciences* 268, pp. 202–219 (2014).
31. Miettinen, K., Ruiz, F., Wierzbicki, A.P.: Introduction to multiobjective optimization: interactive approaches. In *Multiobjective Optimization*, Lecture Notes in Computer Science 5252, Springer Berlin Heidelberg, pp. 27–57 (2008).
32. Cruz, L., Fernandez, E., Gomez, C., Rivera, G., Perez, F.: Many-objective portfolio optimization of interdependent projects with ‘a priori’ incorporation of decision-maker preferences. *Appl. Math. Inf. Sci.* 8(4), pp. 1517–1531 (2014).
33. Roy, B.: *Multicriteria methodology for decision aiding, Nonconvex Optimization and Its Applications*. Springer (1996).
34. Roy, B.: The Outranking Approach and the Foundations of ELECTRE methods. In *Multiple Criteria Decision Aid*. Bana e Costa CA (eds), Springer Berlin, pp. 155–183 (1990).
35. Brans, J.P., Mareschal, B.: PROMETHEE methods. In *Multiple criteria decision analysis: state of the art surveys*, Springer, pp. 163–190 (2005).
36. Fernandez, E., Navarro, J.: A new approach to multi-criteria sorting based on fuzzy outranking relations: The THESEUS method. *Eur. J. Oper. Res.* 213(2), pp. 405–413 (2011).
37. Doumpos, M., Zopounidis, C.: *Multicriteria Decision Aid Classification Methods*. Kluwer Academic Publishers (2002).
38. Fernandez, E., Navarro, J., Salomon, E.: Automatic Enhancement of the Reference Set for Multi-Criteria Sorting in The Frame of Theseus Method. *Found. Comput. Decis. Sci.*, 39(2), pp. 57–77 (2014).
39. Fernandez, E., Navarro, J., Covantes, E., Rodriguez, J.: Analysis of the effectiveness of the THESEUS multi-criteria sorting method: Theoretical remarks and experimental evidence. *TOP*, pp. 1–26 (2016).
40. Fernandez, E., Navarro, J., Mazcorro, G.: Evolutionary multi-objective optimization for inferring outranking model’s parameters under scarce reference information and effects of reinforced preference. *Found. Comput. Decis. Sci.* 37(3), pp. 163–197 (2012).
41. Fernandez, E., Navarro, J., Duarte, A., Ibarra, G.: Core: A decision support system for regional competitiveness analysis based on multi-criteria sorting. *Decis. Support Syst.* 54(3), pp. 1417–1426 (2013).
42. Yuan, Y., Xu, H., Wang, B.: An improved NSGA-III procedure for evolutionary many-objective optimization. In *Proc. of the 2014 conference on Genetic and evolutionary computation*, USA, pp. 661–668 (2014).
43. Jain, H., Deb, K.: An improved adaptive approach for elitist nondominated sorting genetic algorithm for many-objective optimization. In *Evolutionary Multi-Criterion Optimization*, Springer Berlin Heidelberg, pp. 307–321 (2013).
44. Deb, K., Jain, H.: An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints. *IEEE T. Evol. Comput.* 18(4), pp. 577–601 (2014).