

# Preference Incorporation into Evolutionary Multiobjective Optimization using Preference Information Implicit in a Set of Assignment Examples

Laura Cruz-Reyes<sup>1</sup> Eduardo Fernandez<sup>2</sup> Rafael Olmedo<sup>2</sup>

Patricia Sanchez<sup>1</sup> Jorge Navarro<sup>2</sup>

<sup>1</sup>Instituto Tecnológico de Ciudad Madero, Cd. Madero, Mexico

<sup>2</sup>Universidad Autonoma de Sinaloa, Culiacan, Mexico

lauracruzreyes@itcm.edu.mx, jpatricia.sanchez@gmail.com, {eddyf, rafael, jnavarro}@uas.edu.mx

## Abstract

Nowadays, most approaches in the evolutionary multiobjective optimization literature concentrate mainly on adapting an evolutionary algorithm to generate an approximation of the Pareto frontier. However, this does not solve the problem. We present a new idea to incorporate into a *MOEA* the Decision Maker (*DM*) preferences, expressed in a set of solutions assigned to ordered categories. We modified the Non-dominated Sorting Genetic Algorithm 2 (*NSGA2*) to make selective pressure towards non-dominated solutions that belong to the most preferred category. In several instances, on the project portfolio problem, our proposal outperforms the standard *NSGA2*, achieving non-dominated solutions that best match the *DM*'s preferences.

**Keywords:** evolutionary algorithms; multiobjective optimization; preference incorporation; multicriteria sorting.

## 1. Introduction

Most real world optimization problems involve multiple criteria to be considered 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 of a multiobjective problem (*MOP*) cannot be reached. As was stated by Fernandez et al. ([3]), 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 one.

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 them to obtain an approximation of the Pareto frontier in a single algorithm's run. Thus, by using *MOEAs* the *DM* and/or the decision

analyst does not need to perform a set of separate single-objective optimizations (as normally required when using operations research methods) in order to generate compromise solutions. Additionally, *MOEAs* are more robust regarding the shape or continuity of the Pareto front, whereas these two issues are a real concern for operations research optimization methods (cf. [4]). These features are shared by other population-based multiobjective metaheuristics. However, according to Deb ([5]) and Fernandez et al. ([3, 6]), one aspect that is often disregarded in the *MOEAs*' literature is the fact that the solution of a problem involves not only the search, but also the decision making process. Most current approaches in the metaheuristic multiobjective optimization literature concentrate on adapting a metaheuristic algorithm to generate an approximation of the Pareto optimal set. Nevertheless, finding this set 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 2 or 3 objectives. However, as the number of criteria increases, two important difficulties arise ([3]):

- The algorithm's capacity to find this Pareto frontier quickly degrades (e.g. [2, 7]).
- The cardinal of a representative portion of the known Pareto frontier may be too large. 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 approaches from the field of multicriteria decision analysis do not perform well on such large decision problems, making difficult to obtain a unique solution.

In order to make the decision making phase easier, the *DM* would agree with incorporating his/her multicriteria preferences into the search process. This preference information is used to guide the search towards the Region Of Interest (*ROI*) ([8]), the privileged zone of the Pareto frontier that best matches the *DM*'s preferences.

The *DM*'s preference information can be expressed in different ways. According to Bechikh ([9]), the following are the most commonly used ones:

- Those in which importance factors (weights) are assigned by the *DM* to each objective function.
- Those in which the *DM* makes pair-wise comparisons on a subset of the current *MOEA*'s population in order to rank the sample's solutions.
- When pairwise comparisons between pairs of objective functions are performed in order to rank the set of objective functions.
- Those based on goals or aspiration levels to be achieved by each objective (reference point).
- When the *DM* supplies his/her aspiration levels and his/her reservation point, that is, the worst value he/she considers acceptable.
- When the *DM* identifies acceptable trade-offs between objective functions.
- When the *DM* supplies the model's parameters to build a fuzzy outranking relation.
- Construction of a desirability function that is based on the assignment of some desirability thresholds.

An explicit expression of preferences is used by the above methods. But preferences may be also implicitly given. Let us suppose that a set of solutions has been sorted by the *DM* on a set of ordered categories like {Very Good, Good, Acceptable, No Good, Bad}. These assignments express the *DM*'s assignment policy. So, the set of assignments can be considered preference knowledge. Using this knowledge in the framework of a multicriteria sorting method, each new solution generated by the search process can be assigned to one of those categories. To some extent, such preference information surrogates the *DM* in judging new solutions. A way of using this capacity to make selective pressure towards the *ROI* is needed in order to complete this original approach.

The paper is structured as follows: some background is given in Section 2. Our proposal is described in Section 3, followed by numerical experiments in Section 4. Finally, we draw concluding remarks in the last section.

## 2. Background

### 2.1. Notation and premises

Let us consider a *MOP* of the form:

$$\text{Max } F = (f_1(z), f_2(z), \dots, f_n(z)) \quad (1)$$

$$z \in R_F$$

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

Let us denote by  $O$  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.

Some premises follow:

i) The *DM* can define a finite set of ordered categories  $Ct = \{C_1, \dots, C_M\}$ , which he/she judges adequate for representing the quality of the solutions;  $C_M$  is assumed to be the preferred category.

ii) The *DM* agrees with certain decision policy defined on a subset  $O'$  of  $O$ . According to this policy, there should be a certain function  $G: O' \rightarrow Ct$  such that for each  $x \in O'$  the *DM* accepts that  $G(x)$  is the most appropriate assignment of  $x$ .  $G$  expresses the assignment policy from the *DM*.

iii) The *DM* is able to provide a set of reference objects or training examples  $T$ , which is composed of elements  $b_{k,h} \in O'$  assigned to category  $C_k$ , ( $k=1, \dots, M$ ).

iv) The *DM*'s multi-criteria preferences can be aggregated in a fuzzy outranking relation  $\sigma(x, y)$  defined on a representative subset of  $O \times O$ . Its value models the truth value of the predicate 'x is at least as good as y' from the *DM*'s perspective.  $\sigma$  can be set by using outranking multicriteria methods (e.g. [10]). The parameter setting of  $\sigma$  can be directly performed by the *DM* (probably in a constructive framework collaborating with a decision analyst), or be inferred by using an indirect elicitation method ([11, 12]).

Let us consider a real value  $\lambda > 0.5$ .

**Definition 1:** The following crisp binary relations are given on a representative subset of  $O$ :

$(x, y) \in S(\lambda)$  iff  $\sigma(x, y) \geq \lambda$  ( $\lambda$ -outranking)

$(x, y) \in P(\lambda)$  iff  $\sigma(x, y) \geq \lambda \wedge \sigma(y, x) < 0.5$  ( $\lambda$ -strict preference)

$(x, y) \in Q(\lambda)$  iff  $\sigma(x, y) \geq \lambda \wedge 0.5 \leq \sigma(y, x) < \lambda$  ( $\lambda$ -weak preference)

$(x, y) \in I(\lambda)$  iff  $\sigma(x, y) \geq \lambda \wedge \sigma(y, x) \geq \lambda$  ( $\lambda$ -indifference).

$(x, y) \in R(\lambda)$  iff  $\sigma(x, y) < \lambda \wedge \sigma(y, x) < \lambda$  ( $\lambda$ -incomparability).

### 2.2. A brief reminder of the *THESEUS* method

Proposed by Fernandez and Navarro ([13]), the *THESEUS* method is based on comparing a new object to be assigned with reference objects through models of preference and indifference relations. The assignment is not a consequence of the object intrinsic properties; it is rather the result of comparisons with other objects whose assignments are known. In the following  $C(x)$  denotes a potential assignment of object  $x$  and  $C(b)$  is the actual assignment of  $b$ . According to *THESEUS*,  $C(x)$  should satisfy:

$$\forall x \in U, \forall b \in T$$

$$xP(\lambda)b \Rightarrow C(x) \succeq C(b) \quad (2.a)$$

$$bP(\lambda)x \Rightarrow C(b) \succeq C(x)$$

$$xQ(\lambda)b \Rightarrow C(x) \succeq C(b) \quad (2.b)$$

$$bQ(\lambda)x \Rightarrow C(b) \geq C(x)$$

$$\begin{aligned} xI(\lambda)b &\Rightarrow C(x) \geq C(b) \wedge C(b) \geq C(x) \\ &\Leftrightarrow C(x) = C(b) \end{aligned} \quad (2.c)$$

Note that  $C(x)$  is a variable whose domain is the set of ordered categories. Equations (2.a-c) express the necessary consistency amongst the preference model, the reference set and the appropriate assignments of  $x$ . The assignment  $C(x)$  should be as compatible as possible with the current knowledge about the assignment policy.

*THESEUS* uses the inconsistencies with Equations (2.a-c) to compare the possible assignments of  $x$ . More specifically:

- The set of  $P(\lambda)$ -inconsistencies for  $x$  and  $C(x)$  is defined as  $D_P = \{(x,b), (b,x), b \in T \text{ such that } (2.a) \text{ is FALSE}\}$ .
- The set of  $Q(\lambda)$ -inconsistencies for  $x$  and  $C(x)$  is defined as  $D_Q = \{(x,b), (b,x) \mid b \in T \text{ such that } (2.b) \text{ is FALSE}\}$ .
- The set of  $I(\lambda)$ -inconsistencies for  $x$  and  $C(x)$  is defined as  $D_I = \{(x,b), b \in T \text{ such that } (2.c) \text{ is FALSE}\}$ .

Some  $I(\lambda)$ -inconsistencies might be explained by ‘discontinuity’ of the description based on the set of categories. The cases in which  $xI(\lambda)b \wedge |k-j|=1$  will be called second-order  $I(\lambda)$ -inconsistencies and grouped in the set  $D_{2I}$ . The set  $D_{1I} = D_I - D_{2I}$  contains the so-called first-order  $I(\lambda)$ -inconsistencies, which are not consequences of the described discontinuity effect. Let  $n_P, n_Q, n_{1I}, n_{2I}$  denote the cardinality of the above-defined inconsistency sets, and  $N_1 = n_P + n_Q + n_{1I}, N_2 = n_{2I}$ .

*THESEUS* suggests an assignment that minimizes the above inconsistencies with lexicographic priority favoring  $N_1$ , which is the most important criterion (cf. [13]). The basic assignment rule is:

- a. Assign the minimum credibility level  $\lambda > 0.5$ .
- b. Starting with  $k = 1$  ( $k=1, \dots, M$ ) and considering each  $b_{k,h} \in T$ , calculate  $N_1(C_k)$ .
- c. Identify the set  $\{C_j\}$  whose elements hold  $C_j = \operatorname{argmin} N_1(C_k)$ .
- d. Select  $C_{k^*} = \operatorname{argmin}_{\{C_j\}} N_2(C_j)$ .
- e. If  $C_{k^*}$  is a single solution, assign  $x_j$  to  $C_{k^*}$ ; other situations are approached below.

The suggestion may be a single category or a sequence of categories. The first case is called a well-defined assignment; otherwise, the obtained solution highlights the highest category ( $C_H$ ) and the lowest category ( $C_L$ ) which is appropriate for assigning the object, but fails in determining the most appropriate.

### 3. The MultiCriteria Sorting Genetic Algorithm (MCSGA)

In a *MOP* the *ROI* should be composed of solutions that belong to the most preferred category  $C_M$ . So, the *ROI* is characterized by two features: i) to be non-dominated; ii) to be assigned to  $C_M$ . The simple idea behind this new proposal consists in making selective pressure towards solutions that hold both properties.

We propose to use a variant of the popular *NSGA2*, which is called here the MultiCriteria Sorting Genetic Algorithm. It works like the *NSGA2*, with the following differences:

- a. Each solution of the first front of *NSGA2* (the non-dominated front) is assigned by *THESEUS* to one category of the set  $C_T$ ;
- b. The first front of *NSGA2* is divided in  $M \leq M$  sub-fronts; the first ranked sub-front contains the solutions that were assigned to the most preferred category;
- c. The fronts of the current *NSGA2* population are re-ordered by considering each sub-front of the original non-dominated front as a new front;
- d. The same operations of *NSGA2* are applied, but considering the new fronts; particularly, the *NSGA2* elitism concerns the new first front, which is now composed of non-dominated solutions belonging to the most preferred category.

The basic procedure is shown below:

```

PROCEDURE MCSGA (L, Number_of_Generations)
Set preference information
  Initialize Reference set T
  Initialize  $\sigma$ - parameters
Initialize Population P
Generate random population with size L
Evaluate objective values
Generate non-dominated fronts on P
Assign to these fronts a rank (level)
Calculate  $\sigma$  on  $R_1 \times R_1$ 
For each  $x \in R_1$ , assign  $x$  to one preference category
Form  $M'$  sub-fronts of  $R_1$ 
Assign to these sub-fronts a rank (level) and
update the levels of the remaining fronts
Generate Child Population Q with size L
  Perform Binary Tournament Selection
  Perform Recombination and mutation
FOR I = 1 to Number_of_Generations DO
  Assign  $P' = P \cup Q$ 
  Generate non-dominated fronts on  $P'$ 
  Assign to these fronts a rank (level)
  Calculate  $\sigma$  on  $R_1 \times R_1$ 
  For each  $x \in R_1$ , assign  $x$  to one preference category
  Form  $M'$  sub-fronts of  $R_1$ 
  Assign to these sub-fronts a rank (level) and
  update the levels of the remaining fronts
  FOR each parent and child in  $P'$  DO
    Calculate crowding distance
    Loop (inside) by adding solutions to the next generation
    until L individuals have been found
  End FOR
  Replace P by the L individuals found
  Generate Child Population Q with size L
  Perform Binary Tournament Selection

```

```

Perform Recombination and mutation
End FOR
End PROCEDURE

```

Remarks:

In order to form the reference set, the *DM* should evaluate several existing or fictitious solutions. This information can be used by an indirect elicitation method to infer the outranking model's parameters.

Some inconvenience could arise when *T* is populated with fictitious solutions. In the beginning of the search process, the *DM* is usually unable to identify good solutions in the objective space with pre-image in the feasible decision variable space. In this starting phase the *DM* does not know even a rough approximation to his/her *ROI*. So, he/she can hardly identify feasible solutions belonging to the most preferred categories. Hence, the selective pressure towards the *ROI* is weak.

To overcome this drawback, we propose to hybridize the above *MCSGA*. This will be combined with another multiobjective metaheuristic approach, which provides an approximation to the Pareto frontier. The *DM*'s preferences will be expressed on a subset of that approximation, *T* being thus constructed. Once *T* has been initialized, the outranking model's parameters can be inferred by an indirect elicitation method ([11]). The hybrid procedure is described below:

```

PROCEDURE H-MCSGA (L, Number_of_Generations)
Run a multiobjective optimization method and
obtain an approximation to the Pareto frontier PF
  Initialize Reference set T by using a subset of PF
  Set  $\sigma$ -parameters agreeing with T
Initialize Population P
Generate random population with size L
Evaluate objective values
Generate non-dominated fronts on P
Assign to these fronts a rank (level)
Calculate  $\sigma$  on  $R1 \times R1$ 
For each  $x \in R1$ , assign  $x$  to one preference category
Form  $M'$  sub-fronts of  $R1$ 
Assign to these sub-fronts a rank (level) and
update the levels of the remaining fronts
Generate Child Population Q with size L
  Perform Binary Tournament Selection
  Perform Recombination and mutation
FOR I = 1 to Number_of_Generations DO
  Assign  $P' = P \cup Q$ 
  Generate non-dominated fronts on  $P'$ 
  Assign to these fronts a rank (level)
  Calculate  $\sigma$  on  $R1 \times R1$ 
  For each  $x \in R1$ , assign  $x$  to one preference category
  Form  $M'$  sub-fronts of  $R1$ 
  Assign to these sub-fronts a rank (level) and
  update the levels of the remaining fronts
  FOR each parent and child in  $P'$  DO
    Calculate crowding distance
    Loop (inside) by adding solutions to the next generation
    until L individuals have been found
  End FOR
  Replace P by the L individuals found
  Generate Child Population Q with size L
    Perform Binary Tournament Selection
    Perform Recombination and mutation
End FOR
End PROCEDURE

```

## 4. Several computer experiments

### 4.1. Multiobjective optimization of a portfolio's measures derived from a project ranking

Let us suppose that in the first phase of the process low-quality project proposals are discarded, and that the ranked projects are acceptable proposals for the funding organization.

The information provided by the simple project ranking is very poor for portfolio optimization purposes. It is necessary to compare the quality of the possible portfolios in order to find the best one. This problem was approached by Fernandez and Olmedo ([14]) under the assumption that "the portfolio impact on a decision maker's mind is determined by the number of supported projects and their particular rank". If project *a* is clearly ranked better than *b*, then *a* is admitted to have "more social impact" than *b*. The *DM* should take this information from the ranking into account. The appropriateness of a portfolio is not only defined by the quality of the included projects, but also by the amount of contained projects. The purpose is to build a good portfolio by increasing the number of supported projects and controlling the possible disagreements with respect to the *DM*'s preferences, which, it is assumed, are incorporated into the input ranking. Some discrepancies may be acceptable between the information provided by the ranking and the decisions concerning the approval (hence supporting) of projects, whenever this fact increases the number of projects in the portfolio. However, this inclusion should be controlled because the admission of unnecessary discrepancies is equivalent to underestimating the ranking information. A multiobjective optimization problem is solved by using *NSGA2*, in which the objective functions are the number of supported projects and the number of discrepancies (separately in several functions, in regard to the importance of each kind of discrepancy).

Fernandez and Olmedo ([14]) define:

- A set of absolute discrepancies  $D_a = \{ (a,b) \in P_r \times P_r \text{ such that } a \gg b, b \in C \text{ and } a \notin C \}$
- A set of strict discrepancies  $D_s = \{ (a,b) \in P_r \times P_r \text{ such that } a > b, b \in C \text{ and } a \notin C \}$
- A set of weak discrepancies  $D_w = \{ (a,b) \in P_r \times P_r \text{ such that } a \succsim b, b \in C \text{ and } a \notin C \}$

where

*C* is a portfolio;

$P_r$  is the set of projects

$a \gg b$  means absolute preference according to the relative ranks of projects *a,b*;

$a > b$  means strict preference according to the relative ranks of projects *a,b*;

$a \succsim b$  denotes weak preference.

Let  $n_a$ ,  $n_s$ ,  $n_w$  denote the respective cardinality of the above sets. Considering also the number of supported projects in *C* (denoted by  $n_{pr}$ ), Fernandez and Olmedo ([14]) propose to solve the problem:

$$\text{Max } n_{pr}, \text{Min } (n_a, n_s, n_w) \quad (3)$$

$$C$$

As illustration, let us consider the problem of distributing 2.5 billion dollars among 100 projects, all of them deserving individual financing. Information about these projects (costs in million dollars and their ranks according to some preference ordering) is shown in Table 1.

The NSGA2 was used as the first step of the hybrid MCSGA. Binary encoding was used; a '1' in the individual  $m$ -th allele means that the  $m$ -th project belongs to this particular portfolio. One-point crossover and the standard mutation operator were implemented (cf. [15]). Binary tournament selection was performed as suggested by [1].

The approach for handling constraints is based on the principle that any individual satisfying the constraints is

always better than any individual that does not. If the feasible individuals in the population are classified into  $k$  non-dominated classes, then any unfeasible individual is relegated to class  $k+1$ . This implies that in binary tournaments any feasible individual shall have priority over any non-feasible individual.

The parameters of the evolutionary search were: cross-over probability = 1; mutation probability = 0.02; population size = 100, number of generations= 500.

A representative sample of the Pareto front is shown in Table 2. Solution 63  $(n_{pr}, n_a, n_s, n_w) = (22, 0, 0, 0)$  corresponds to allocating funds according to the ranking. If the DM is willing to accept some minor discrepancies, he/she can select the solutions  $(n_{pr}, n_a, n_s, n_w) = (25, 0, 16, 0)$ , or  $(24, 0, 13, 0)$ .

Table 1. Information about projects

Position in Ranking	Requirements (million dollars)	Position in Ranking	Requirements (million dollars)	Position in ranking	Requirements (million dollars)	Position in ranking	Requirements (million dollars)
1	84.00	26	31.25	51	27.50	76	46.50
2	124.50	27	26.50	52	41.25	77	44.00
3	129.75	28	36.25	53	29.50	78	25.75
4	147.75	29	50.00	54	25.25	79	38.25
5	126.00	30	34.75	55	40.00	80	40.75
6	137.25	31	48.25	56	30.75	81	42.75
7	96.00	32	46.00	57	39.00	82	43.00
8	84.75	33	36.75	58	44.50	83	32.25
9	93.00	34	34.00	59	47.50	84	37.75
10	121.50	35	26.00	60	36.00	85	44.75
11	102.75	36	31.75	61	28.50	86	27.00
12	141.75	37	29.75	62	29.00	87	39.50
13	105.75	38	37.25	63	30.25	88	30.00
14	98.25	39	26.75	64	49.50	89	37.50
15	101.25	40	43.75	65	33.00	90	49.00
16	83.25	41	27.25	66	38.50	91	41.75
17	109.50	42	47.00	67	33.50	92	39.25
18	107.25	43	41.00	68	48.50	93	34.50
19	135.00	44	30.50	69	35.00	94	49.75
20	97.50	45	45.25	70	28.75	95	48.00
21	127.50	46	26.25	71	25.50	96	29.25
22	114.00	47	45.50	72	40.25	97	47.75
23	106.50	48	44.25	73	38.75	98	42.25
24	94.50	49	48.75	74	46.75	99	46.25
25	43.50	50	33.25	75	37.00	100	39.75
						Total	5542.00

Table 2. Some Pareto solutions

Solution	$(n_{pr}, n_a, n_s, n_w)$	Solution	$(n_{pr}, n_a, n_s, n_w)$
63	(22, 0, 0, 0)	117	(31, 0, 78, 0)
91	(24, 0, 13, 0)	107	(32, 0, 86, 0)
93	(25, 0, 16, 0)	286	(33, 0, 103, 0)
100	(26, 0, 27, 0)	198	(34, 3, 114, 0)
110	(26, 0, 27, 0)	25	(34, 6, 112, 0)
112	(27, 0, 31, 0)	123	(35, 6, 129, 0)
191	(28, 0, 44, 0)	400	(36, 12, 156, 1)
114	(29, 0, 57, 0)	154	(37, 16, 170, 0)
252	(30, 0, 70, 0)	155	(38, 24, 182, 2)
48	(31, 0, 78, 0)	30	(37, 25, 159, 1)

In order to construct the reference set, the DM expresses his/her preferences through the set of categories {Good, Probably Good, No Good, Bad, Very Bad}. It is shown in Table 3.

Note that the reference set contains five fictitious solutions, which are useful for providing the DM's tradeoffs.

Applying the indirect parameter elicitation method from ([11]), we obtained:

- Weights = (0.210, 0.608, 0.122, 0.060)
- Indifference thresholds = (0, 0, 0, 0)
- Strict preference thresholds = (1, 1, 1, 2)
- Pre-veto thresholds = (1, 1.818, 2.95, 3.96)
- Veto thresholds = (3.006, 2.550, 4.880, 5.996)
- The credibility threshold  $\lambda = 0.67$ .

These parameter values are used to calculate the degree of credibility of outranking according to ELECTRE III (cf. [10]), with the simplification introduced by Mousseau and Dias ([16]).

Table 3. The reference set

Id	$n_{pr}$	$n_a$	$n_s$	$n_w$	Category
1	22	0	0	0	Probably Good
2	24	0	13	0	Good
3	25	0	16	0	Good
4	26	0	27	0	Probably Good
5	27	0	31	0	Probably Good
6	28	0	44	0	No Good
7	29	0	57	0	No Good
8	30	0	70	0	No Good
9	31	0	78	0	No Good
10	34	3	114	0	Bad
11	34	6	112	0	Very Bad
12	35	6	129	0	Very bad
13	23	0	0	2	Good
14	24	0	0	15	Good
15	25	0	0	20	Good
16	23	1	0	0	No Good
17	23	2	0	0	Bad

Running the second phase of the  $H-MCSGA$  the following solutions are identified as shown by Table 4:

Table 4. The ROI

Id	$(n_{pr}, n_a, n_s, n_w)$	Category
1	(25, 0, 12, 0)	Good
2	(24, 0, 11, 0)	Good



Figure 1. The ROI represented in the decision variable space

#### 4.2. Multiobjective optimization of project portfolios described by many criteria

Let us consider a decision making situation in which the  $DM$  is choosing among  $L$  different public policies (projects) each with a direct social impact. This is measured by using a  $K$ -dimensional vector  $(N_1, N_2, \dots, N_K)$ .  $N_i = n_{kj}$  denotes the number of people belonging to the  $k$ -th social category (e.g. Extreme Poverty, Poverty, Middle) who receive the  $j$ -th benefit level (e.g. High Impact, Middle Impact, Low Impact) from that policy or project. In the following  $N_i^m$  denotes the value of  $N_i$  associated to the  $m$ -th project. The project quality is measured as the number of beneficiaries for each of  $i$  criteria established previously.  $C'$  denotes a portfolio (a subset of the  $L$  projects which receives financial support). The value of  $N_i$  for the whole portfolio is  $N_i(C') = x_1 N_i^1 + \dots + x_L N_i^L$  where  $x_j = 1$  if the  $j$ -th project is supported and  $x_j = 0$ , otherwise. The aim of this decision problem is to choose the “best” portfolio satisfying some budget constraints. Formally, the problem is:

$$\begin{aligned} \text{Max } (N_1(C'), N_2(C'), \dots, N_K(C')) \\ C' \in R_F \end{aligned} \quad (4)$$

Note that these solutions dominate those previously considered Good by the  $DM$ . Represented in the decision variable space, the above solutions are shown in Figure 1.

Analyzing the chromosome structure, Solution 1 should be preferred since it contains slightly more and higher ranked projects.

where  $R_F$  is a feasible region determined by budget constraints. There is a total budget that the organization is willing to invest, which is denoted as  $B$ , and each project proposal  $l$  has an associated cost to carry it out, denoted as  $c_l$ , it is understood that the formation of any portfolio  $C'$  is subject to the constraint:

$$\sum_{l=1}^L x_l c_l \leq B \quad (5)$$

Other budget constraints are imposed by the area of project (educational, health, etc.), and its geographic region. In either case, the budget is bounded by a lower and upper limit. In this problem the only accepted solutions are those that satisfy the constraints.

Let us consider, as an illustration to the application of  $H-MCSGA$ , the problem of distributing 2.5 billion dollars in a set of applicant projects. This set is composed of 100 proposals and nine objectives each, all of which deserve financing. Information about some projects (values of objectives, costs, area, and the region) is shown in Table 5.

In the first phase of the  $H-MCSGA$ , the Ant Colony Optimization Algorithm (ACO) by Rivera et al. ([17]) was

used. This algorithm incorporates preferences by using the outranking model from ([3]). The *ACO* parameters were the same as those reported in the article, but we did

not use local search or consider synergy. The solutions were categorized by simulating a *DM* whose preferences are compatible with the outranking model from ([3]).

Table 5. Information about some projects

Project	$N_1$	$N_2$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	Cost	Area	Region
1	42890	8915	25565	13685	28980	15900	32300	13500	67295	9450	1	2
2	47565	24220	35845	11540	50315	14320	37085	21950	49010	6360	3	2
3	35435	18125	31280	23450	32650	24875	64150	18000	48310	5890	2	1
4	11010	32855	43650	13995	37600	28755	29025	37485	29625	7820	3	2
5	23030	31265	29890	26510	66610	24140	78980	37430	15740	9555	2	2

The model's parameters were set as follows:

Weights: (10, 13, 10, 12, 7, 13, 10, 7, 18).

Veto thresholds = (60000, 45000, 75000, 50000, 84000, 60000, 100000, 78000, 79000).

Indifference thresholds = (7500, 6000, 9000, 6000, 12000, 7500, 12000, 10500, 10500).

The credibility threshold  $\lambda = 0.67$ .

Examples of solutions included in a reference set are shown in Table 6. It is noteworthy that the number of elements of the most preferred category in the reference set is very small.

The classified solutions were processed in order to construct a consistent reference set that satisfies Equations (2.a-c). In cases of poorly populated reference sets, it is necessary to add *fictitious solutions* (derived from an ex-

isting solution). These solutions are used to extend and intensify the reference set when some category is poorly represented.

The procedure to generate them is as follows: search a pair of objectives with similar weights, choose a solution and at the corresponding objectives add and subtract the same amount. The way in which fictitious solutions are generated is pointed-out in Table 7.

We experimented with five random instances, creating for each of them one distinct reference set. Every instance was run ten times, and the parameters of the evolutionary search in the second phase of the *H-MCSGA* were: crossover probability = 1; mutation probability = 0.05; population size = 100, number of generations = 500.

The second phase of the *H-MCSGA* was run, and the obtained solutions were compared with the solutions from the standard *NSGA2*. The results are illustrated in Table 8.

Table 6. Some *ACO* solutions included in the reference set

Portfolio	$N_1$	$N_2$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	Category
1	1247615	884570	1396870	914930	1689460	1147495	1773720	1421840	1520465	Bad
2	1303565	1000930	1547800	949515	1684295	1260020	2008440	1399745	1623460	Bad
66	1320410	1008765	1612695	977740	1696005	1317720	2046040	1449645	1782705	No Good
67	1341340	976635	1579975	1021985	1731610	1337660	2076355	1464060	1695565	No Good
98	1346255	1013065	1563525	1012980	1678160	1292620	2089570	1432275	1820135	Probably Good
99	1327040	1014050	1596565	982735	1681330	1328230	2095395	1464250	1793820	Probably Good
100	1334615	1038825	1616815	976525	1714145	1322735	2099215	1474210	1775945	Good
101	1333585	1001165	1569390	1000915	1705255	1307305	2098165	1499430	1807120	Good

Table 7. Real and fictitious solutions

Reference element	$N_1$	$N_2$	$N_3$	$N_4$	$N_5$	$N_6$	$N_7$	$N_8$	$N_9$	Category
12	<b>1320410</b>	1008765	<b>1612695</b>	977740	1696005	1317720	2046040	1449645	1782705	No Good (real)
13	<b>1320910</b>	1008765	<b>1612195</b>	977740	1696005	1317720	2046040	1449645	1782705	No Good (fictitious)

Table 8. Comparative results between *NSGA2* and *H-MCSGA*

Instance	Instance description	Algorithm	Average		Highest Category	Highest Category Frequency	Highest Net Flow Frequency
			Size of the solution set	Solutions that remain Non-dominated in $A \cup B$			
1	9 objectives	<i>NSGA2</i>	117	70	3	100%	0%
	100 projects	<i>H-MCSGA</i>	54	54	4	100%	100%
2	9 objectives	<i>NSGA2</i>	120	76	3	100%	0%
	100 projects	<i>H-MCSGA</i>	24	24	4	100%	100%
3	9 objectives	<i>NSGA2</i>	124	70	3	100%	0%
	100 projects	<i>H-MCSGA</i>	54	54	4	100%	100%
4	9 objectives	<i>NSGA2</i>	115	37	3	90%	0%
	150 projects	<i>H-MCSGA</i>	72	72	4	100%	100%
5	9 objectives	<i>NSGA2</i>	112	63	3	100%	0%
	150 projects	<i>H-MCSGA</i>	25	25	4	100%	100%

Note: A is the set of solutions obtained by *NSGA2*; B is the set obtained by *H-MCSGA*

We can see a better performance of the *H-MCSGA*. Our proposal satisfies the *ROI* conditions, that is, it always finds non-dominated solutions of the most preferred category. Many solutions from the *NSGA2* are dominated, and there is no *NSGA2* solution belonging to the best category. Also, we used the outranking net flow score to rank the set  $A \cup B$ . The net flow is a very popular measure defined as (cf. [18])

$$F_n(a) = \sum_{c \in A - \{a\}} [\sigma(a, c) - \sigma(c, a)]$$

As can be seen in the last column of Table 8, the *H-MCSGA* always obtains solutions with the highest net flow, confirming a better convergence towards the *ROI*.

The results also reveal that our proposal provides a better characterization of the most preferred category with respect to the reference set used, which was obtained by ACO. This characteristic is measured in terms of the number of solutions contained in the best category.

## 5. Conclusions

An original idea to incorporate preferences into multi-objective evolutionary optimization has been presented. To the best of our knowledge this is the first time in which an implicit way of modeling the *DM*'s preferences has been suggested.

Our proposal is basically a derivation from the standard *NSGA2* but making selective pressure towards the Region of Interest instead the Pareto front. The *DM*'s preferences are captured in a reference set of solutions assigned to ordered categories. Using the *THESEUS* multi-criteria sorting method, a sort of "assignment machine" is built. Thus, good solutions are identified by the algorithm, and a convenient selective pressure towards the *ROI* is performed.

Unlike Pareto-based evolutionary algorithms, the number of "good" solutions in the earlier populations does not depend on the dimension of the objective space. Therefore, the selective pressure is not degraded with the number of objective functions.

In examples with 9 objectives, this proposal improves solutions obtained by *NSGA2*. Besides, we obtained a better characterization of the most preferred category with respect to the solutions given by an ant colony algorithm with "a priori" incorporation of preferences. More experimentation is required to reach definitive conclusions.

## References

- [1] K. Deb, "Multi-Objective Optimization using Evolutionary Algorithms," *John Wiley & Sons*, Chichester-New York-Weinheim-Brisbane-Singapore-Toronto, 2001.
- [2] C. Coello, G. Lamont, and Van Veldhuizen, D.A., "Evolutionary Algorithms for Solving Multi-Objective Problems," *Second Edition, Springer*, New York, 2007.
- [3] E. Fernandez, E. Lopez, F. Lopez, and C. Coello, "Increasing selective pressure toward the best compromise in Evolutionary Multiobjective Optimization: the extended NOSGA method", *Information Sciences*, pp. 44-56, 181, 2011.
- [4] C. Coello, "A comprehensive survey of Evolutionary-Based Multiobjective Optimization techniques," *Knowledge and Information Systems. An International Journal* 1 (3), pp. 269-308, 1999.
- [5] K. Deb, "Current trends in Evolutionary Multi-objective Optimization," *International Journal for Simulation and Multidisciplinary Design Optimisation*, 1 (1), pp. 1-8, 2007.
- [6] E. Fernandez, E. Lopez, S. Bernal, C. Coello, and J. Navarro, "Evolutionary Fernandez multiobjective optimization using an outranking-based dominance generalization," *Computers & Operations Research*, 37 (2), pp. 390-395, 2010.
- [7] Y. Wang, and Y. Yang, "Particle swarm optimization with preference order ranking for multi-objective optimization," *Information Sciences*, 179 (12), pp. 1944-1959, 2009.



- [8] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization", *Evolutionary Computation*, 14(4), pp. 463–494, 2006.
- [9] S. Bechikh, "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>, 2013.
- [10] B. Roy, "The Outranking Approach and the Foundations of ELECTRE methods," In: Bana e Costa, C.A. (ed.), *Reading in Multiple Criteria Decision Aid*, Springer-Verlag, Berlin, pp. 155-183, 1990.
- [11] E. Fernandez, J. Navarro, and G. Mazcorro, "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*, vol. 37, no. 3, pp. 163-197, 2012.
- [12] M. Doumpos, Y. Marinakis, M. Marimaki, and C. Zopounidis, "An evolutionary approach to construction of outranking models for multicriteria classification: The case of ELECTRE TRI method," *European Journal of Operational Research*, 199 (2), pp. 496-505, 2009.
- [13] E. Fernandez, and J. Navarro, "A new approach to multicriteria sorting problems based on fuzzy outranking relations: The THESEUS method," *European Journal of Operational Research*, vol. 213, pp.405-413, 2011.
- [14] E. Fernandez and R. Olmedo, "Public project portfolio optimization under a participatory paradigm", *Applied computational intelligence and soft computing*, vol. 2013, Article ID 891781, 13 pages, doi:10.1155/2013/891781, 2013.
- [15] Z. Michalewicz, "Genetic Algorithms + Data Structures = Evolution Programs", Springer Verlag, Berlin-Heidelberg-New York, 1996.
- [16] V. Mousseau, L. Dias, "Valued outranking relations in ELECTRE providing manageable disaggregation procedures", *European Journal of Operational Research*, 156-2, pp. 467–482, 2004.
- [17] G. Rivera, C. Gómez, E. Fernández, L. Cruz, O. Castillo and S. Bastiani, "Handling of Synergy into an Algorithm for Project Portfolio Selection", *Book Recent Advances on Hybrid Intelligent Systems*, pp. 417-430, 2013.
- [18] J. Fodor, M. Roubens, "Fuzzy Preference Modeling and Multicriteria Decision Support", Kluwer, Dordrecht, 1994.