

**This is an electronic reprint of the original article.  
This reprint *may differ* from the original in pagination and typographic detail.**

**Author(s):** Ruiz, Ana B.; Luque, Mariano; Miettinen, Kaisa; Saborido, Rubén

**Title:** An Interactive Evolutionary Multiobjective Optimization Method: Interactive WASF-GA

**Year:** 2015

**Version:**

**Please cite the original version:**

Ruiz, A. B., Luque, M., Miettinen, K., & Saborido, R. (2015). An Interactive Evolutionary Multiobjective Optimization Method: Interactive WASF-GA. In A. Gaspar-Cunha, C. H. Antunes, & C. C. Coello (Eds.), *Evolutionary Multi-Criterion Optimization : 8th International Conference, EMO 2015, Guimarães, Portugal, March 29 --April 1, 2015. Proceedings, Part II* (pp. 249-263). Springer. Lecture Notes in Computer Science, 9019. [https://doi.org/10.1007/978-3-319-15892-1\\_17](https://doi.org/10.1007/978-3-319-15892-1_17)

All material supplied via JYX is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of the repository collections is not permitted, except that material may be duplicated by you for your research use or educational purposes in electronic or print form. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone who is not an authorised user.

# An Interactive Evolutionary Multiobjective Optimization Method: Interactive WASF-GA

Ana B. Ruiz<sup>1</sup>, Mariano Luque<sup>1</sup>, Kaisa Miettinen<sup>2</sup>, and Rubén Saborido<sup>3</sup>

<sup>1</sup> Department of Applied Economics (Mathematics), Universidad de Málaga (Spain)  
abruiz@uma.es, mluque@uma.es

<sup>2</sup> University of Jyväskylä, Department of Mathematical Information Technology  
(Finland)  
kaisa.miettinen@jyu.fi

<sup>3</sup> Polytechnique Montréal Researchers in Software Engineering, École Polytechnique  
de Montréal (Canada)  
ruben.saborido-infantes@polymtl.ca

**Abstract.** In this paper, we describe an interactive evolutionary algorithm called *Interactive WASF-GA* to solve multiobjective optimization problems. This algorithm is based on a preference-based evolutionary multiobjective optimization algorithm called WASF-GA. In Interactive WASF-GA, a decision maker (DM) provides preference information at each iteration simple as a reference point consisting of desirable objective function values and the number of solutions to be compared. Using this information, the desired number of solutions are generated to represent the region of interest of the Pareto optimal front associated to the reference point given. Interactive WASF-GA implies a much lower computational cost than the original WASF-GA because it generates a small number of solutions. This speeds up the convergence of the algorithm, making it suitable for many decision-making problems. Its efficiency and usefulness is demonstrated with a five-objective optimization problem.

**Keywords:** Multiobjective programming; Pareto optimal solutions; Reference point approach; Interactive methods; Evolutionary algorithms.

## 1 Introduction

Many real-world applications arising in e.g. engineering involve solving multiobjective optimization problems where several conflicting objectives must be optimized over a set of feasible solutions. In many occasions, these problems can be complex to solve because they deal with different types of functions (nonlinear, nondifferentiable, discontinuous, etc.) and different types of variables (continuous, integer, binary, etc.). They may even involve black-box functions, whose computational cost can be high.

Commonly, there is no solution where all the objectives can reach their individual optima and we look for so-called *Pareto optimal solutions*. These solutions are defined as solutions where an improvement of any objective always implies

a sacrifice in at least one of the others. The set of Pareto optimal solutions is called the *Pareto optimal set* and its image in the objective space is known as the *Pareto optimal front*. A *decision maker (DM)*, a person who is interested in solving the problem, decides which Pareto optimal solution best satisfies his/her preferences and this solution is commonly known as *the most preferred solution*.

There exists a great amount of methods to deal with multiobjective optimization problems in the literature. On the one hand, interactive Multiple Criteria Decision Making (MCDM) methods are widely used due to the gradual incorporation of the DM's preferences into the solution process in order to generate one or a small set of Pareto optimal solutions according to these preferences [13, 16]. On the other hand, during the last decades, Evolutionary Multiobjective Optimization (EMO) algorithms have become very popular for solving different types of problems [1, 2]. Their main aim is the approximation of the whole Pareto optimal front. However, although knowing the ranges of the objectives functions and the conflict degree among them can be of great help for having a good knowledge of the problem itself, the task of identifying a single preferred Pareto optimal solution that pleases the DM may not be easy. Also, approximating the whole Pareto optimal front may be impossible in e.g. large scale or computationally complex problems. These difficulties can be managed by considering an interactive method that uses tools from an EMO algorithm. To be more precise, one can incorporate preference information into EMO algorithms to overcome various (computational and cognitive) challenges [8].

Some interactive EMO methods have been proposed in the literature, including the following ones. The Reference-Point-Based NSGA-II (R-NSGA-II) proposed in [5] modifies NSGA-II [4] as follows. According to one or several reference points given by a DM, the crowding distance used in NSGA-II is replaced by a preference distance, which equally emphasizes objective vectors that are close to any of the reference points with respect to the Euclidean distance. In [21] an interactive EMO method called the Preference Based Evolutionary Algorithm (PBEA) was proposed, which modifies the EMO algorithm IBEA [27]. PBEA allows the DM to interactively give reference points, with which the binary quality indicator of IBEA (which measures the minimal distance by which an individual needs to be improved in each objective to become nondominated) is redefined using an achievement scalarizing function [24] from MCDM. A Preference-based Interactive Evolutionary (PIE) algorithm was proposed in [18]. Starting from a solution selected from a randomly generated population or from a reference point, PIE progressively improves the objective function values by minimizing an achievement scalarizing function [15] at each iteration using a single-objective evolutionary algorithm. The DM guides interactively the algorithm by deciding from which solution, at which distance from the Pareto optimal front and in which direction the search for the next solutions is continued. iMOEA/D [9] is an interactive version of the well-known MOEA/D method [25], where a set of solutions is shown to the DM at intermediate generations, who must choose one of them. Then, the search is guided to the neighbourhood of the selected solution by relocating the weight vectors which determine the search directions.

In [23], an interactive EMO method called iPICEA-G, which is based on the PICEA-G algorithm (Preference-Inspired Co-Evolutionary Algorithm) [22], was proposed. In this method, the DM's preferences can be given either as a search direction or as a reference point. In the former case, the DM has to indicate the importance (s)he gives to each objective function and an angle between 0 and  $\Pi/2$  which determine the search range. This kind of information may be difficult to provide for the DM. In the case of a reference point, all objectives are given the same importance and the search range is set according to the number of objective functions. In [19], an interactive evolutionary algorithm was suggested which tries to find the most preferred solution with a limited number iterations expecting DM's involvement. The preference information is given by choosing a desirable solution among a set of solutions.

Whatever algorithm is used, the final purpose of solving any multiobjective optimization problem is that the DM can find her/his most preferred solution. Thus, once a set of solutions that approximates the Pareto optimal front is found, we cannot overlook the decision making phase in which the DM must make an adequate decision to choose the final solution. Obviously, the DM plays an active role in the process and an interactive method is supposed to be appealing and acceptable to her/him because (s)he is involved in the process. However, it is important to consider several issues. On the one hand, for decision making purposes, only a few solutions must be analysed by the DM in order not to overwhelm her/him. Comparing too many solutions may be difficult in the presence of a high number of objectives. On the other hand, asking preference information in a format as simple as possible is very important since it makes the interactive process more meaningful. Besides, if the DM feels that the solutions obtained reflect well enough her/his wishes, and they are improved progressively, (s)he is more motivated and it is more likely that (s)he wants to keep on iterating until the most preferred solution is found.

Based on this, in this paper, we concentrate on the decision making phase and the interaction with the DM necessary for solving any multiobjective optimization problem. Taking into account the previous ideas, we propose an interactive EMO method that generates a small set of solutions at each iteration and which needs from the DM preference information which is not cognitively demanding. The interactive method proposed is called *Interactive WASF-GA* and it is an interactive version of a preference-based EMO algorithm called WASF-GA [17]. At each iteration of Interactive WASF-GA, the DM indicates the number of solutions to be compared and a reference point containing aspiration levels, that is, objective function values that are desirable. According to this, WASF-GA is executed iteratively to generate the desired number of solutions in the region of interest defined by the given reference point. However, we do not only propose an interactive algorithm, but we also suggest a user interface aimed at enhancing the interaction with DM when solving a problem with Interactive WASF-GA.

The rest of this paper is organized as follows. In Section 2, we introduce the main concepts and notations used, including a brief overview of the WASF-GA algorithm. Interactive WASF-GA is motivated and described in Section 3, where

we also carry out a comparative analysis with respect to other interactive EMO algorithms. In Section 4, a computational implementation is described, showing the graphical user interface proposed and the solution process of a five-objective optimization problem. Finally, conclusions are drawn in Section 5.

## 2 Formulation and Background Concepts

### 2.1 Concepts and Notation

A general *multiobjective optimization problem* is defined by

$$\begin{aligned} & \text{minimize} && \{f_1(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{1}$$

where  $f_i : S \rightarrow \mathbf{R}$ , for  $i = 1, \dots, k$  ( $k \geq 2$ ) are the *objective functions* that we wish to optimize (to minimize in our case) simultaneously. The decision variables  $\mathbf{x} = (x_1, \dots, x_n)^T$  are referred to as *solutions* or *decision vectors* and they belong to  $S \subset \mathbf{R}^n$ , called the feasible set. The images of the solutions  $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$  are called *objective vectors*. The image of the feasible set in the *objective space*  $\mathbf{R}^k$  is called the *feasible objective region*  $Z = \mathbf{f}(S)$ .

Since, in the presence of conflicting objective functions, it is not possible to find a solution where all the objectives can reach their individual optima, there exist solutions that are mathematically incomparable. In these solutions, no objective function can be improved without deteriorating at least one of the others. A solution  $\mathbf{x} \in S$  is said to be *Pareto optimal* if there does not exist another  $\mathbf{x}' \in S$  such that  $f_i(\mathbf{x}') \leq f_i(\mathbf{x})$  for all  $i = 1, \dots, k$  and  $f_j(\mathbf{x}') < f_j(\mathbf{x})$  for at least one index  $j$ . The corresponding objective vector  $\mathbf{f}(\mathbf{x})$  is called a *Pareto optimal objective vector*. The set of all Pareto optimal solutions is called a *Pareto optimal set*, denoted by  $E$ , and the set of all Pareto optimal objective vectors is called a *Pareto optimal front*, denoted by  $\mathbf{f}(E)$ .

Given two objective vectors  $\mathbf{z}, \mathbf{z}' \in Z$ , we say that  $\mathbf{z}$  *dominates*  $\mathbf{z}'$  if and only if  $z_i \leq z'_i$  for all  $i = 1, \dots, k$ , with at least one strict inequality. In the context of EMO algorithms, we refer to a *nondominated set* as a set of solutions whose objective vectors are not dominated by the objective vector corresponding to any other solution in the set.

The *ideal objective vector* and the *nadir objective vector* are defined, respectively, as  $\mathbf{z}^* = (z_1^*, \dots, z_k^*)^T$  such that  $z_i^* = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$  ( $i = 1, \dots, k$ ), and as  $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_k^{\text{nad}})^T$  such that  $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$  ( $i = 1, \dots, k$ ). That is, the ideal and the nadir values are, respectively, the best and the worst values that each objective function can achieve in the Pareto optimal front (that is, they define lower and upper bounds for the objective functions). While the ideal objective vector can be easily obtained, the nadir objective vector is, in general, more difficult to calculate and typically we need to settle for approximations [3, 20]. In what follows, we assume that the Pareto optimal front is bounded and that there are available estimations of the ranges of the objective function values.

From the mathematical point of view, all Pareto optimal solutions can be regarded as equally desirable and we need information about the preferences of a DM to identify one as the final solution to be implemented [13]. A natural way to express preferences consists of specifying desirable objective function values, which constitute the components of the so-called reference point. A *reference point* is given by  $\mathbf{q} = (q_1, \dots, q_k)^T$ , where  $q_i$  is an aspiration level for the objective function  $f_i$  provided by the DM, for all  $i = 1, \dots, k$ . Usually,  $\mathbf{q}$  is said to be *achievable* for (1) if  $\mathbf{q} \in Z + \mathbb{R}_+^k$  (where  $\mathbb{R}_+^k = \{\mathbf{y} \in \mathbb{R}^k \mid y_i \geq 0 \text{ for } i = 1, \dots, k\}$ ), that is, if either  $\mathbf{q} \in Z$  or  $\mathbf{q}$  is dominated by some Pareto optimal objective vector. Otherwise, the reference point is said to be *unachievable*, that is, for an unachievable reference point, all components cannot be achieved simultaneously (in some situations, a reference point is unachievable because some components cannot be achieved although other ones can be attained).

Once a reference point is given, a so-called *achievement (scalarizing) function* (ASF) [24] can be minimized over the feasible set in order to find the Pareto optimal solution which best fits the reference point. These functions combine the original objective functions with the preferences of the DM into a scalar valued function. For an overview about ASFs, see [15].

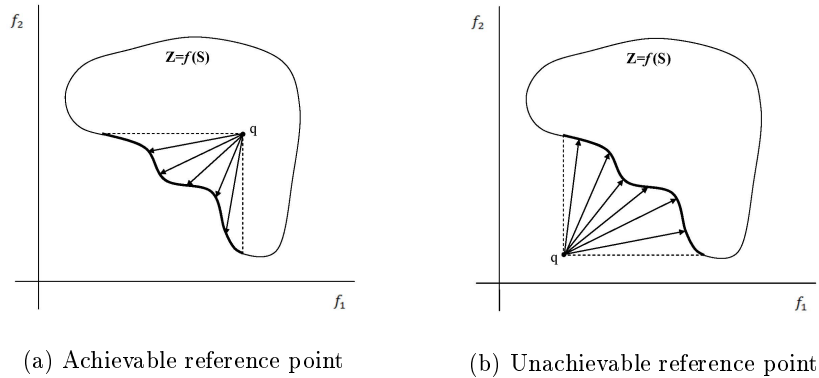
## 2.2 WASF-GA Algorithm

As previously mentioned, the interactive method we proposed is based on the preference-based EMO algorithm called WASF-GA [17]. This algorithm tries to approximate the region of interest of the Pareto optimal front defined by a reference point  $\mathbf{q}$  given by a DM. In [17], the *region of interest of the Pareto optimal front associated to  $\mathbf{q}$*  is defined as followed. When  $\mathbf{q}$  is achievable, the region of interest is the subset of Pareto optimal objective vectors  $\mathbf{f}(\mathbf{x})$ , with  $\mathbf{x} \in E$ , which verify that  $f_i(\mathbf{x}) \leq q_i$ , for every  $i = 1, \dots, k$ . On the other hand, if  $\mathbf{q}$  is unachievable, the region of interest is formed by the Pareto optimal objective vectors  $\mathbf{f}(\mathbf{x})$ , with  $\mathbf{x} \in E$ , which verify that  $f_i(\mathbf{x}) \geq q_i$ , for every  $i = 1, \dots, k$ . Therefore, in the achievable case, this region of interest contains all the Pareto optimal solutions which dominate the reference point and, thus, which are the most interesting solutions for the DM. In the unachievable case, the region of interest is formed by the Pareto optimal solutions which are dominated by the reference point. In this case, solutions lying in this region are likely to be more appealing for the DM than the ones outside it because, at them, the objective function values differ from the aspiration values as little as possible, although they do not improve any of them. The solutions outside this region may improve some of the aspiration levels (and not all of them) but at the expense of a sacrifice in the rest of reference levels, what may not be so attractive for the DM.

To approximate the region of interest, WASF-GA maintains a diverse set of nondominated solutions by considering, on the one hand, a predefined set of weight vectors in the weight vector space  $(0, 1)^k$  (let us consider  $N_\mu$  vectors of weights) and, on the other hand, by minimizing at each generation the ASF proposed by Wierzbicki in [24] for the reference point given. Roughly speaking, at each generation of WASF-GA, parents and offspring are classified into several

fronts. This classification is done according to the values that each individual takes on the ASF, for the reference point and for each of the weight vectors in the set. To be more precise, the first front is formed by the solutions which reach the lowest value of the ASF for each of the  $N_\mu$  weight vectors; the second front is constituted by the individuals with the next lowest value of the ASF for each of the  $N_\mu$  weight vectors, and so on until every individual has been included into some front. Afterwards, the solutions which are passed to the next generation are those in the lower level fronts until completing the new population. The solutions selected can be considered as the best individuals at the current generation for minimizing the ASF with respect to the weight vectors considered. The outcome of WASF-GA is the first front of the last generation, which has  $N_\mu$  individuals. From the practical point of view, the region of interest is approximated by projecting the reference point onto the Pareto optimal front in different ways, by using the set of projection directions (or search directions) defined by the inverses of the  $N_\mu$  weight vectors considered.

Figure 1 gives a graphical idea of the working procedure of WASF-GA in a biobjective optimization problem. The region of interest in the Pareto optimal front has been highlighted with a bold line in both cases, and the arrows represent the projection directions determined by a set of weight vectors. It can be seen that, by varying the weight vectors and by emphasizing at each generation those individuals which minimize the ASF for each weight vector, the region of interest can be approximated by projecting the reference point onto the Pareto optimal front using several projection directions, for both unachievable and achievable reference points. For more details about WASF-GA, see [17].



**Fig. 1.** Idea of the working procedure of WASF-GA

### 3 Interactive WASF-GA

Based on the success of interactive MCDM methods, we propose a new interactive method using a preference-based EMO algorithm. In Interactive WASF-GA, the preference information indicated by the DM at each iteration  $it$  are aspiration levels for the objective functions, which determine a reference point denoted by  $\mathbf{q}^{it}$ , and the number of solutions (s)he wants to compare, denoted by  $N_S^{it}$ . The set of new solutions is generated by applying the WASF-GA algorithm using as many weight vectors as the number of solutions indicated by the DM, that is,  $N_\mu = N_S^{it}$  at each iteration. Let us denote by  $\mu^{it,j}$  the weights vectors used at iteration  $it$ , for  $j = 1, \dots, N_S^{it}$ .

#### 3.1 Algorithm of Interactive WASF-GA

The steps of the Interactive WASF-GA are the following ones:

- Step 1. Initialization.** Initialize  $it = 1$ .
- Step 2. Preference information I.** If  $it > 1$  and the DM wants to generate new solutions using the previous reference point, set  $\mathbf{q}^{it} = \mathbf{q}^{it-1}$ . Otherwise, ask the DM to specify a reference point  $\mathbf{q}$  and set  $\mathbf{q}^{it} = \mathbf{q}$ .
- Step 3. Preference information II.** Ask the DM how many solutions (s)he would like to see,  $N_S^{it}$ . If  $it > 1$  and  $N_S^{it} = N_S^{it-1}$ , set  $\mu^{it,j} = \mu^{it-1,j}$  for every  $j = 1, \dots, N_S^{it}$  and go to Step 5. Otherwise, continue.
- Step 4. Generation of the weight vectors.** Following the procedure described in [17], generate  $N_S^{it}$  weight vectors, denoted by  $\mu^{it,j}$  for  $j = 1, \dots, N_S^{it}$ .
- Step 5. Generation of solutions.** Generate  $N_S^{it}$  solutions with the WASF-GA algorithm using the set of weight vectors  $\mu^{it,j}$  for  $j = 1, \dots, N_S^{it}$ , and show the solutions to the DM.
- Step 6. Termination rule.** Ask the DM to select the most preferred of the  $N_S^{it}$  solutions and denote it by  $\mathbf{x}^{it}$ . If the DM wishes to *Stop*, the solution process concludes with  $\mathbf{x}^{it}$  as the final solution and  $\mathbf{f}(\mathbf{x}^{it})$  as a final objective vector. Otherwise, set  $it = it + 1$  and go to Step 2.

Next, let us make some remarks about some aspects of the algorithm:

- When  $it = 1$ , the DM must give a reference point in Step 2 because no reference point was provided previously.
- The number of solutions to be shown to the DM can be changed at each iteration in Step 3, but (s)he can alternatively maintain the same number along several iterations. In that case, the same set of weight vectors can be used through these iterations and only the reference point changes.



- When the DM decides to generate new solutions using the same reference point, only the  $N_S^{it}$  weight vectors must be recalculated. Given that the procedure described in [17] generates an initial large number of vectors from which the weight vectors needed are selected, the new  $N_S^{it}$  weight vectors can be obtained using again the same initial vectors to reduce the computational effort. Furthermore, the weight vectors that were already used at the previous iteration must be internally removed in order to assure that different solutions are provided to the DM.
- The final population generated at one iteration can be used as the initial population at the next iteration, which allows to accelerate the speed of the solution process. This increases the convergence speed of the algorithm since the initial population is already close to the Pareto optimal front.
- In order to guarantee at least local Pareto optimality of the final solution, the last solution chosen by the DM can be locally improved by minimizing the ASF proposed in [24] using the objective function values achieved by this solution as the reference point, with some local optimization method.

### 3.2 Comparative analysis

In order to compare Interactive WASF-GA with some of the reference point-based interactive EMO algorithms mentioned in Section 1, we present Table 1, which summarizes the main features of each algorithm. Next, we detail the information given on each column and, as an example, we explain this information for the Reference-Point-Based NSGA-II algorithm [5]. In this algorithm, at each iteration, the DM must specify one or several reference points, which is indicated in the *'Preference information'* column. At each iteration with preferences, the outcome population shown to the DM consists of individuals in the first non-dominated front of the last generation. This is indicated in the *'Solutions shown to the DM'* column. The *'Computational cost'* column contains the complexity of the basic operations of each algorithm in one iteration, considering their worst cases. In this column,  $k$  represents the number of objective functions and  $N$  is the population size used. For the Reference-Point-Based NSGA-II algorithm, we have used only one reference point and we have taken into account the computational cost needed for carrying out the nondominated sorting procedure, the preference distance assignment, the preference distance sorting and the  $\epsilon$ -based selection strategy (see [5]). Finally, if the algorithm needs to set any additional parameters to be executed, they are indicated in the last column. In the example considered, a value for  $\epsilon$  is necessary to compute the niching operator.

Let us now analyse Table 1. Firstly, from the cognitive point of view, the preference information required from the DM in Interactive WASF-GA is very simple compared to some of the other methods. For example, in the PIE algorithm or in iPICEA-G, the DM is asked for the percentage of distance to the (unknown) Pareto optimal front or for a search angle, respectively. This type of information may not be easy to understand by the DM. Secondly, as it can be seen in the third column, the only method that generates exactly the number of solutions the DM wants to see is Interactive WASF-GA. Except from

**Table 1.** Comparison of several methods

At each iteration	Preference information	Solutions shown to the DM	Computational cost	Additional parameters
Interactive WASF-GA	A reference point and the number of solutions to be compared ( $N_S$ )	The $N_S$ solutions generated at the last generation	$O(k \cdot N \cdot N_S)$	No
Reference-Point-Based NSGA-II [5]	One or several reference points	First nondominated front of the last generation	$O(k \cdot N^2)$	To control the extent of solutions, an $\epsilon$ -clearing idea is used in the niching operator
PBEA [21]	A reference point	Population of the last generation	$O(k \cdot N^2)$	The extent of solutions is controlled by an operator $\delta$
PIE [18]	Preferential weights, a reference point and the distance to the Pareto optimal front	A solution at the distance indicated to the Pareto optimal front	$O(k \cdot N)$	If the DM wants to investigate solutions previously obtained, (s)he must indicate the number of solutions to be shown
iMOEA/D [9]	The number of solutions to be shown and choosing one solution among a set of solutions	Solutions at intermediate generations	$O(k \cdot N^2)$	The number of iterations to be taken and a reduction factor of the preferred region
iPECEA-G [23]	A reference point or a search direction with a search angle	Population of the last generation	$O(k \cdot N^2)$	Search angle to control the extent of solutions

PIE (which generates one solution at each iteration) and i-MOEA/D, the rest of methods show the solutions generated at the last generation. Consequently, the number of solutions shown to the DM may be too high for making a fair comparison and cannot be known beforehand. Besides, Interactive WASF-GA shows nondominated solutions which approximate the region of interest, instead of showing nondominated solutions generated at intermediate generations, as in i-MOEA/D. This may be seen as a strength of our algorithm, since the solutions found at intermediate generations may be still far from the region of interest and may not give a good idea about the real trade-offs among the objectives in this region. Thirdly, regarding the 'Computational cost' column, the algorithm proposed has a much lower computational cost than those needed by other algorithms given that the number of solutions the DM wants to compare in Interactive WASF-GA, denoted by  $N_S$ , is expected to be much lower than the population size. The only method with a lower computational cost than Interactive WASF-GA is PIE because this method internally solves a single-objective (scalarized) optimization problem with a single-objective algorithm instead of solving the multiobjective optimization problem itself. Finally, it is worthy to mention that Interactive WASF-GA does not need to set any additional parameter during the solution process, while the other algorithms do require some (see last column). In most of them, these additional parameters control the ex-

tent of solutions in the region approximated in the Pareto optimal front and, consequently, they affect the outcome of the algorithm.

The previous analysis highlights that, in comparison with some of the state-of-art interactive EMO algorithms, Interactive WASF-GA requires very simple preference information from the DM and it is able to generate exactly the number of solutions the DM wants to see in the region of interest. Besides, its computational cost is quite limited and it does not need to set any additional parameter during the solution process. As shortcomings of Interactive WASF-GA, we can say that, on the one hand, the distribution of the  $N_S$  weight vectors influences the distribution of the solutions generated and, thus, special emphasis must be taken for using weights which produce well-distributed projection directions [17]. This may be overcome by producing a large number of weight vectors (e.g., 100 or more) and then using the  $k$ -means clustering [12] to select the  $N_S$  most representative ones.

On the other hand, one may think that it may be not assured that the DM is shown exactly  $N_S$  solutions because minimizing the ASF using different weight vectors does not assure to generate different Pareto optimal solutions (for example, in problems with discontinuous Pareto optimal fronts). In order to avoid such a situation, more than  $N_S$  weight vectors may be used in WASF-GA, e.g. we can use  $N_S^* = 2 \cdot N_S$  vectors. In this way, more solutions are generated in the region of interest and, afterwards, the set of solutions obtained may be filtered using e.g. the  $k$ -means clustering in order to get the  $N_S$  most representative solutions. This procedure, which is also used in [11], increases the computational cost and it must only be applied in case this situation is internally detected.

## 4 Computational Implementation

In this section, we demonstrate the computational implementation created for Interactive WASF-GA, which is in a preliminary development phase. It has been developed in Java by using jMetal [7], a Java-based framework for multiobjective optimization. In order to check the performance of the method proposed, we have introduced into the platform several test problems from the ZDT, DTLZ and WFG families [6, 10, 26], respectively by now, for which the number of objectives can vary between 2 and 6. Of course, this implementation must be further improved so that other multiobjective optimization problems considered.

The main menu can be seen in Figure 2, where we consider the DTLZ2 test problem with 5 objective functions. The information is organized as follows:

- **Algorithm’s configuration.** There are three parameters in this box: (a) the number of solutions the DM would like to compare at the current iteration; (b) the population size and (c) the number of generations, the latter two being technical parameters. In the implementation, default values are recommended for these technical parameter for each problem although they can be modified if so desired. For example, if the DM thinks that the solutions are not good enough and (s)he wishes to obtain solutions closer to the Pareto optimal front, one can allow more generations to be carried out.

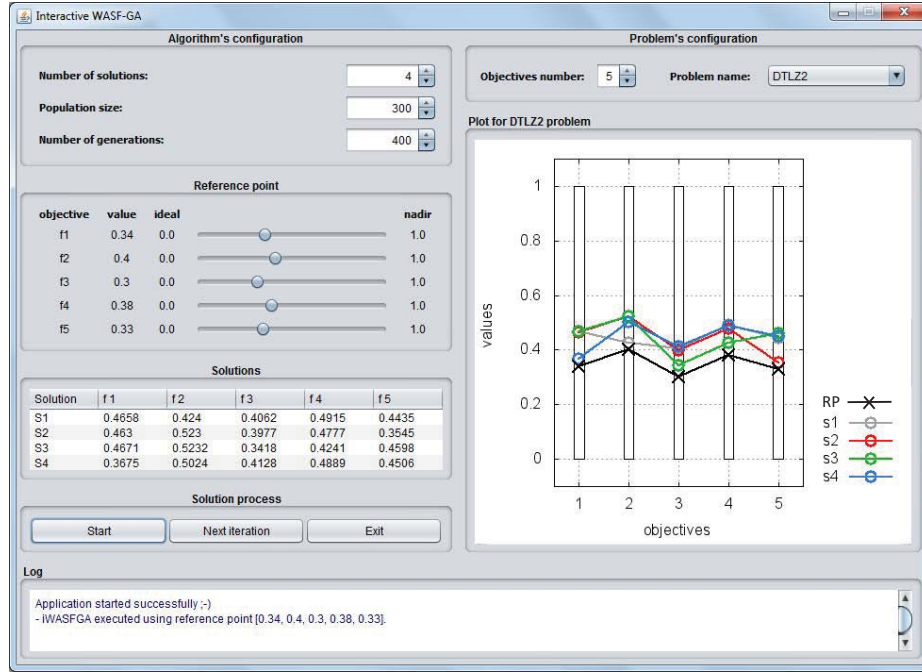


Fig. 2. Interface of Interactive WASF-GA - DTLZ2 problem, iteration 1

- **Problem's configuration.** In this box, the multiobjective optimization problem to be solved is selected.
- **Reference point.** Approximations of the ideal and the nadir values are provided to the DM in order to let her/him know the ranges of the objective functions. By clicking on each slider and moving it, the DM can set the aspiration level for each objective, and the corresponding numerical values are shown in the *Value* column.
- **Solution process.** To generate the  $N_S^{it}$  solutions the DM wishes to compare, (s)he must click the *Start* button. If (s)he decides to take a new iteration by changing some preference information (the reference point and/or the number of solutions to be generated), (s)he must click the *Next Iteration* button to generate new solutions.
- **Solutions.** Here, the objective values of the  $N_S^{it}$  solutions obtained for the current reference point are shown.
- **Plot for the problem.** The objective vectors of the solutions found and the reference point are shown graphically in order to ease the comparison among them. For bi-objective optimization problems, they are plotted in  $\mathbf{R}^2$ , and also the Pareto optimal front is shown if it is known. For multiobjective optimization problems with three or more objective functions, we use a *value path* [14] representation to show the solutions obtained, as can be seen in Figure 2. The reference point (labelled as *RP*) and each solution are plot by

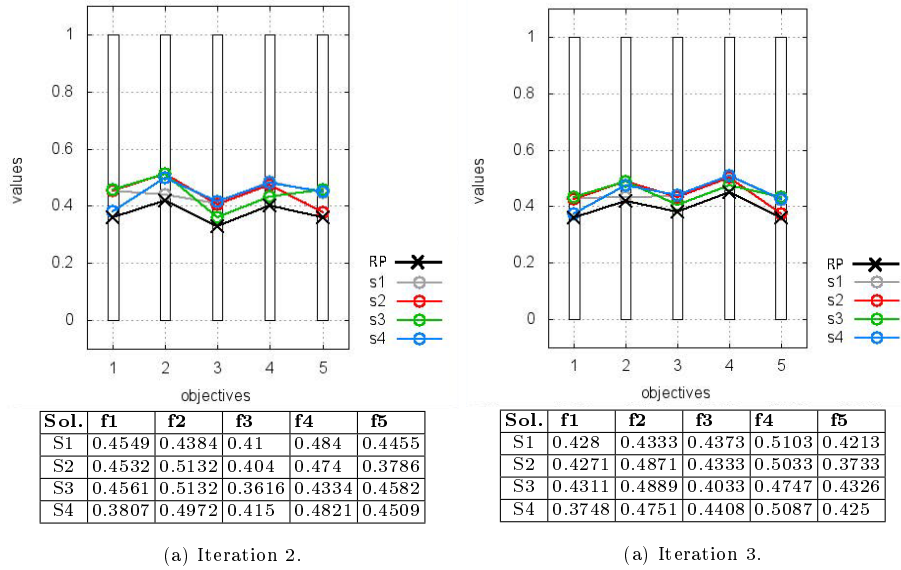
lines that go across different columns which represent the objective function values they reach. The lower and upper ends of each column represent the total values range of each objective function, that is, its ideal and nadir values, respectively.

- **Log.** The *Log* box indicates if there has been any error during the execution.

In what follows, we illustrate the performance of Interactive WASF-GA with the DTLZ2 problem with 5 objectives. Let us assume that the preference information given by the DM at the first iteration is the one shown in Figure 2, that is, he set the first reference point as  $\mathbf{q}^1 = (0.34, 0.4, 0.3, 0.38, 0.33)$  and wanted to generate four solutions ( $N_S^1 = 4$ ). Analysing the value path and the objective values of the solutions generated, it can be seen that none of the solutions obtained has improved any aspiration value. Besides, a careful analysis of them highlights the conflict degree among the objective functions. It can be observed that, when a solution reaches an objective value closer to the corresponding aspiration value, the values achieved by the rest of objective functions are further from their aspiration values. This can be easily seen, for example, in solutions S2 and S4, which attain values close to the aspiration levels for objective 5 and objective 1, respectively, at the expense of the rest of the objective functions.

According to the above analysis, the DM decided to relax all the aspiration levels and he set the new reference point as  $\mathbf{q}^2 = (0.36, 0.42, 0.33, 0.4, 0.36)$  for generating four new solutions ( $N_S^2 = 4$ ). The solutions generated are shown in image and table (a) of Figure 3. As at the previous iteration, no solution reaches or improves any aspiration level, but it can be seen that now the ranges of objective values achieved by all the solutions are closer to their aspiration levels. Based on this, the DM wished to have another iteration in order to check the solutions that could be obtained if he maintained the same aspiration levels for the objectives 1, 2 and 5 and he relaxed a bit more the ones for the objectives 3 and 4. He fixed the reference point as  $\mathbf{q}^3 = (0.36, 0.42, 0.38, 0.45, 0.36)$ . The four solutions ( $N_S^3 = 4$ ) found can be seen in image and table (b) of Figure 3 and it can be observed that now they are even closer to the reference point. Although the reference point was still unattainable, the DM was satisfied enough with solution S2. This solution improved the values achieved for the objective functions 1, 2 and 5 when compared to the ones reached by most of the solutions at the previous iteration and, at the same time, it attained the second best values for the objectives 3 and 4. After three iterations the DM found the most preferred solution and was convinced of its goodness.

With this example, we have shown the behaviour of Interactive WASF-GA and the user interface proposed. If the DM changes the reference point, we have seen that the solutions generated are different from the ones previously produced. And if the DM indicates a higher or a smaller number of solutions, more or less solutions are produced accordingly. We have not computationally compared our algorithm with other interactive methods because a quantitative assessment of interactive approaches is very difficult in practice when interacting with a DM. Furthermore, traditional comparative tables which evaluate the performance of EMO algorithms after several independent runs are not meaningful for assessing



**Fig. 3.** Solution process of the DTLZ2 problem.

Interactive WASF-GA because we focus on the DM's interaction and the decision making phase, and not only on the approximation of the Pareto optimal front.

## 5 Conclusions

In this paper, a new interactive evolutionary algorithm has been proposed for solving multiobjective optimization problems. The new algorithm is called Interactive WASF-GA and it is based on the preference-based EMO algorithm WASF-GA. At each iteration of Interactive WASF-GA, very easy to understand preference information is asked to the DM: just a reference point (containing desirable objective function values) and the number of solutions the DM wishes to compare. According to this information, a set with this number of solutions is generated in order to represent the region of interest of the Pareto optimal front defined by the reference point given. Subsequently, the DM analyses the solutions found and decides either to stop or to carry out a new iteration by redefining the preference information.

While the original WASF-GA approximates the region of interest with a high number of nondominated solutions, the interactive version only needs to generate few representative nondominated solutions. This fact allows to accelerate the solution process and reduces the computational cost needed. Besides, as the DM just compares a small number of solutions in the region of interest in order to find her/his most preferred solution, the solution process is not very demanding and requires a low cognitive effort. Furthermore, Interactive WASF-GA is able to generate as many solutions as the DM indicates, and this is a strength in

comparison to other interactive EMO algorithms, which provide the DM with sets of solutions that may be too large to be compared. Also, it is noteworthy that Interactive WASF-GA does not need to set any additional parameter.

We have demonstrated the applicability of Interactive WASF-GA with a five-objective optimization problem which has shown how our algorithm can be used for reaching a solution interesting for the DM. Next, we plan to apply the algorithm proposed to real-life multiobjective optimization problems.

## Acknowledgements

This research was partly supported by the Spanish Ministry of Innovation and Science (MTM2010-14992) and by the Andalusia Regional Ministry of Innovation, Science and Enterprises (PAI groups SEJ-445 and SEJ-532).

## References

1. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Second Edition. Springer, New York (2007)
2. Deb, K.: *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Chichester (2001)
3. Deb, K., Miettinen, K., Chaudhuri, S.: Towards an estimation of nadir objective vector using a hybrid of evolutionary and local search approaches. *IEEE Transactions on Evolutionary Computation* 14(6), 821–841 (2010)
4. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
5. Deb, K., Sundar, J., Ubay, B., Chaudhuri, S.: Reference point based multi-objective optimization using evolutionary algorithm. *International Journal of Computational Intelligence Research* 2(6), 273–286 (2006)
6. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: *Congress on Evolutionary Computation, CEC-2002*. pp. 825–830 (2002)
7. Durillo, J.J., Nebro, A.J.: jMetal: A java framework for multi-objective optimization. *Advances in Engineering Software* 42, 760–771 (2011)
8. Figueira, J.R., Greco, S., Mousseau, V., Slowinski, R.: Interactive multiobjective optimization using a set of additive value functions. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (eds.) *Multiobjective Optimization, Interactive and Evolutionary Approaches*. pp. 97–120. Springer (2008)
9. Gong, M., Liu, F., Zhang, W., Jiao, L., Zhang, Q.: Interactive MOEA/D for multi-objective decision making. In: *13th Annual Conference on Genetic and Evolutionary Computation, GECCO 2011*. pp. 721–728 (2011)
10. Huband, S., Hingston, P., Barone, L., While, L.: A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation* 10(5), 477–506 (2006)
11. Luque, M., Ruiz, F., Steuer, R.E.: Modified interactive Chebyshev algorithm (MICA) for convex multiobjective programming. *European Journal of Operational Research* 204(3), 557–564 (2010)

12. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5-th Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 281–297. Berkeley, University of California Press (1967)
13. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston (1999)
14. Miettinen, K.: Survey of methods to visualize alternatives in multiple criteria decision making problems. *OR Spectrum* 36(1), 3–37 (2014)
15. Miettinen, K., Mäkelä, M.M.: On scalarizing functions in multiobjective optimization. *OR Spectrum* 24(2), 193–213 (2002)
16. Miettinen, K., Ruiz, F., Wierzbicki, A.P.: Introduction to multiobjective optimization: Interactive approaches. In: Branke, J., Deb, K., Miettinen, K., Slowinski, R. (eds.) *Multiobjective Optimization, Interactive and Evolutionary Approaches*. pp. 27–58. Springer (2008)
17. Ruiz, A.B., Saborido, R., Luque, M.: A preference-based evolutionary algorithm for multiobjective optimization: The weighting achievement scalarizing function genetic algorithm. *Journal of Global Optimization in press*, DOI 10.1007/s10898-014-0214-y (2014)
18. Sindhya, K., Ruiz, A.B., Miettinen, K.: A preference based interactive evolutionary algorithm for multi-objective optimization: PIE. In: Takahashi, R., Deb, K., Wanner, E., Greco, S. (eds.) *Evolutionary Multi-Criterion Optimization, Proceedings*. pp. 212–225 (2011)
19. Sinha, A., Korhonen, P., Wallenius, J., Deb, K.: An interactive evolutionary multi-objective optimization algorithm with a limited number of decision maker calls. *European Journal of Operational Research* 233(3), 674–688 (2014)
20. Szczepanski, M., Wierzbicki, A.P.: Application of multiple criterion evolutionary algorithm to vector optimization, decision support and reference point approaches. *Journal of Telecommunications and Information Technology* 3(3), 16–33 (2003)
21. Thiele, L., Miettinen, K., Korhonen, P., Molina, J.: A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation* 17(3), 411–436 (2009)
22. Wang, R., Purshouse, R.C., Fleming, P.J.: Preference-inspired coevolutionary algorithms for many-objective optimization. *IEEE Transactions on Evolutionary Computation* 17(4), 474–494 (2013)
23. Wang, R., Purshouse, R.C., Fleming, P.J.: "Whatever works best for you" - a new method for a priori and progressive multi-objective optimisation. In: Purshouse, R.C., Fleming, P.J., Fonseca, C.M., Greco, S., Shaw, J. (eds.) *Evolutionary Multi-Criterion Optimization, Proceedings*. pp. 337–351. Springer (2013)
24. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) *Multiple Criteria Decision Making, Theory and Applications*. pp. 468–486. Springer (1980)
25. Zhang, Q., Li, H.: MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation* 11(6), 712–731 (2007)
26. Zitzler, E., Deb, K., Thiele, L.: Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8(2), 173–195 (2000)
27. Zitzler, E., Kuenzli, S.: Indicator-based selection in multiobjective search. In: Yao, X., Burke, E., Lozano, J.A., Smith, J., Merelo-Guervos, J.J., Bullinaria, J.A., Rowe, J., Tino, P., Kaban, A., Schwefel, H.P. (eds.) *8th International Conference on Parallel Problem Solving from Nature - PPSN VIII*. pp. 832–842. Springer (2004)