

A Preference-based Evolutionary Algorithm for Multiobjective Optimization: The Weighting Achievement Scalarizing Function Genetic Algorithm

Submitted version of the published paper with DOI
<https://doi.org/10.1007/s10898-014-0214-y>

Ana Belén Ruiz, Rubén Saborido and Mariano Luque
Universidad de Málaga
Calle Ejido 6, 29071 Málaga, Spain
abruiz@uma.es, rsain@uma.es, mluque@uma.es

September 26, 2024

Abstract

When solving multiobjective optimization problems, preference-based Evolutionary Multiobjective Optimization (EMO) algorithms introduce preference information into an evolutionary algorithm in order to focus the search for objective vectors towards the region of interest of the Pareto optimal front. In this paper, we suggest a preference-based EMO algorithm called *Weighting Achievement Scalarizing Function Genetic Algorithm (WASF-GA)*, which considers the preferences of the decision maker (DM) expressed by means of a reference point. The main purpose of WASF-GA is to approximate the region of interest of the Pareto optimal front determined by the reference point, which contains the Pareto optimal objective vectors that obey the preferences expressed by the DM in the best possible way. The proposed approach is based on the use of an achievement scalarizing function (ASF) and on the classification of the individuals into several fronts. At each generation of WASF-GA, this classification is done according to the values that each solution takes on the ASF for the reference point and using different weight vectors. These vectors of weights are selected so that the vectors formed by their inverse components constitute a well-distributed representation of the weight vectors space. The efficiency and usefulness of WASF-GA is shown in several test problems in comparison to other preference-based EMO algorithms. Regarding a metric based on the hypervolume, we can say that WASF-GA has outperformed the other algorithms considered in most of the problems.

Keywords: multiobjective optimization, Pareto optimal solutions, reference point approach, achievement scalarizing function, evolutionary algorithm.

1 Introduction

Most real-life decision problems have to deal with several criteria, which must be optimized simultaneously. Such problems are called *multiobjective optimization problems* when both the criteria and the constraints that determine the feasible set of solutions can be mathematically expressed by functions. Typically, as it is not possible to find a solution where all the objectives can reach their individual optimum, we are interested in identifying a set of mathematically equally good solutions, the so-called *Pareto optimal solutions*. These solutions are defined as the solutions where none of the objective values can be improved without impairing at least one of the others. The set of Pareto optimal solutions is referred to as the *Pareto optimal set* and its image in the objective space is called the *Pareto optimal front*. When one or a set of Pareto optimal solutions are found, the *decision maker (DM)*, a person who is interested in solving the problem, decides which Pareto optimal solution satisfies better his/her preferences, which is commonly known as *the most preferred solution*.

As stated in [44], *Multiple Criteria Decision Making (MCDM)* [26, 35] and *Evolutionary Multi-objective Optimization (EMO)* [5, 8] are two research fields focussed on solving multiobjective optimization problems. Typically, the purpose in MCDM methods is to support a DM to find the most preferred solution. The multiobjective optimization problem is often scalarized into a single objective optimization problem taking into account the DM's preference information, which is solved using an appropriate mathematical programming technique to find a Pareto optimal solution. Different types of preference information asked from the DM include marginal rates of substitution, surrogate values for trade-offs, classification of objective functions and reference points. A reference point is defined by desirable objective function values. Depending on the moment when the DM's preferences are considered, MCDM methods can be classified into *a priori*, *a posteriori* and *interactive* methods. For further details, see [4, 26, 33, 35, 43] and references therein.

Compared to MCDM, EMO algorithms are population-based approaches whose main goal is to find a set of nondominated objective vectors which approximate the whole (unknown) Pareto optimal front. Once this set is generated, it is shown to the DM in order to allow him/her to find his/her most preferred solution. EMO algorithms have received a great deal of attention, mainly because they can generate a set of solutions in just one run. Among the most well-known EMO algorithms, we can mention the *Non-dominated Sorting Genetic Algorithm (NSGA-II)* [14], the *Strength Pareto Evolutionary Algorithm 2 (SPEA2)* [53] and the *Pareto Archive Evolutionary Strategy (PAES)* [29]. A recent group of EMO approaches is formed by the *EMO algorithms based on decomposition*. These techniques decompose the multiobjective optimization problem into a set of scalar (single-objective) optimization subproblems and optimize them simultaneously or subsequently following a population evolution based rule. One of them is the *Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D)* [50], which relies on the assumption that neighbour weighting vectors produce neighbour solutions.

Even though they are applied to similar problems, MCDM methods and EMO algorithms are based on different methodologies, and both of them have advantages and disadvantages. On the one hand, although MCDM methods concentrate only on the search for one or a few Pareto optimal solutions which result interesting for the DM, it may happen that these techniques cannot be applied to some types of problems due to limitations of mathematical programming. These limitations often occur in multiobjective problems with integer variables, nonconvexity conditions, nondifferentiable or discontinuous functions, etc.

On the other hand, EMO algorithms have shown to be very versatile in handling problems with different types of variables and objectives [56], but their main drawback is that they do not consider the preferences of the DM during the solution process. This means that the DM has to inspect a large set of solutions to find the most preferred solution, requiring both high computational and cognitive efforts⁽¹⁾. Besides, in EMO algorithms based on the Pareto dominance, the proportion of nondominated objective vectors in the population increases considerably in the presence of many objectives. This fact does not leave much room for new solutions to be included in the population, what may slow down the convergence of the algorithm and may decrease the diversity of the solutions [7, 15].

The advantages of MCDM and EMO philosophies may be hybridized in order to overcome some of their disadvantages. Considering preferences is important when dealing with a multiobjective optimization problem and it is not always necessary to explore the whole set of Pareto optimal objective vectors. In many situations, when the DM gives information about his/her preferences, (s)he is not usually interested either in a single Pareto optimal objective vector, or in the whole Pareto optimal front, but in a subset of Pareto optimal objective vectors which are relevant to him/her. We will refer to this subset as the *region of interest* of the Pareto optimal front.

Recently, a new group of algorithms, known as *preference-based EMO algorithms*, have been developed, whose main purpose is the approximation of just the region of interest of the Pareto optimal front. They are population-based algorithms which introduce information about the DM's preferences in order to focus the search for nondominated objective vectors in the most preferred region. As a result, the computational effort is reduced given that objective vectors far from the region of interest are progressively discarded. Besides, the DM only has to compare the trade-offs between nondominated objective vectors that are supposed to please him/her better, avoiding to analyse undesirable solutions and reducing the cognitive burden. Surveys of preference-based EMO algorithms can be seen in [6, 40], and more recent ones are given in [1, 27, 42].

Probably, the earliest EMO algorithm incorporating preferences was proposed by Fonseca and Fleming in 1993 [22]. In this paper, they suggested a multiobjective genetic algorithm (MOGA) and they proposed to introduce preferences by ranking the population according to the goals specified by the DM. The idea behind this approach is to give a higher priority to the objectives for which the goals are not fulfilled in order to improve them and push the search towards the region of interest of the Pareto optimal front.

We can also mention the method suggested by Branke et al [3], the *Guided Multi-Objective Evolutionary Algorithm* (G-MOEA), where the domination criterion is modified in order to incorporate the DM's preferences expressed in terms of acceptable trade-offs. Later, [2] compares G-MOEA with a modified version of NSGA-II, which incorporates a biased crowding distance based on a reference direction given by the DM.

The *Reference-Point-Based NSGA-II* (R-NSGA-II) proposed by Deb et al [17] modifies NSGA-II in the way the individuals of the last nondominated front are selected to be passed to the new population. According to one or several reference points, the crowding distance used in NSGA-II is replaced by a *preference distance*, which equally emphasizes objective vectors are close to any of the reference points with respect to the Euclidean distance.

In [10], Deb and Kumar defined the *Reference Direction Based NSGA-II* (RD-NSGA-II), which is based both on NSGA-II and on the interactive *Reference Direction Method* [30]. In RD-NSGA-II, according to a reference point provided by the DM and a starting

⁽¹⁾We mean by 'cognitive effort' or 'cognitive burden' the DM's effort required to understand the information provided by an algorithm, interpret its meaning and learn from it.

objective vector, a reference direction is defined by the difference of both vectors. Then, the individuals at each generation are classified into several fronts using an achievement scalarizing function and a set of equidistant reference points lying on the reference direction. In the proposed classification, the individuals having the smallest values of the achievement scalarizing function for each one of the reference points are preferred.

In the *Light Beam Search based EMO* [11], NSGA-II is hybridized with the classical *Light Beam Search Method* [28]. In this approach, the DM provides a reference direction and a veto threshold vector, which is used to find possibly interesting neighbouring objective vectors around the point defined by the reference direction. Thiele et al [47] proposed a modification of the *Indicator Based Evolutionary Algorithm* (IBEA) [52] called the *Preference Based Evolutionary Algorithm* (PBEA), where the binary quality indicator of IBEA is redefined using an achievement scalarizing function and one or several reference points. Later, Figueira et al [21] made use of PBEA in the *Parallel Multiple Reference Point Evolutionary Algorithm* (PMRPEA). In this method, the area between the worst and the best (preferred) objective values given by the DM is explored by solving multiple PBEAs in parallel.

A progressively interactive EMO algorithm is proposed by Deb et al [16]. The search for the most preferred solution is done by accepting preference information from the DM progressively, which is used to build a value function. Next, an extension of this work based on polyhedral cones has been published in [46]. After a predefined number of generations, a polyhedral cone is built using a solution selected by the DM and the domination criterion is modified in order to focus the search for objective vectors towards the most interesting region.

In [39], a new dominance relation called *g-dominance* is suggested based on a reference point. In the new relation, objective vectors satisfying all the aspiration values and objective vectors fulfilling none of them are preferred over objective vectors satisfying some of the aspiration levels. The main advantage of the *g-dominance* is that it can be incorporated in any metaheuristic algorithm, given that it does not modify the architecture of the algorithm. However, the *g-dominance* does not preserve the Pareto dominance, as said in [42], and a dominated objective vector may be preferred to the objective vector that dominates it.

Another concept of dominance based on reference points, called the *r-dominance*, is proposed in [42]. This new dominance relation distinguishes objective vectors according to their Euclidean distance to a reference point given by the DM, emphasizing those closer to the reference point. A parameter $\delta \in [0, 1]$ controls the selection pressure. In [42], the performance of the new relation is shown replacing in NSGA-II the usual Pareto dominance by the *r-dominance*, and they refer to the modified algorithm as *r-NSGA-II*.

Recently, a *Preference-based Interactive Evolutionary* (PIE) algorithm was proposed in [45]. Starting from a solution selected by the DM from a randomly generated population, this algorithm progressively improves the objective function values, until the most preferred solution is found. The idea behind this approach is the minimization of an achievement scalarizing function at each iteration by using a single-objective evolutionary algorithm. In addition, the PIE algorithm stores all the solutions of the evolutionary algorithm in an archive set, what allows the DM to retrieve previously generated solutions if (s)he wants to go backwards. Another recently proposed approach is the *interactive MOEA/D* (iMOEA/D) [24], which interactively modifies (MOEA/D) [50].

In this paper, we propose a new preference-based EMO algorithm called *Weighting Achievement Scalarizing Function Genetic Algorithm* (WASF-GA). It considers the DM's preferences expressed by means of aspiration levels, that is, objective function values that

are desirable for the DM and which constitute a so-called reference point. The main purpose of WASF-GA is to generate a well-distributed set of nondominated objective vectors approximating the region of interest of the Pareto optimal front defined by the reference point.

The philosophy of the proposed approach is based on the use of an achievement scalarizing function (ASF) [37, 48] and on the classification of the individuals into several fronts, as in NSGA-II. In general, an ASF can be minimized in order to find the Pareto optimal solution whose objective function values adjust the reference point "as best as possible". However, the Pareto optimal solution obtained highly depends on a vector of weights used in the ASF, whose role can vary from purely normalizing coefficients to parameters expressing the DM's preferences [32, 41].

In WASF-GA, we consider a sample of weight vectors in $(0, 1)^k$. Then, at each generation, the classification of the individuals into different fronts is done according to the values that each solution takes on the ASF considered, using the reference point given by the DM, for each one of the weight vectors in the sample. The use of an appropriate set of weight vectors in the ASF for the reference point enables us to obtain objective vectors which approximate the region of the Pareto optimal front that best adjusts the reference point. We will refer to this region as the *region of interest defined by the reference point*.

WASF-GA is somehow similar to the algorithm RD-NSGA-II previously introduced, since both methods are based on reference points and use the values of an ASF to classify the individuals at each generation into several fronts. However, they differ in the following: RD-NSGA-II fixes the weight vector used in the ASF and considers a set of reference points lying on the reference direction, while WASF-GA fixes the reference point in the ASF and employs a set of weight vectors. In practice, WASF-GA is aimed at generating a set of solutions whose objective vectors approximate the region of interest of the Pareto optimal front defined by the reference point. In contrast, RD-NSGA-II finds a set of solutions whose objective vectors approximate only the projection of the reference direction onto the Pareto optimal front. In problems with three or more objective functions, the region of interest approximated by WASF-GA (a subspace of R^k of dimension $k - 1$) is totally different from the one approximated by RD-NSGA-II (a subspace of R^k of dimension 1).

The rest of this paper is organized as follows. In Section 2, we introduce the main concepts and notations used. The new method WASF-GA is motivated and described in Section 3. Section 4 is devoted to the analysis of the performance of the new method in several test problems. We compare the populations generated by WASF-GA with the ones obtained by other preference-based EMO methods. Later, we discuss the advantages of WASF-GA in compared to the algorithms considered in the computational tests in Section 5. Finally, the conclusions are given in Section 6.

2 Concepts and Notation

A *multiobjective optimization problem* can be mathematically written in the form

$$\begin{aligned} & \text{minimize} && \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ & \text{subject to} && \mathbf{x} \in S, \end{aligned} \tag{1}$$

where $f_i : S \rightarrow \mathbf{R}$, for $i = 1, \dots, k$ ($k \geq 2$) are the conflicting *objective functions* that must be minimized simultaneously, and $S \subset \mathbf{R}^n$ is the *feasible set*. The n -dimensional vectors $\mathbf{x} = (x_1, x_2, \dots, x_n)^T \in S$ are referred as *solutions* or *decision vectors*, and their images $\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x}))^T$ are called *objective vectors*. The image of the feasible set in the *objective space* \mathbf{R}^k is called the *feasible objective region* $Z = \mathbf{f}(S)$.

As mentioned earlier, it is not possible to find a solution where all the objectives can reach their individual optimum, but there is a set of solutions that are mathematically equivalent, those solutions where none of their objective values can be improved without deteriorating at least one of the others. We say that a decision vector $\mathbf{x} \in S$ is a *Pareto optimal solution* 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 *Pareto optimal objective vector*. The set of all the Pareto optimal solutions is called the *Pareto optimal set* and it is denoted by E , and the set of all the Pareto optimal objective vectors is called the *Pareto optimal front*, and it is denoted by $\mathbf{f}(E)$. A decision vector $\mathbf{x} \in S$ is said to be a *weakly Pareto optimal solution* if there does not exist another $\mathbf{x}' \in S$ such that $f_i(\mathbf{x}') < f_i(\mathbf{x})$ for all $i = 1, \dots, k$. The corresponding objective vector $\mathbf{f}(\mathbf{x})$ is called *weakly Pareto optimal objective vector*. Besides, a decision vector $\mathbf{x} \in S$ is said to be a *properly Pareto optimal solution* if it is Pareto optimal and if there is a real number $M > 0$ such that, for each i and each $\mathbf{x}' \in S$ satisfying $f_i(\mathbf{x}') < f_i(\mathbf{x})$, there exists at least one j such that $f_j(\mathbf{x}) < f_j(\mathbf{x}')$ and $\frac{f_i(\mathbf{x}) - f_i(\mathbf{x}')}{f_j(\mathbf{x}') - f_j(\mathbf{x})} \leq M$. The corresponding objective vector $\mathbf{f}(\mathbf{x})$ is called *properly Pareto optimal objective vector*. The set of properly Pareto optimal solutions is included in the Pareto optimal set, which is, in turn, included into the set of weakly Pareto optimal solutions.

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 $i = 1, \dots, k$ and there exists one j such that $z_j < z'_j$. In the context of EMO algorithms, the subset of solutions in a population whose objective vectors are not dominated by any other objective vector of the population is called the nondominated set. EMO algorithms usually aim at generating nondominated objective vectors representing the Pareto optimal front as well as possible (i.e., being close to the Pareto optimal front).

Commonly, it is useful to know the ranges of objective vectors in the Pareto optimal front. Upper and lower bounds for the objective functions are defined by the *ideal objective vector* and the *nadir objective vector*, which represent the best and the worst values that each objective function can reach in the Pareto optimal front, respectively. The components of the *ideal objective vector* $\mathbf{z}^* = (z_1^*, \dots, z_k^*)^T \in \mathbf{R}^k$ are obtained by minimizing each objective function individually in the feasible set, that is, $z_i^* = \min_{\mathbf{x} \in S} f_i(\mathbf{x}) = \min_{\mathbf{x} \in E} f_i(\mathbf{x})$ for all $i = 1, \dots, k$. The *nadir objective vector* $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \dots, z_k^{\text{nad}})^T$ can be defined as $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ for all $i = 1, \dots, k$. Often, the nadir objective vector is difficult to obtain, so it is estimated [12, 13, 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 functions.

Because all Pareto optimal solutions can be regarded as equally desirable in the mathematical sense, we need information about the preferences of a DM [35]. A natural way to express preferences consists of specifying desirable objective function values, which constitute the components of the so-called reference point. That is, a *reference point* is given by $\mathbf{q} = (q_1, \dots, q_k)^T$, where q_i is an aspiration value for the objective function f_i provided by the DM, for all $i = 1, \dots, k$. Usually, a reference point is said to be *achievable* if there is a feasible solution whose objective values simultaneously achieve or improve the corresponding reference levels; otherwise, the reference point is said to be *unachievable*.

Many reference points-based MCDM methods use *achievement scalarizing functions* (ASFs), which combine the original objective functions of the problem (1) and the preference information given by the reference point in a single real-valued function. Then, the original problem (1) is transformed into a single-objective optimization problem consisting of the minimization of the ASF over the feasible set. By solving this problem, we can generate all (properly) Pareto optimal solutions of problem (1) [35]. For an overview

about ASFs, see [37].

One of the most used ASF is the one proposed by Wierzbicki in 1980 [48], whose formulation is based on the L_∞ distance. For a given reference point $\mathbf{q} = (q_1, \dots, q_k)^T$, a vector of weights $\mu = (\mu_1, \dots, \mu_k)$, with $\mu_i > 0$ for all $i = 1, \dots, k$, and a parameter $\rho > 0$, the Wierzbicki's ASF is given by:

$$s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu) = \max_{i=1, \dots, k} \{ \mu_i (f_i(\mathbf{x}) - q_i) \} + \rho \sum_{i=1}^k \mu_i (f_i(\mathbf{x}) - q_i), \quad (2)$$

which must be minimized over S :

$$\begin{aligned} & \text{minimize} && s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu) \\ & \text{subject to} && \mathbf{x} \in S. \end{aligned} \quad (3)$$

The weight μ_i given to each objective function f_i must be strictly positive. The weights can be defined as normalizing coefficients [41], but they can also have a preferential meaning [32]. The parameter $\rho > 0$, which must be a small positive value, is the so-called *augmentation coefficient* and it is used to assure that the optimal solution of (3) is a Pareto optimal solution of problem (1) for any reference point. Besides, this term can also improve the computational efficiency [38]. If the ranges of the objective functions are in very different scales, the values of the objective functions must be normalized. Commonly, they can be normalized in (2) dividing $f_i(\mathbf{x}) - q_i$ by $z_i^{\text{nad}} - z_i^*$, for every $i = 1, \dots, k$, although there are other types of normalization.

For a given reference point \mathbf{q} , minimizing an ASF means to find the "best" Pareto optimal objective vector with respect to \mathbf{q} and the weight vector μ . This implies that the solution obtained does not only depend on the reference point considered, but also on the vector of weights used. Indeed, for a given reference point, Pareto optimal objective vectors generated using different weights tend to be different, as several studies have shown [32, 36, 37, 41]. In practice, if the reference point is unachievable, minimizing the ASF in (2) is equivalent to the minmax distance, while if the reference point is achievable, the objective values in the optimal solution of (3) improve the given reference levels in the best possible way. Furthermore, solving (3) means to project \mathbf{q} onto the Pareto optimal front in the direction determined by the inverse of the weights.

Now then, once a DM gives a reference point \mathbf{q} , which Pareto optimal objective vectors are the most interesting ones according to \mathbf{q} ? It seems logical that, if \mathbf{q} is achievable, the most interesting ones are those that dominate the reference point, namely, the subset of Pareto optimal objective vectors which do not deteriorate any coordinate of the reference point. When \mathbf{q} is unachievable, the subset of the most interesting Pareto optimal objective vectors is not so accurately identified because, in this case, no objective vector improves all the reference levels at the same time. However, we may suppose that the probably most interesting ones are those Pareto optimal objective vectors dominated by the reference point. This is motivated because the Pareto optimal objective vectors that are not dominated by the reference point may improve one or several reference levels at the expense of a sacrifice of the other reference levels.

Then, we define the *region of interest of the Pareto optimal front defined by \mathbf{q}* in the following way. When \mathbf{q} is achievable, the region of interest is the subset of Pareto optimal objective vectors which dominate \mathbf{q} , that is, the 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 which are dominated by \mathbf{q} , that is, the objective vectors $\mathbf{f}(\mathbf{x})$ with $\mathbf{x} \in E$ which verify that $f_i(\mathbf{x}) \geq q_i$, for

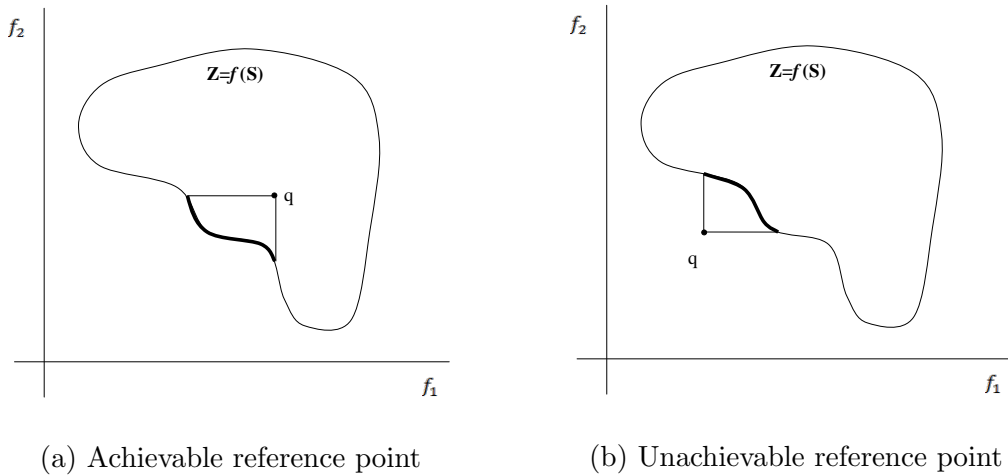


Figure 1: Region of interest of the Pareto optimal front for a reference point.

every $i = 1, \dots, k$. Note that, in most problems, the region of interest considered in the unachievable case is likely to be constituted by the closest Pareto optimal objective vectors to the reference point regarding the minmax distance, obtained by varying the weight vector in the entire weight vector space. However, it should be mentioned that this is not always true.

Figure 1 gives a graphical idea of the region of interest for achievable and unachievable reference points in a biobjective optimization problem as (1). The region of interest has been highlighted with a bold line in both cases. In the achievable case, we assume that the DM is interested in the Pareto optimal objective vectors lying inside the region of interest, given that, as said before, these objective vectors improve all the aspiration values at the same time, without having to impair any of them. When the reference point is unachievable, no Pareto optimal objective vector dominates the reference point. In that case, although the Pareto optimal objective vectors lying inside the region of interest do not improve any aspiration value, they deteriorate them as little as possible. It can be seen that the Pareto optimal objective vectors outside the region of interest may improve some of the aspiration values, but at the expense of a sacrifice of other ones. Furthermore, the higher the improvement of an aspiration value, the higher is the sacrifice of the others.

3 The Weighting Achievement Scalarizing Function Genetic Algorithm (WASF-GA)

This section is devoted to explain the main features of the proposed algorithm, called *Weighting Achievement Scalarizing Function Genetic Algorithm* (WASF-GA). As previously mentioned, the main purpose of WASF-GA is to approximate the region of interest of the Pareto optimal front defined by a reference point given by the DM.

Any (properly) Pareto optimal objective vector can be found by minimizing the ASF given in (2) for different reference points and vectors of weights [35]. In particular, when a DM gives a reference point, any Pareto optimal objective vector in our region of interest can be found by minimizing (2) over S for that reference point and varying the vector of weights in the whole weight vector space $(0, 1)^k$. Based on this, the idea behind the proposed algorithm is to approximate the region of interest by minimizing (2) at each

generation using the reference point given by the DM and taking into account a sample of weight vectors.

The diversity of the nondominated objective vectors generated by WASF-GA depends on the set of weight vectors considered. As pointed out in Section 2, when the ASF given in (2) is minimized over S using strictly positive weights, in practice, the reference point is projected onto the Pareto optimal front in the direction defined by the inverses of the weights used. Because of that, the weight vectors considered in WASF-GA verify that the vectors formed by their inverse components are evenly distributed in the weight vector space $(0, 1)^k$. In that way, by minimizing (2) at each generation for these weight vectors, in practice, WASF-GA projects the reference point onto the region of interest taking into account a sample of evenly distributed projection directions.

Let us denote by N the size of the population at each generation, N_μ the number of weight vectors in the sample, h the generation number, P^h the population of individuals at generation h and F_n^h the n -th front at generation h . The main steps of the WASF-GA algorithm are described in Subsection 3.1. At each generation, the individuals of the population formed by parents and offsprings (of size $2N$) are divided into different fronts. First of all, let us assume that any feasible solution is better than any infeasible solution. Consequently, we classify first the feasible individuals and, later, the infeasible ones.

For the classification of the feasible solutions, we will consider the ASF given in (2) for the reference point given by the DM and the set of N_μ weight vectors obtained according to the procedure described in Subsection 3.2. Then, the feasible solutions are divided into several fronts according to the values that they take on the ASF for the different weight vectors in the set. Specifically, the first front is formed by the feasible individuals with the lowest ASF values for each one of the weight vectors. Next, the feasible solutions with the next lowest ASF values for each one of the weight vectors are passed to the second front, and so on until all the feasible solution have been included into some front.

Once the feasible solutions have been classified, the infeasible ones are next classified taking into account their overall constraint violation. Obviously, the smaller the constraint violation, the better the solution is. Then, the infeasible solution with the lowest constraint violation is placed in the next front; subsequently, the infeasible solution with the next lowest constraint violation forms the following front, and so on until all the infeasible solutions have been classified. This constraint handling procedure is similar to the ones used by other EMO algorithms, as NSGA-II [14].

Finally, the population for the next generation (of size N) is formed by the solutions in the lowest level fronts. If there are more solutions in the last front considered than the remaining space in the new population, the solutions with the lowest ASF values are selected. Following this procedure, when all the solutions of the population are feasible (after several generations), each front is formed by N_μ solutions, except the last one that may have less than N_μ solutions. Consequently, if we want to have, at least, one solution in the new population corresponding to each one of the weight vectors, it is convenient to consider a value of N_μ less or equal than N .

The final population of WASF-GA is formed by the solutions in the first front of the last generation, thus N_μ solutions are shown to the DM. Consequently, the higher N_μ , the larger is the final population generated by WASF-GA. This means that the parameter N_μ has an understandable meaning, and, if so desired, the DM can easily set it by indicating the number of solutions (s)he wants to have in the final population. However, N_μ can be defined as the population size by default.

3.1 Step-by-step of the WASF-GA algorithm

Step 1. Generation of the sample of weight vectors. Following the procedure described in Subsection 3.2, generate N_μ weight vectors, denoted by μ^j for every $j = 1, \dots, N_\mu$.

Step 2. Initialization. Set $h = 0$ and let P^0 a population of N randomly created individuals.

Step 3. Mutation and crossover. Apply the crossover and mutation operators considered to P^h (population of parents), to generate N new individuals (population of offsprings). Let P the combined population of parents and offsprings, with $2N$ individuals.

Step 4. Classification of the individuals into fronts. For every feasible solution $\mathbf{x} \in P$, calculate $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$ for every $j = 1, \dots, N_\mu$. Then, for each $j = 1, \dots, N_\mu$, include in F_1^h and remove temporarily from P the feasible solution in P with the minimum value of $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$ ⁽²⁾. Subsequently, considering the rest of solutions in P , for each $j = 1, \dots, N_\mu$, include in F_2^h and remove temporarily from P the feasible solution in P with the next minimum value of $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$. This process is repeated until every feasible individual has been included into some front. When all the feasible solutions have been classified, the infeasible solutions are classified according to their overall constraint violation.

Step 5. Selection. Set $P^{h+1} = \emptyset$ and $n = 1$. If $\#(P^{h+1} \cup F_n^h) \leq N$, then $P^{h+1} = P^{h+1} \cup F_n^h$ ($\#(A)$ denotes the number of elements in a set A). Update $n = n + 1$ and repeat that process until $\#(P^{h+1} \cup F_n^h) > N$, in whose case the individuals from F_n^h which reach the lowest values of (2) will be included in P^{h+1} until P^{h+1} has N individuals.

Step 6. Stopping criterion. If the stopping criterion is fulfilled, the individuals in F_1^h represents the outcome of WASF-GA and *Stop*. Otherwise, set $h = h + 1$ and go to step 3.

3.2 Generation of the sample of weight vectors

Let $\{\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^{N_\mu}\}$ be a set of well-distributed weight vectors in $(0, 1)^k$ and let us denote by $\{\mu^1, \mu^2, \dots, \mu^{N_\mu}\}$ the sample of weight vectors considered in WASF-GA. Once the well-distributed weight vectors in $(0, 1)^k$ are obtained, for every $j = 1, \dots, N_\mu$, the weight vector $\mu^j = (\mu_1^j, \dots, \mu_k^j)$ considered in WASF-GA is defined by:

$$\mu_i^j = \frac{1}{u_i^j}, \quad \text{for every } i = 1, \dots, k. \quad (4)$$

Note that, in order to assure that the vectors in $\{\mu^1, \mu^2, \dots, \mu^{N_\mu}\}$ belong to $(0, 1)^k$, they are normalized dividing each component by the sum of all the components.

In what follows, the proposed procedure to generate the weight vectors $\{\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^{N_\mu}\}$ is described, which will depends on the number of objective functions. It must be noted that any procedure to generate well-distributed weight vectors in $(0, 1)^k$ could be used, such as i.e. the generation of weights used in MOEA/D [50].

⁽²⁾To maintain the diversity of the population, repeating solutions in the same front is not allowed in case one solution reaches the lowest ASF value for several weight vectors at the same time. That is, once a solution is selected as the one with the lowest value of $s(\mathbf{q}, \mathbf{f}(\mathbf{x}), \mu^j)$ for one $j = 1, \dots, N_\mu$, it is removed and cannot be chosen for another index $r > j$.

3.2.1 Two objective functions

In this case, a set of N_μ weight vectors equally distributed in $(0, 1)^2$ can be generated in the following way. For each $j = 1, \dots, N_\mu$, $\mathbf{u}^j = (u_1^j, u_2^j)$ is defined by:

$$\begin{aligned} u_1^j &= \varepsilon + (j - 1) \frac{1 - 2\varepsilon}{N_\mu - 1} \\ u_2^j &= 1 - u_1^j \end{aligned}$$

where ε is a small positive value. We suggest to use $\varepsilon = 0.01$ in order not to move too far from the extremes of the interval $(0, 1)$. However, this value can be changed if so desired. It can be observed that the value of N_μ defines the size of the step from one weight to the next one in $(0, 1)$, which is $\frac{1 - \varepsilon}{N_\mu - 1}$.

We would like to remark that the generation of these weight vectors only depends on the number N_μ , but not on the problem to be solved. That is, once $\{\mathbf{u}^1, \mathbf{u}^2, \dots, \mathbf{u}^{N_\mu}\}$ have been generated for a biobjective problem, they can be used to solve any other biobjective problem using WASF-GA with N_μ vectors of weights.

3.2.2 Three or more objectives

In problems with three or more objectives ($k \geq 3$), it is more complicated to generate a set of N_μ vectors equally distributed along $(0, 1)^k$. Basically, the procedure described here initially generates a large enough set of weight vectors in $(0, 1)^k$ and, subsequently, the N_μ most representative vectors of this set are selected as the weight vectors needed.

To be more specific, equally distant weights in $(0, 1)$ are first generated, which will be the components of the vectors of weights. To do so, we consider two sufficiently small positive real parameters ε and S , where ε is used to avoid having weights equal to 0 and S is the size of the step between two consecutive weights. Although they can be changed if so desired, we suggest to use $\varepsilon = 0.01$ and $S = 0.03$. Now, we generate the following weights in $(0, 1)$:

$$u^r = \varepsilon + r \cdot S \text{ for every } r = 0, \dots, L,$$

where $L = E(\frac{1 - 2\varepsilon}{S})$ ($E(\cdot)$ denotes the integer part of a number). Next, we consider the set of weight vectors in $(0, 1)^k$ for which each weight component is equal to one u^r , for $r = 1, \dots, L$. Subsequently, these weight vectors are normalized, dividing each component by the sum of all the components. Let us refer to the resulting set of normalized weight vectors as W , which contains L^k vectors. Finally, the set W is divided into N_μ clusters by using e.g. the k -means clustering [34], and the centroid weight vectors of the clusters are the N_μ vectors of weights needed.

As in the previous case, the sample of N_μ weight vectors generated for a problem with $k \geq 3$ objectives does not depend on the problem itself, and it can be used in WASF-GA to solve any other problem with k objectives using N_μ vectors of weights. Besides, the set of randomly generated weight vectors W can be reused whenever we want to run WASF-GA for another value of N_μ .

3.3 Properties

If repeating solutions in the same front is allowed, a feature of the new algorithm is that the classification of the individuals in WASF-GA into several fronts defines a strict partial order which is complete with the Pareto dominance. Firstly, let \prec be the binary relation defined over a finite set of solutions P , where $\mathbf{x} \prec \mathbf{y}$ means that \mathbf{x} belongs to a lower level front of WASF-GA than \mathbf{y} . The relation \prec defines a strict partial order in P given that:

(a) it is irreflexive (for all $\mathbf{x} \in P$, it is clear that $\mathbf{x} \not\prec \mathbf{x}$); (b) it is asymmetric (if $\mathbf{x} \prec \mathbf{y}$, \mathbf{x} belongs to a lower level front than \mathbf{y} and, consequently, $\mathbf{y} \not\prec \mathbf{x}$); and (c) it is transitive (if $\mathbf{x} \prec \mathbf{y}$ and $\mathbf{y} \prec \mathbf{z}$, the \mathbf{x} belongs to a lower level front than \mathbf{y} , which is, in turn, in a lower level front than \mathbf{z} ; then, \mathbf{x} belongs to a lower level front than \mathbf{z} and, consequently, $\mathbf{x} \prec \mathbf{z}$).

Secondly, let us prove that \prec is *complete* with the Pareto dominance defined in Section 2. In general, if \triangleright is a binary relation where $\mathbf{x} \triangleright \mathbf{y}$ implies that \mathbf{x} is preferred to \mathbf{y} , \triangleright is said to be *complete with the Pareto dominance* if and only if \mathbf{x} Pareto dominates \mathbf{y} implies that $\mathbf{x} \triangleright \mathbf{y}$ [55]. In our case, let us consider $\mathbf{x}^1, \mathbf{x}^2 \in P$ such that \mathbf{x}^1 Pareto dominates \mathbf{x}^2 (being both of them feasible) and let us see that $\mathbf{x}^1 \prec \mathbf{x}^2$. If \mathbf{x}^1 Pareto dominates \mathbf{x}^2 , we can say that $s(\mathbf{q}, \mathbf{f}(\mathbf{x}^1), \mu) < s(\mathbf{q}, \mathbf{f}(\mathbf{x}^2), \mu)$, for any strictly positive weight vector μ [35], and, in particular, for each one of the N_μ weight vectors considered in WASF-GA. Then, it is obvious that \mathbf{x}^1 belongs to a lower level front of WASF-GA than \mathbf{x}^2 , so $\mathbf{x}^1 \prec \mathbf{x}^2$.

4 Computational Tests

In order to evaluate the performance of WASF-GA, we have carried out a series of computational experiments. Firstly, we have compared WASF-GA with two other preference-based EMO algorithms also based on the reference point preferential scheme: r-NSGA-II [42] and R-NSGA-II [17]. Although these algorithms approximate a region of the Pareto optimal front that fits the reference point used, they are not technically designed to approximate just what we have considered as the region of interest approximated by WASF-GA. However, as it is shown hereafter, the parameters which r-NSGA-II and R-NSGA-II depend on have been adjusted to generate final populations whose objective vectors approximate our region of interest as well as possible. Because of this inconvenience, we have carried out a second set of experiments, given that we can find in the literature other techniques that also exclusively return nondominated objective vectors in our region of interest, as [22, 39]. In these second tests, WASF-GA has been compared with the following reference point-based EMO algorithms: the NSGA-II version that replaces the usual Pareto dominance by the g-dominance proposed in [39], which we will refer to as g-NSGA-II, and the preference-based MOGA algorithm proposed in [22], which we will refer to as P-MOGA. All the algorithms have been implemented by the same software developer using jMetal [19], a Java-based framework for multi-objective optimization, and we have carried out 30 independent runs on the same computer (Intel Core2 Duo E7600 @3.06GHz, 4.00GB).

We have considered problems with 2 and 3 objective functions and we have used achievable and unachievable reference points. We have used a population size of 200 individuals and 300 generations for the problems with 2 objectives, and a population size of 300 individuals and 400 generations for the 3-objective problems. The SBX crossover operator and the polynomial mutation [9] have been considered, with a distribution index of 20 and 20, respectively. The crossover probability has been set to 0.9 and the mutation probability to $1/n$, where n is the number of variables. In our tests, the objective values calculated in the ASF given in (2) and in the Euclidean distances computed in r-NSGA-II and R-NSGA-II have been normalized as indicated in Section 2. The ideal and the nadir objective vectors can be estimated using any suitable technique. Given that there are approximations of the Pareto optimal fronts of the problems considered elsewhere, we have estimated these boundary vectors using the best and the worst objective function values reached in these approximations.

Regarding WASF-GA, in all test problems, we have considered the number of weight vectors, N_μ , equal to the size of the population used. For the generation of the weight vectors following the procedure described in Subsection 3.2, we have used the values of ε

and S recommended there. That is, $\varepsilon = 0.01$ for the 2-objective problems and $\varepsilon = 0.01$ and $S = 0.03$ for the 3-objective problems. Let us remember that the sample of N_μ weight vectors does not depend on the problem itself, but on the number of objectives.

In order to analyse the performance of each algorithm, we have compared: (a) the quality of the final populations generated by each algorithm using the metric based on the hypervolume described in Subsection 4.1; (b) the computational complexity of each algorithm; and (c) the number of solutions in the final population whose objective vectors lie in the region of interest. For each test problem, we show the mean of the obtained results over the 30 independent runs. In order to statistically analyse the results obtained in the metric based on the hypervolume, we have performed the Wilcoxon rank-sum test [49]. This is a non-parametric statistical hypothesis test, which permits us to make pairwise comparisons between algorithms to study the significance of the obtained results.

4.1 Hypervolume of the Region of Interest defined by \mathbf{q}

The quality of the final population has been measured using a metric we have defined based on the hypervolume indicator [23, 54]. For a set of objective vectors P , the *hypervolume of P* , denoted by $HV(P)$, can be defined as the hypervolume of the portion of the objective space that is dominated by the objective vectors in P and is bounded by an (achievable) reference point, $\mathbf{R} = (r_1, \dots, r_k)$. The higher the hypervolume, the better is the population.

For a population of objective vectors P and a reference point \mathbf{q} , let us denote by $HV_{\mathbf{q}}(P)$ the *hypervolume of P in the region of interest defined by \mathbf{q}* . $HV_{\mathbf{q}}(P)$ is defined as the hypervolume of the subset of objective vectors in P which lies inside the region of interest of the Pareto optimal front for \mathbf{q} , defined in Section 2 (we will refer to this subset of P as $P_{\mathbf{q}}$). Then, $HV_{\mathbf{q}}(P) = HV(P_{\mathbf{q}})$. In order to calculate $HV_{\mathbf{q}}(P)$, the reference point \mathbf{R} needed to compute $HV(P_{\mathbf{q}})$ is set in the following way:

- If \mathbf{q} is achievable, then $\mathbf{R} = \mathbf{q}$, that is, $r_i = q_i$ for all $i = 1, \dots, k$.
- If \mathbf{q} is unachievable, then \mathbf{R} is defined as follows. Let A be a reference set, which is a good approximation of the Pareto optimal front. Then, $r_i = \max_{\mathbf{x} \in A_{\mathbf{q}}} f_i(\mathbf{x})$ for all $i = 1, \dots, k$, where $A_{\mathbf{q}}$ is the subset of objective vectors in A which lies inside the region of interest for \mathbf{q} ($A_{\mathbf{q}} = \{\mathbf{z} \in A \text{ with } z_i \geq q_i \text{ for every } i = 1, \dots, k\}$).

Figure 2 gives a graphical idea of the area whose hypervolume is calculated in the $HV_{\mathbf{q}}$ metric for a biobjective problem.

The $HV_{\mathbf{q}}$ values which are shown hereafter in Subsections 4.2 and 4.3 have been normalized taking into account the hypervolume of the true region of interest, that is, taking into account $HV(A_{\mathbf{q}})$. The Pareto optimal fronts of the problems considered in the computational tests are available elsewhere.

4.2 WASF-GA versus r-NSGA-II and R-NSGA-II

Here, we will show the results obtained by WASF-GA, r-NSGA-II [42] and R-NSGA-II [17] for achievable and unachievable reference points on several test problems.

In r-NSGA-II, a threshold $\delta \in [0, 1]$ controls the selection pressure of the r-dominance, which is stronger for values of δ closer to 0, and softer for values closer to 1. In [42], the authors proposed a procedure that adjusts δ adaptively during the process, by which δ is initially set as 0 and it progressively varies until reaching a given value $\delta_0 \in [0, 1]$. The idea is to guide the search gradually during the evolution process towards the region of

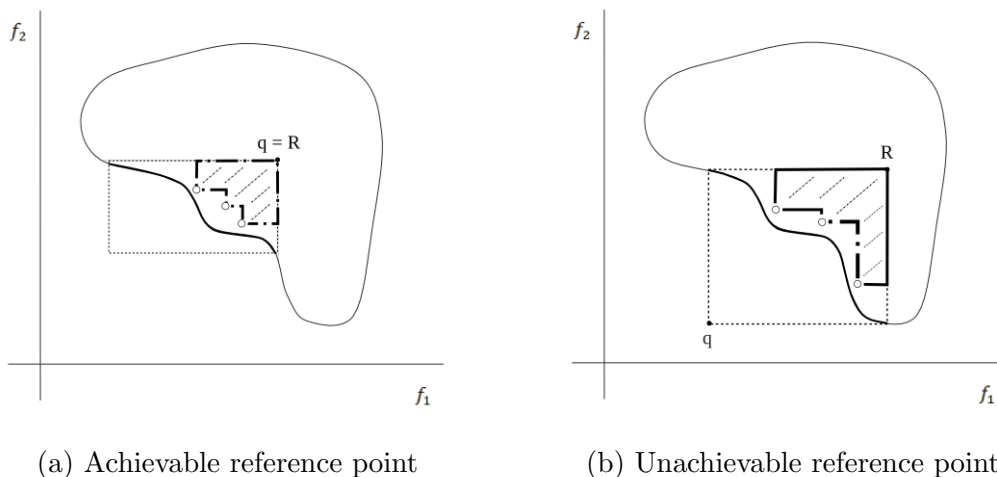


Figure 2: Graphical idea of $HV_{\mathbf{q}}$ in a 2-objective problem.

interest, avoiding a lack of diversity and a premature convergence. We have used this adaptive procedure in r-NSGA-II in our experiments. On the other hand, an ϵ -based selection strategy is used in R-NSGA-II to control the diversity. For a small positive quantity ϵ , this strategy makes that all objective vectors having a distance of ϵ or less between them are represented by just one of them. The larger the value of ϵ , the more extended is the generated set of nondominated objective vectors.

Given that the spread of the final populations of r-NSGA-II and R-NSGA-II depends on δ_0 and ϵ , their values have been adjusted on each case so that to obtain final populations which can be compared to the final population of WASF-GA. That is, since the purpose of WASF-GA is to approximate the region of interest defined by the reference point, we have adjusted the values of δ_0 and ϵ in order to generate final populations whose objective vectors approximate this region.

We have considered the 2-objective ZDT1, ZDT2 and ZDT3 with 30 variables each one [51] and the 3-objective problems DTLZ2 and DTLZ7 with 12 and 22 variables [18], respectively. For each test problem, Table 1 shows the reference points and the values of the parameters δ_0 and ϵ used in r-NSGA-II and in R-NSGA-II, respectively. Given that it has been tedious to set δ_0 and ϵ for each test problem and each reference point in order to fit the region approximated to our region of interest, we decided that these test problems were enough to show the performance of the three algorithms.

Problem	Obj.	Achievable			Unachievable		
		\mathbf{q}	δ	ϵ	\mathbf{q}	δ	ϵ
ZDT1	2	(0.8, 0.6)	0.91	0.0064	(0.2, 0.4)	0.73	0.0016
ZDT2	2	(0.8, 0.8)	0.83	0.0045	(0.5, 0.3)	0.93	0.0049
ZDT3	2	(0.3, 0.8)	0.8405	0.0018	(0.2, 0)	0.71	0.0016
DTLZ2	3	(0.6, 0.7, 0.7)	0.65	0.07	(0.4, 0.4, 0.4)	0.6	0.078
DTLZ7	3	(0.1, 0.8, 6)	0.8	0.07	(0.1, 0.1, 4)	0.8	0.06

Table 1: r-NSGA-II and R-NSGA-II parameters setting.

The mean and the standard deviation of the $HV_{\mathbf{q}}$ values obtained for the achievable reference points can be seen in Table 2, and the ones for the unachievable reference points in Table 3. We have coloured in dark gray the algorithm which has reached the best $HV_{\mathbf{q}}$ mean and in light gray the one with the second best $HV_{\mathbf{q}}$ mean. From Tables 2 and 3, we can see that WASF-GA has reached the best $HV_{\mathbf{q}}$ mean in all the problems, for both

the achievable and the unachievable cases. Although all the objective vectors generated by WASF-GA lie inside the region of interest, not all the objective vectors generated by r-NSGA-II and R-NSGA-II are inside this area, given that the extension of the region approximated by them is controlled by δ_0 and ϵ , respectively. Then, it is important to know how many objective vectors were considered on each algorithm to calculate $HV_{\mathbf{q}}$. On average, 100% of the objective vectors generated by WASF-GA, 52.6% of the ones obtained by r-NSGA-II and 63.4% of the ones provided by R-NSGA-II were inside the region of interest. We would like to remark that we have tested several values of δ_0 and ϵ before choosing the ones which approximate regions as similar as possible to our region of interest on each case. In practice, one does not know beforehand how wide the region approximated by r-NSGA-II and R-NSGA-II will be according to the values of δ_0 and ϵ . In this regard, we could say that WASF-GA is more robust, given that it always generates objective vectors approximating the region of interest defined by the reference point considered.

Problem	Obj.	WASF-GA	r-NSGA-II	R-NSGA-II
ZDT1	2	$5.61e - 01_{5.6e-07}$	$5.59e - 01_{2.6e-04}$	$5.59e - 01_{3.6e-04}$
ZDT2	2	$4.50e - 01_{7.5e-07}$	$4.49e - 01_{2.5e-04}$	$4.37e - 01_{9.5e-03}$
ZDT3	2	$3.61e - 01_{1.5e-07}$	$3.59e - 01_{2.7e-03}$	$2.48e - 01_{5.1e-02}$
DTLZ2	3	$8.49e - 02_{6.5e-06}$	$4.60e - 02_{4.9e-03}$	$4.85e - 02_{2.5e-03}$
DTLZ7	3	$3.08e - 01_{3.0e-05}$	$2.75e - 01_{9.0e-03}$	$2.75e - 01_{1.0e-02}$

Table 2: Mean and standard deviation of $HV_{\mathbf{q}}$, when \mathbf{q} is achievable.

Problem	Obj.	WASF-GA	r-NSGA-II	R-NSGA-II
ZDT1	2	$5.22e - 01_{7.2e-07}$	$5.19e - 01_{6.9e-04}$	$4.78e - 01_{3.4e-02}$
ZDT2	2	$4.56e - 01_{4.7e-07}$	$4.52e - 01_{4.1e-04}$	$4.49e - 01_{6.2e-03}$
ZDT3	2	$5.32e - 01_{7.2e-07}$	$5.15e - 01_{3.8e-03}$	$5.12e - 01_{7.3e-03}$
DTLZ2	3	$6.86e - 01_{4.2e-04}$	$5.98e - 01_{1.2e-02}$	$5.91e - 01_{9.4e-03}$
DTLZ7	3	$4.96e - 01_{2.3e-04}$	$4.84e - 01_{2.8e-03}$	$4.34e - 01_{5.9e-02}$

Table 3: Mean and standard deviation of $HV_{\mathbf{q}}$, when \mathbf{q} is unachievable.

As previously said, the Wilcoxon rank-sum test has been used on each test problem to determine whether the comparative among the $HV_{\mathbf{q}}$ means of each pair of algorithms is statistically significant or not. Tables 4 and 5 contain the results obtained in the achievable and in the unachievable cases, respectively. We have used a 5% significance level and the null hypothesis is "The two algorithms have the same $HV_{\mathbf{q}}$ mean". That is, the Wilcoxon test is used to determine whether the difference between the $HV_{\mathbf{q}}$ means is statistically significant. If p -value ≤ 0.05 , the null hypothesis is rejected and, then, the $HV_{\mathbf{q}}$ means of the two algorithms are comparable. In this case, in Tables 4 and 5, a symbol \blacktriangle indicates that the algorithm in the row has reached a better $HV_{\mathbf{q}}$ mean than the algorithm in the column, and a symbol ∇ is used if the algorithm in the column has a better $HV_{\mathbf{q}}$ mean than the algorithm in the row. If the null hypothesis is true (p -value ≥ 0.05), a symbol $-$ is used to indicate that the comparative is not statistically significant. According to Tables 4 and 5, we observe that statistical confidence has been found in all the results. From them, we can see that WASF-GA performs better than r-NSGA-II and R-NSGA-II in all the test problems, for the achievable and the unachievable reference points.

Next, we have compared the computational complexities of WASF-GA, r-NSGA-II and R-NSGA-II. Table 6 shows the complexities of their basic operations in one iteration, considering their worst cases. The complexities of r-NSGA-II and R-NSGA-II have been calculated using one reference point. Taking into account that we suggest to set $N_{\mu} \leq N$, the global complexity of WASF-GA is lower or equal than those of the other two methods.

ZDT1, ZDT2, ZDT3 (2 objectives)		
	r-NSGA-II	R-NSGA-II
WASF-GA	▲ ▲ ▲	▲ ▲ ▲
r-NSGA-II		▽ ▲ ▲

DTLZ2, DTLZ7 (3 objectives)		
	r-NSGA-II	R-NSGA-II
WASF-GA	▲ ▲	▲ ▲
r-NSGA-II		▽ -

Table 4: Wilcoxon test in the two and three-objective problems, when \mathbf{q} is achievable.

ZDT1, ZDT2, ZDT3 (2 objectives)		
	r-NSGA-II	R-NSGA-II
WASF-GA	▲ ▲ ▲	▲ ▲ ▲
r-NSGA-II		▲ - ▲

DTLZ2, DTLZ7 (3 objectives)		
	r-NSGA-II	R-NSGA-II
WASF-GA	▲ ▲	▲ ▲
r-NSGA-II		▲ ▲

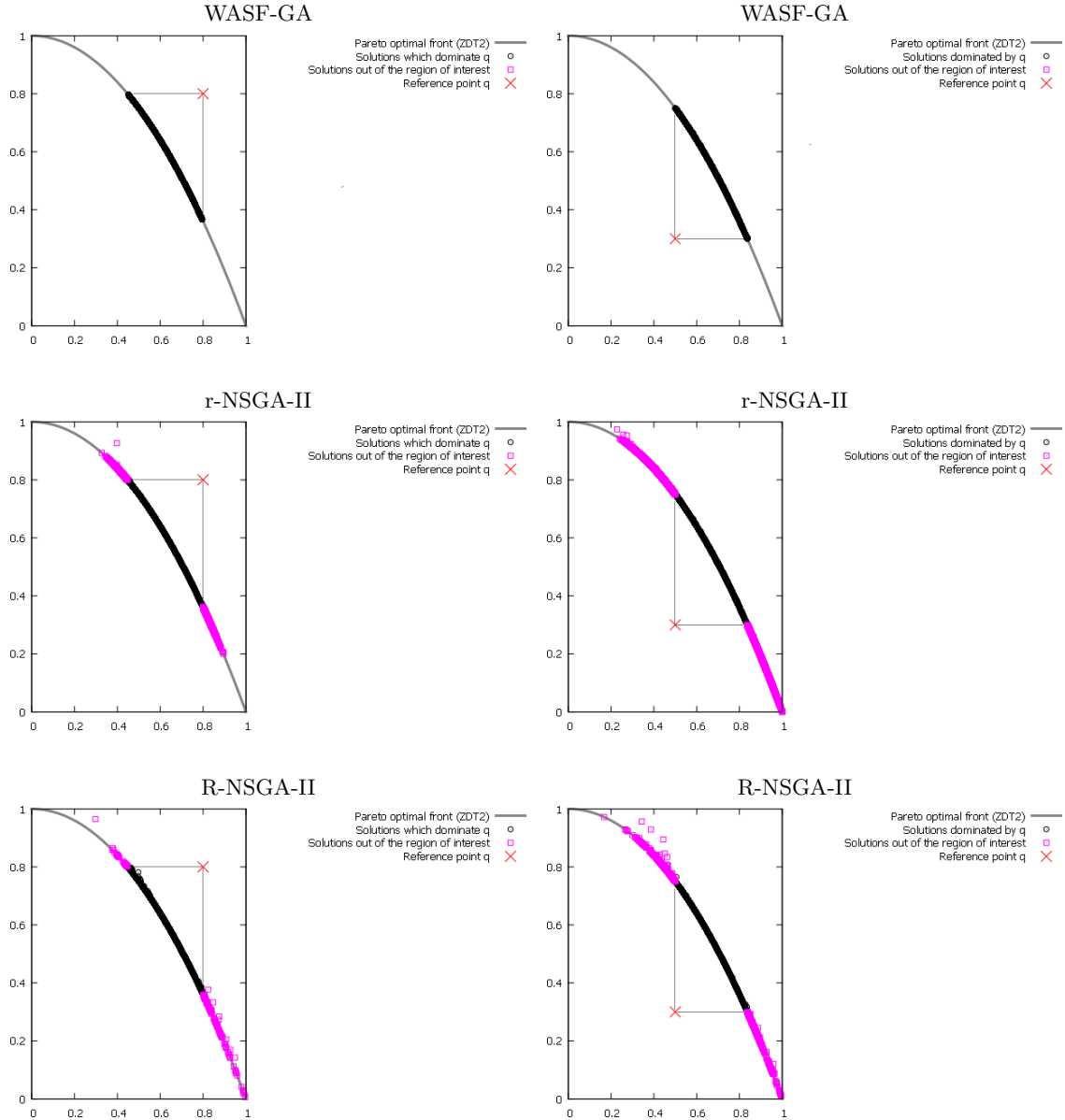
Table 5: Wilcoxon test in the two and three-objective problems, when \mathbf{q} is unachievable.

Algorithm	Basic operations	Complexities	Global Complexity
WASF-GA	ASF assignment Classification into several fronts	$O(k \cdot 2N \cdot N_\mu)$ $O(2N \cdot N_\mu)$	$O(k \cdot N \cdot N_\mu)$
R-NSGA-II	Nondominated sorting Preference distance assignment Preference distance sorting ϵ -based selection strategy	$O(k \cdot (2N)^2)$ $O(k \cdot 2N)$ $O(2N \cdot \log(2N))$ $O(k \cdot (2N)^2)$	$O(k \cdot N^2)$
r-NSGA-II	r-nondominated sorting Crowding distance assignment Crowding distance sorting	$O(k \cdot (2N)^2) + O(k \cdot 2N)$ $O(k \cdot 2N \cdot \log(2N))$ $O(2N \cdot \log(2N))$	$O(k \cdot N^2)$

Table 6: Computational complexities.

Besides, we would like to show the performances of WASF-GA, r-NSGA-II and R-NSGA-II in practice. Due to lack of space, we will plot just the populations generated in the problem ZDT2. The graphics of the other problems are available upon request. The objective vectors provided by each method for ZDT2 in the 30 independent runs and the reference point considered can be seen in Figure 3. In the graphics for the achievable case, we have distinguished between objective vectors dominating the reference point (and which approximate the region of interest) and objective vectors not dominating it. Similarly, in the unachievable case, objective vectors which are dominated by the reference point (and which approximate the region of interest) and objective vectors not dominating it have been represented in a different way. In this test problem, the non-convexity of the Pareto optimal front does not cause any difficulty to the proposed method. In the achievable case, it can be seen that WASF-GA provides a final set of objective vectors that are inside of the region of interest, which means that all the generated objective vectors by WASF-GA improve all the given aspiration values. The results obtained by r-NSGA-II

and R-NSGA-II are similar: they generate objective vectors both inside and outside the region of interest. This implies that these algorithms provide the DM objective vectors that improve the reference point (the ones inside the region of interest) and objective vectors that worsen it (the ones out of the region of interest). Regarding the unachievable reference point, similar results are obtained. WASF-GA generates a set of objective vectors which are distributed along the whole region of interest and the objective vectors found by the other two algorithms lie both inside and outside this region.



Achievable reference point (0.8,0.8).

Unachievable reference point (0.5,0.3).

Figure 3: Objective vectors obtained for the ZDT2 problem.

4.3 WASF-GA versus g-NSGA-II and P-MOGA

Hereafter, we have compared the populations retrieved by WASF-GA to the ones generated by g-NSGA-II (the NSGA-II version that replaces the usual Pareto dominance by the g-dominance [39]) and P-MOGA (the preference-based MOGA algorithm proposed in [22]).

Given that the spread of the final objective vectors of the populations generated by g-NSGA-II and P-MOGA does not depend on any parameter, we have been able to use more test problems than in the previous Subsection 4.2. Then, as benchmark problems, we have chosen problems with 2 and 3 objectives from the families ZDT [51], DTLZ [18], WFG [25] and LZ09 [31], which are designed to represent various complicated situations. Table 7 contains the problems considered and the achievable and unachievable reference points used on each case.

Problem	Obj.	q achievable	q unachievable
ZDT1	2	(0.80, 0.60)	(0.20, 0.40)
ZDT2	2	(0.80, 0.80)	(0.50, 0.30)
ZDT3	2	(0.30, 0.80)	(0.20, 0.00)
ZDT4	2	(0.99, 0.95)	(0.08, 0.25)
ZDT6	2	(0.78, 0.61)	(0.39, 0.21)
DTLZ1	2	(0.41, 0.36)	(0.00, 0.02)
DTLZ2	2	(0.83, 0.92)	(0.07, 0.51)
DTLZ3	2	(0.87, 1.00)	(0.15, 0.42)
DTLZ4	2	(0.97, 0.59)	(0.41, 0.51)
DTLZ5	2	(0.97, 0.59)	(0.03, 0.27)
DTLZ6	2	(0.76, 0.84)	(0.08, 0.48)
DTLZ7	2	(0.85, 3.88)	(0.62, 1.27)
WFG1	2	(1.31, 1.61)	(0.49, 0.88)
WFG2	2	(1.80, 2.91)	(0.23, 0.20)
WFG3	2	(1.75, 2.55)	(0.56, 1.61)
WFG4	2	(1.88, 3.71)	(0.29, 2.93)
WFG5	2	(1.88, 2.46)	(0.47, 1.98)
WFG6	2	(1.46, 3.44)	(1.28, 0.10)
WFG7	2	(1.17, 3.74)	(0.11, 3.03)
WFG8	2	(1.92, 3.60)	(0.29, 3.56)
WFG9	2	(1.83, 3.92)	(0.81, 2.15)
LZ09_F1	2	(0.77, 0.29)	(0.09, 0.01)
LZ09_F2	2	(0.78, 0.38)	(0.49, 0.10)
LZ09_F3	2	(0.98, 0.52)	(0.62, 0.10)
LZ09_F4	2	(0.37, 0.71)	(0.21, 0.24)
LZ09_F5	2	(0.79, 0.81)	(0.05, 0.09)
LZ09_F7	2	(0.91, 0.67)	(0.22, 0.25)
LZ09_F8	2	(0.78, 0.82)	(0.06, 0.57)
LZ09_F9	2	(0.73, 0.90)	(0.25, 0.59)
DTLZ1	3	(0.38, 0.07, 0.41)	(0.90, 0.00, 0.01)
DTLZ2	3	(0.60, 0.70, 0.70)	(0.40, 0.40, 0.40)
DTLZ3	3	(0.37, 0.79, 1.00)	(0.21, 0.31, 0.19)
DTLZ4	3	(1.00, 0.11, 0.67)	(0.75, 0.02, 0.00)
DTLZ5	3	(0.54, 0.65, 0.99)	(0.37, 0.17, 0.54)
DTLZ6	3	(0.67, 0.69, 0.96)	(0.30, 0.40, 0.58)
DTLZ7	3	(0.10, 0.80, 6.00)	(0.10, 0.10, 4.00)
WFG1	3	(0.80, 2.70, 4.93)	(0.17, 0.06, 0.13)
WFG2	3	(1.26, 2.87, 5.12)	(0.07, 0.19, 1.87)
WFG3	3	(0.71, 1.40, 3.41)	(0.27, 0.40, 2.62)
WFG4	3	(0.83, 1.30, 6.00)	(0.03, 0.01, 4.50)
WFG5	3	(1.90, 3.90, 2.87)	(0.83, 2.16, 2.06)
WFG6	3	(1.06, 3.79, 3.35)	(0.45, 0.20, 1.49)
WFG7	3	(1.42, 3.94, 1.95)	(0.23, 0.02, 1.54)
WFG8	3	(1.24, 3.56, 5.81)	(0.40, 2.05, 0.55)
WFG9	3	(1.05, 2.50, 5.95)	(1.35, 0.20, 2.59)
LZ09_F6	3	(0.49, 0.94, 0.57)	(0.21, 0.42, 0.46)

Table 7: Problems and reference points used in WASF-GA, g-NSGA-II and P-MOGA.

In Table 8, we can see the means and standard deviations of the $HV_{\mathbf{q}}$ values obtained

for the achievable case. The results for the unachievable case are included in Table 9. The color of the cells means the same as in Subsection 4.2. On average, the percentage of objective vectors generated by each algorithm inside the region of interest is 96.5% for WASF-GA, 88.48% for g-NSGA-II and 89.35% for P-MOGA. According to Tables 8 and 9, we can say that WASF-GA has reached the best $HV_{\mathbf{q}}$ mean in more problems than g-NSGA-II and P-MOGA. To be more precise, from the 46 problems considered, WASF-GA wins in 33 problems, g-NSGA-II in 4 and P-MOGA in 9 when \mathbf{q} is achievable; in the unachievable case, WASF-GA is the best in 35 problems, g-NSGA-II in 2 and P-MOGA in 9. It can be noted that the $HV_{\mathbf{q}}$ mean obtained by g-NSGA-II is 0 for the 3-objective problem DTLZ1 (achievable case) and for the 3-objective problem DTLZ3 (achievable and unachievable case). In these cases, we observed that g-NSGA-II had convergence problems, and this is because the g-dominance does not preserve the Pareto dominance, and may wrongly emphasize dominated objective vectors over nondominated ones (this fact can be observed, for example, in Figure 4). Regarding the standard deviation, WASF-GA obtained the lowest standard deviation in most of the problems, which means that the results generated by WASF-GA in the 30 independent runs were very similar. This analysis shows that WASF-GA outperforms g-NSGA-II and P-MOGA in terms of the $HV_{\mathbf{q}}$ metric.

The results obtained by the Wilcoxon rank-sum test can be seen in Tables 10 and 11, for the achievable and the unachievable reference points, respectively. In the achievable case, from the 46 problems considered, WASF-GA is better than g-NSGA-II in 32 problems, g-NSGA-II wins in just 5 problems and the difference between their $HV_{\mathbf{q}}$ is not statistically significant in 9 of them. Regarding the results obtained by WASF-GA against P-MOGA, WASF-GA is better in 28, P-MOGA in 7 and their $HV_{\mathbf{q}}$ means are not statistically different in 11 problems. For the unachievable reference points, WASF-GA reaches better $HV_{\mathbf{q}}$ means than g-NSGA-II in 29 problems, g-NSGA-II performs better just in 7 problems and the difference between their $HV_{\mathbf{q}}$ means is not statistically significant in 10 of them. If we check the results of WASF-GA against P-MOGA, we see that WASF-GA is better in 29 cases, P-MOGA in 10 and the $HV_{\mathbf{q}}$ mean difference between the algorithms is not statistically significant in 7 problems.

Regarding the computational complexities, Table 12 shows the complexities of the basic operations of WASF-GA, g-NSGA-II and P-MOGA in one iteration, considering their worst cases. Having in mind that we propose to set $N_{\mu} \leq N$, WASF-GA has a global complexity that is lower or equal than those of the other two methods.

Finally, Figure 4 gives a graphical idea of the performances of WASF-GA, g-NSGA-II and P-MOGA in the LZ09.F6 problem. The graphics of the rest of problems are available upon request. In these graphics, we have plotted the objective vectors obtained for the 30 runs following the same indications stated for Figure 3. In the achievable case, the solutions obtained by the three algorithms are inside the region of interest but, given that the highest $HV_{\mathbf{q}}$ value is obtained by WASF-GA, we could say that the solutions generated by WASF-GA are closer to the Pareto optimal front than the ones provided by g-NSGA-II and P-MOGA. Regarding the unachievable case, it can be seen that the three algorithms generate solutions both inside and outside the region of interest. Specifically, on average, the percentage of solutions inside the region of interest are 67.07% for WASF-GA, 58.51% for g-NSGA-II and 26.52% for P-MOGA. This fact shows that, besides the convergence difficulties of the three algorithms for the unachievable reference point, WASF-GA has better approximated this region of interest. In this case, g-NSGA-II generates many weakly Pareto dominated solutions, which are Pareto dominated, because it is not assure that the g-dominance preserves the Pareto dominance, as said before. Regarding

Problem	Obj.	WASF-GA	g-NSGA-II	P-MOGA
ZDT1	2	$5.61e - 01_{5.9e-07}$	$5.60e - 01_{9.4e-05}$	$5.60e - 01_{8.9e-05}$
ZDT2	2	$4.50e - 01_{8.5e-07}$	$4.50e - 01_{6.9e-05}$	$4.50e - 01_{7.5e-05}$
ZDT3	2	$3.61e - 01_{1.1e-07}$	$3.61e - 01_{3.6e-05}$	$3.61e - 01_{3.7e-05}$
ZDT4	2	$6.47e - 01_{1.4e-04}$	$6.46e - 01_{2.6e-04}$	$6.46e - 01_{1.8e-04}$
ZDT6	2	$4.79e - 01_{6.6e-04}$	$4.77e - 01_{1.0e-03}$	$4.77e - 01_{9.3e-04}$
DTLZ1	2	$4.97e - 01_{2.3e-04}$	$4.96e - 01_{8.3e-04}$	$4.96e - 01_{4.9e-04}$
DTLZ2	2	$4.03e - 01_{4.9e-07}$	$4.02e - 01_{7.8e-05}$	$4.02e - 01_{9.9e-05}$
DTLZ3	2	$3.05e - 01_{7.3e-04}$	$4.28e - 02_{9.8e-02}$	$3.04e - 01_{1.5e-03}$
DTLZ4	2	$3.95e - 01_{7.3e-02}$	$4.08e - 01_{8.6e-05}$	$3.81e - 01_{1.0e-01}$
DTLZ5	2	$4.08e - 01_{1.5e-06}$	$4.08e - 01_{7.5e-05}$	$4.08e - 01_{6.8e-05}$
DTLZ6	2	$4.07e - 01_{5.4e-02}$	$1.59e - 02_{3.4e-02}$	$4.10e - 01_{6.1e-02}$
DTLZ7	2	$3.05e - 01_{3.5e-07}$	$3.05e - 01_{3.6e-05}$	$3.05e - 01_{4.8e-05}$
WFG1	2	$2.94e - 01_{1.4e-01}$	$2.53e - 01_{1.6e-01}$	$2.77e - 01_{1.5e-01}$
WFG2	2	$4.74e - 01_{3.1e-04}$	$4.75e - 01_{3.3e-04}$	$4.75e - 01_{3.5e-04}$
WFG3	2	$4.97e - 01_{2.4e-04}$	$4.96e - 01_{2.6e-04}$	$4.96e - 01_{3.6e-04}$
WFG4	2	$3.56e - 01_{1.8e-06}$	$3.55e - 01_{1.8e-04}$	$3.55e - 01_{1.6e-04}$
WFG5	2	$2.52e - 01_{2.0e-05}$	$2.53e - 01_{7.2e-03}$	$2.52e - 01_{6.7e-05}$
WFG6	2	$4.08e - 01_{6.0e-02}$	$3.98e - 01_{6.5e-02}$	$4.16e - 01_{4.1e-02}$
WFG7	2	$4.44e - 01_{8.9e-05}$	$4.43e - 01_{1.5e-04}$	$4.43e - 01_{1.6e-04}$
WFG8	2	$2.29e - 01_{1.1e-03}$	$2.28e - 01_{1.1e-03}$	$2.28e - 01_{1.0e-03}$
WFG9	2	$3.29e - 01_{1.0e-03}$	$3.28e - 01_{1.2e-03}$	$3.28e - 01_{1.0e-03}$
LZ09.F1	2	$5.14e - 01_{8.9e-05}$	$5.11e - 01_{2.0e-04}$	$5.11e - 01_{2.3e-04}$
LZ09.F2	2	$2.27e - 01_{8.0e-02}$	$1.98e - 01_{9.4e-02}$	$2.03e - 01_{8.6e-02}$
LZ09.F3	2	$4.06e - 01_{2.7e-02}$	$3.76e - 01_{4.5e-02}$	$4.01e - 01_{4.9e-02}$
LZ09.F4	2	$5.46e - 01_{4.4e-04}$	$5.43e - 01_{6.1e-04}$	$5.43e - 01_{7.9e-04}$
LZ09.F5	2	$5.62e - 01_{8.0e-03}$	$5.54e - 01_{1.8e-02}$	$5.57e - 01_{1.2e-02}$
LZ09.F7	2	$2.41e - 01_{9.3e-02}$	$2.75e - 01_{6.2e-02}$	$2.84e - 01_{1.8e-02}$
LZ09.F8	2	$2.93e - 01_{4.6e-02}$	$2.70e - 01_{5.8e-02}$	$3.08e - 01_{5.0e-02}$
LZ09.F9	2	$1.68e - 01_{6.2e-02}$	$1.22e - 01_{7.9e-02}$	$1.45e - 01_{6.3e-02}$
DTLZ1	3	$3.71e - 01_{4.6e-05}$	$0.00e + 00_{0.0e+00}$	$3.74e - 01_{2.2e-03}$
DTLZ2	3	$8.49e - 02_{4.8e-06}$	$7.66e - 02_{6.6e-04}$	$7.66e - 02_{8.2e-04}$
DTLZ3	3	$2.67e - 01_{4.4e-04}$	$0.00e + 00_{0.0e+00}$	$2.50e - 01_{4.1e-03}$
DTLZ4	3	$2.92e - 01_{2.1e-04}$	$2.86e - 01_{1.5e-03}$	$2.86e - 01_{1.4e-03}$
DTLZ5	3	$1.90e - 01_{3.8e-06}$	$1.98e - 01_{5.4e-05}$	$1.98e - 01_{5.2e-05}$
DTLZ6	3	$1.73e - 01_{6.1e-03}$	$4.78e - 03_{2.6e-02}$	$1.79e - 01_{5.6e-05}$
DTLZ7	3	$3.08e - 01_{3.4e-05}$	$3.07e - 01_{1.6e-03}$	$3.08e - 01_{1.6e-03}$
WFG1	3	$7.01e - 01_{3.5e-02}$	$7.06e - 01_{5.5e-03}$	$7.06e - 01_{5.5e-03}$
WFG2	3	$8.10e - 01_{1.1e-04}$	$8.00e - 01_{1.9e-03}$	$7.99e - 01_{2.0e-03}$
WFG3	3	$3.29e - 01_{2.2e-04}$	$3.30e - 01_{3.6e-04}$	$3.30e - 01_{6.1e-04}$
WFG4	3	$2.96e - 01_{3.5e-04}$	$2.89e - 01_{3.5e-03}$	$2.90e - 01_{2.7e-03}$
WFG5	3	$1.80e - 01_{6.0e-05}$	$1.75e - 01_{9.0e-04}$	$1.75e - 01_{9.3e-04}$
WFG6	3	$1.70e - 01_{3.3e-03}$	$1.61e - 01_{4.7e-03}$	$1.61e - 01_{6.0e-03}$
WFG7	3	$2.61e - 01_{5.2e-05}$	$2.51e - 01_{1.3e-03}$	$2.51e - 01_{1.4e-03}$
WFG8	3	$1.76e - 01_{4.7e-04}$	$1.56e - 01_{1.7e-03}$	$1.57e - 01_{1.7e-03}$
WFG9	3	$2.49e - 01_{1.1e-03}$	$2.26e - 01_{2.3e-03}$	$2.26e - 01_{2.8e-03}$
LZ09.F6	3	$7.36e - 02_{3.3e-02}$	$8.26e - 02_{2.2e-02}$	$6.12e - 02_{2.7e-02}$

Table 8: Mean and standard deviation of $HV_{\mathbf{q}}$, when \mathbf{q} is achievable.

P-MOGA, we believe that it has generated so many solutions outside the region of interest because, in the unachievable case, the preference relation used may consider as equivalent a nondominated solution outside the region of interest and a nondominated solution inside it.

5 Discussion

In this section, we will point out the advantages and disadvantages of the algorithm WASF-GA in comparison with r-NSGA-II, R-NSGA-II, g-NSGA-II and P-MOGA.

On the one hand, as said before, the distribution of the final populations generated by r-NSGA-II and R-NSGA-II highly depends on the parameters δ_0 and ϵ , respectively, which must be set for each problem and for each reference point, independently. There are no clear indications in [42] and [17] about how to set δ_0 and ϵ and, in our computational tests,

Problem	Obj.	WASF-GA	g-NSGA-II	P-MOGA
ZDT1	2	$5.22e - 01_{6.7e-07}$	$5.21e - 01_{8.8e-05}$	$5.21e - 01_{8.2e-05}$
ZDT2	2	$4.56e - 01_{3.8e-07}$	$4.55e - 01_{7.7e-05}$	$4.55e - 01_{7.0e-05}$
ZDT3	2	$5.32e - 01_{6.7e-07}$	$5.32e - 01_{2.1e-05}$	$5.32e - 01_{3.1e-05}$
ZDT4	2	$5.72e - 01_{2.0e-04}$	$5.71e - 01_{3.2e-04}$	$5.71e - 01_{3.0e-04}$
ZDT6	2	$4.32e - 01_{1.2e-04}$	$4.31e - 01_{2.8e-04}$	$4.31e - 01_{3.5e-04}$
DTLZ1	2	$4.97e - 01_{2.3e-04}$	$4.96e - 01_{3.4e-04}$	$4.97e - 01_{2.0e-04}$
DTLZ2	2	$3.13e - 01_{1.7e-06}$	$3.12e - 01_{9.4e-05}$	$3.12e - 01_{8.6e-05}$
DTLZ3	2	$3.21e - 01_{1.3e-03}$	$2.79e - 02_{7.3e-02}$	$3.20e - 01_{1.8e-03}$
DTLZ4	2	$3.68e - 01_{9.8e-02}$	$3.93e - 01_{9.8e-05}$	$3.94e - 01_{1.0e-04}$
DTLZ5	2	$2.69e - 01_{1.4e-07}$	$2.69e - 01_{9.6e-05}$	$2.69e - 01_{1.0e-04}$
DTLZ6	2	$3.11e - 01_{1.1e-02}$	$2.29e - 01_{5.7e-02}$	$3.03e - 01_{2.0e-02}$
DTLZ7	2	$5.90e - 01_{1.3e-04}$	$5.95e - 01_{8.4e-05}$	$5.95e - 01_{9.4e-05}$
WFG1	2	$5.21e - 01_{9.2e-06}$	$5.13e - 01_{4.0e-02}$	$4.95e - 01_{7.4e-02}$
WFG2	2	$5.69e - 01_{7.1e-04}$	$5.69e - 01_{8.7e-04}$	$5.69e - 01_{9.6e-04}$
WFG3	2	$4.97e - 01_{7.1e-04}$	$4.96e - 01_{3.7e-04}$	$4.96e - 01_{5.4e-04}$
WFG4	2	$3.67e - 01_{5.9e-07}$	$3.67e - 01_{1.8e-04}$	$3.67e - 01_{1.4e-04}$
WFG5	2	$2.91e - 01_{5.0e-06}$	$2.91e - 01_{1.1e-04}$	$2.92e - 01_{3.6e-03}$
WFG6	2	$3.02e - 01_{1.5e-02}$	$3.01e - 01_{1.8e-02}$	$3.02e - 01_{9.0e-03}$
WFG7	2	$3.47e - 01_{1.4e-05}$	$3.47e - 01_{1.3e-04}$	$3.47e - 01_{1.3e-04}$
WFG8	2	$3.85e - 01_{2.0e-03}$	$3.86e - 01_{2.1e-03}$	$3.87e - 01_{1.9e-03}$
WFG9	2	$3.95e - 01_{1.6e-03}$	$3.94e - 01_{2.1e-03}$	$3.94e - 01_{1.5e-03}$
LZ09_F1	2	$5.83e - 01_{8.5e-05}$	$5.80e - 01_{2.6e-04}$	$5.80e - 01_{1.6e-04}$
LZ09_F2	2	$2.15e - 01_{8.3e-02}$	$2.14e - 01_{8.7e-02}$	$2.11e - 01_{9.2e-02}$
LZ09_F3	2	$2.16e - 01_{1.2e-01}$	$1.89e - 01_{1.1e-01}$	$2.03e - 01_{1.4e-01}$
LZ09_F4	2	$5.16e - 01_{4.2e-03}$	$5.17e - 01_{2.2e-03}$	$5.20e - 01_{2.3e-03}$
LZ09_F5	2	$5.63e - 01_{6.6e-03}$	$5.59e - 01_{9.9e-03}$	$5.62e - 01_{8.0e-03}$
LZ09_F7	2	$2.41e - 01_{8.7e-02}$	$1.87e - 01_{1.2e-01}$	$1.76e - 01_{1.4e-01}$
LZ09_F8	2	$8.92e - 02_{9.7e-02}$	$7.93e - 02_{8.7e-02}$	$6.69e - 02_{8.3e-02}$
LZ09_F9	2	$1.42e - 01_{8.1e-02}$	$1.40e - 01_{7.4e-02}$	$1.28e - 01_{8.5e-02}$
DTLZ1	3	$7.88e - 01_{3.8e-04}$	$1.83e - 02_{9.9e-02}$	$7.85e - 01_{2.9e-03}$
DTLZ2	3	$6.86e - 01_{4.1e-04}$	$6.60e - 01_{4.9e-03}$	$6.58e - 01_{3.1e-03}$
DTLZ3	3	$5.84e - 01_{9.5e-04}$	$0.00e + 00_{0.0e+00}$	$5.53e - 01_{3.5e-03}$
DTLZ4	3	$4.97e - 01_{3.2e-03}$	$5.09e - 01_{7.4e-03}$	$5.14e - 01_{2.7e-03}$
DTLZ5	3	$2.57e - 01_{2.5e-06}$	$2.57e - 01_{1.4e-03}$	$2.58e - 01_{1.7e-04}$
DTLZ6	3	$2.72e - 01_{1.4e-02}$	$9.91e - 02_{1.0e-01}$	$2.75e - 01_{1.4e-04}$
DTLZ7	3	$4.96e - 01_{2.9e-04}$	$5.03e - 01_{1.5e-03}$	$4.97e - 01_{1.9e-03}$
WFG1	3	$8.16e - 01_{2.9e-03}$	$7.96e - 01_{8.7e-03}$	$8.05e - 01_{5.5e-03}$
WFG2	3	$8.59e - 01_{7.9e-04}$	$8.49e - 01_{1.1e-02}$	$8.50e - 01_{4.5e-03}$
WFG3	3	$3.30e - 01_{2.6e-04}$	$3.26e - 01_{2.2e-03}$	$3.28e - 01_{5.0e-04}$
WFG4	3	$5.12e - 01_{2.2e-03}$	$5.17e - 01_{3.4e-03}$	$5.24e - 01_{2.5e-03}$
WFG5	3	$5.91e - 01_{6.3e-04}$	$5.61e - 01_{9.5e-03}$	$5.62e - 01_{3.4e-03}$
WFG6	3	$5.33e - 01_{2.5e-03}$	$5.10e - 01_{4.3e-03}$	$5.09e - 01_{4.1e-03}$
WFG7	3	$5.12e - 01_{2.5e-03}$	$4.77e - 01_{2.8e-03}$	$4.77e - 01_{3.7e-03}$
WFG8	3	$4.57e - 01_{4.4e-02}$	$3.81e - 01_{2.3e-02}$	$4.02e - 01_{4.5e-02}$
WFG9	3	$6.22e - 01_{3.2e-03}$	$5.87e - 01_{5.1e-03}$	$5.91e - 01_{3.3e-03}$
LZ09_F6	3	$4.74e - 01_{2.0e-01}$	$4.15e - 01_{7.4e-02}$	$2.61e - 01_{2.0e-02}$

Table 9: Mean and standard deviation of $HV_{\mathbf{q}}$, when \mathbf{q} is unachievable.

we found satisfactory values for them by trial and error, following our intuition. Based on our experience, it may not be easy to find appropriate values of them in order to generate objective vectors as much interesting as possible according to the reference point used. Although it is true that r-NSGA-II and R-NSGA-II are not designed for approximating just our region of interest, at least, when the reference point is achievable, there is no doubt that the most interesting Pareto optimal objective vectors which dominate the reference point (that is, which are inside our region of interest). Consequently, the fact that WASF-GA assures the generation of nondominated objective vectors approximating just our region of interest is an advantage against r-NSGA-II and R-NSGA-II, at least in the achievable case.

In WASF-GA, the only extra parameter that must be set is the number of weight vectors, N_{μ} . However, this parameter also represents the number of the objective vectors in the final population shown to the DM, and it can easily set by him/her. Indeed, if the

ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 (2 objectives)										
	g-NSGA-II					P-MOGA				
WASF-GA	▲	▲	▽	▲	▲	▲	▲	▽	▲	▲
g-NSGA-II						-	-	-	-	-

DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7 (2 objectives)														
	g-NSGA-II							P-MOGA						
WASF-GA	▲	▲	▲	▲	▲	▲	▽	▲	▲	▲	▲	▲	▽	
g-NSGA-II								▽	-	▽	-	-	▽	-

WFG1, WFG2, WFG3, WFG4, WFG5, WFG6, WFG7, WFG8, WFG9 (2 objectives)																		
	g-NSGA-II								P-MOGA									
WASF-GA	-	▽	▲	▲	▲	-	▲	-	▲	-	▽	▲	▲	▲	-	▲	▲	▲
g-NSGA-II										-	-	-	-	-	-	-	-	-

LZ09_F1, LZ09_F2, LZ09_F3, LZ09_F4, LZ09_F5, LZ09_F7, LZ09_F8, LZ09_F9 (2 objectives)																	
	g-NSGA-II					P-MOGA											
WASF-GA	▲	-	▲	▲	-	-	-	▲	▲	-	-	▲	-	-	-	-	-
g-NSGA-II									-	-	▽	-	-	-	▽	-	-

DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7 (3 objectives)														
	g-NSGA-II							P-MOGA						
WASF-GA	▲	▲	▲	▲	▽	▲	▲	▽	▲	▲	▲	▽	▽	-
g-NSGA-II								▽	-	▽	-	-	▽	▽

WFG1, WFG2, WFG3, WFG4, WFG5, WFG6, WFG7, WFG8, WFG9 (3 objectives)																		
	g-NSGA-II								P-MOGA									
WASF-GA	-	▲	▽	▲	▲	▲	▲	▲	▲	-	▲	▽	▲	▲	▲	▲	▲	▲
g-NSGA-II										-	-	-	-	-	-	-	▽	-

LZ09_F6 (3 objectives)										
	g-NSGA-II					P-MOGA				
WASF-GA										
g-NSGA-II										▲

Table 10: Wilcoxon test in the two and three-objective problems, when \mathbf{q} is achievable.

DM is unable to give a value, we can use $N_\mu = N$ by default. We would like to remark that the parameters ε and S needed for the generation of the weights do not depend on the problem itself, but on the number of objective functions, and we have proposed appropriate values for them in Subsection 3.2.

On the other hand, the algorithms R-NSGA-II and r-NSGA-II share another weakness: they have been defined to emphasize solutions whose objective vectors are close to the reference point(s) with respect to the Euclidean distance. When the reference point is achievable, the Euclidean distance may produce undesirable effects, since objective vectors are close to the reference point may not be near to the Pareto optimal front. Then, for achievable reference points, the use of the Euclidean distance may emphasize inadequate objective vectors, what may complicate the convergence of the algorithm. In this regards, WASF-GA has been formulated considering the ASF given in (2), which behaves properly for both achievable and unachievable reference points, avoiding undesired effects in the achievable case.

As mentioned before, the g-dominance may prefer a dominated objective vector over

ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 (2 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲ ▲ ▽ ▲ ▲	▲ ▲ ▽ ▲ ▲
g-NSGA-II		- - - - -

DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7 (2 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲ ▲ ▲ ▲ ▲ ▲ ▽	▲ ▲ ▲ ▲ ▲ ▲ ▽
g-NSGA-II		- ▽ ▽ ▽ ▽ ▽ ▽

WFG1, WFG2, WFG3, WFG4, WFG5, WFG6, WFG7, WFG8, WFG9 (2 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲ ▽ ▲ ▲ ▲ - ▲ - -	▲ ▽ ▲ ▲ ▲ - - ▽ ▲
g-NSGA-II		▲ - - ▽ - - ▽ ▽ -

LZ09_F1, LZ09_F2, LZ09_F3, LZ09_F4, LZ09_F5, LZ09_F7, LZ09_F8, LZ09_F9 (2 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲ - - - - - - -	▲ - - ▽ - ▲ - -
g-NSGA-II		▽ - - ▽ - - - - -

DTLZ1, DTLZ2, DTLZ3, DTLZ4, DTLZ5, DTLZ6, DTLZ7 (3 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲ ▲ ▲ ▽ ▽ ▲ ▽	▲ ▲ ▲ ▽ ▽ ▽ ▽
g-NSGA-II		▽ ▲ ▽ ▽ ▽ ▽ ▲

WFG1, WFG2, WFG3, WFG4, WFG5, WFG6, WFG7, WFG8, WFG9 (3 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲ ▲ ▲ ▽ ▲ ▲ ▲ ▲ ▲	▲ ▲ ▲ ▽ ▲ ▲ ▲ ▲ ▲
g-NSGA-II		▽ - ▽ ▽ - - - ▽ ▽

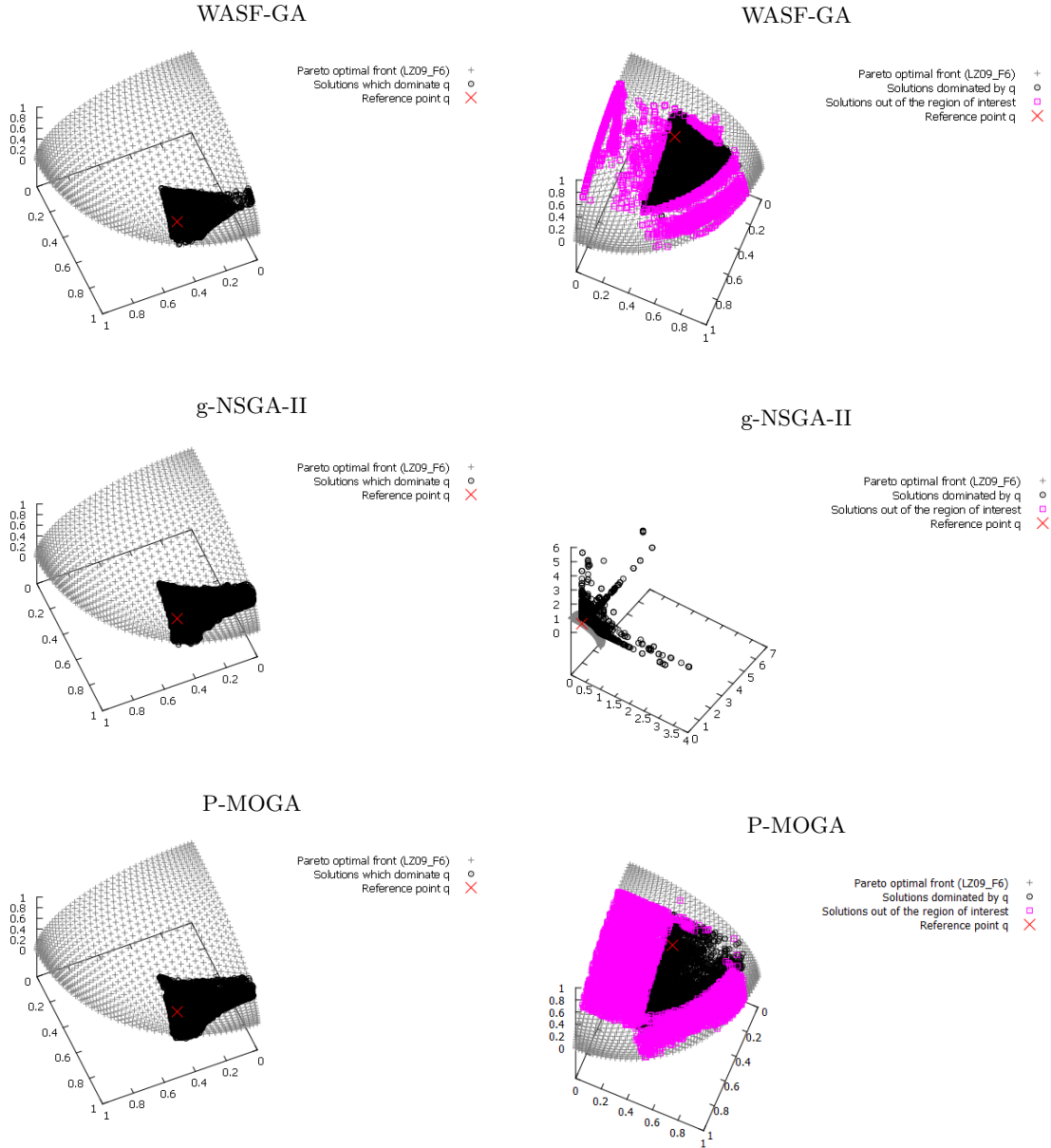
LZ09_F6 (3 objectives)		
	g-NSGA-II	P-MOGA
WASF-GA	▲	▲
g-NSGA-II		▲

Table 11: Wilcoxon test in the two and three-objective problems, when \mathbf{q} is unachievable.

Algorithm	Basic operations	Complexities	Global Complexity
WASF-GA	ASF assignment Classification into several fronts	$O(k \cdot 2N \cdot N_\mu)$ $O(2N \cdot N_\mu)$	$O(k \cdot N \cdot N_\mu)$
g-NSGA-II	Nondominated sorting Crowding distance assignment Crowding distance sorting Compute $Flag_g$	$O(k \cdot (2N)^2)$ $O(k \cdot 2N \cdot \log(2N))$ $O(2N \cdot \log(2N))$ $O(2k \cdot 2N)$	$O(k \cdot N^2)$
P-MOGA	Preference ranking procedure	$O(3k \cdot (2N)^2)$	$O(k \cdot N^2)$

Table 12: Computational complexities.

the objective vector which dominates it, and this is due to the fact that solutions whose objective vectors satisfy all aspiration levels and solutions whose objective vectors fulfill none of the aspiration levels are emphasized over solutions whose objective vectors satisfy some of the aspiration levels. This weakness has been overcome by WASF-GA. According to the classification of the individuals in WASF-GA, an objective vector dominated by



Achievable reference point (0.49, 0.94, 0.57). Unachievable reference point (0.21, 0.42, 0.46).

Figure 4: Solutions obtained for the LZ09_F6 problem.

another one can never be in a lower level front than the one which dominates it.

Regarding P-MOGA, we would like to say that the preference domination defined in [22] has a small inconvenience for unachievable reference points. Specifically, when the reference point is unachievable, an objective vector outside the region of interest can be incomparable to another objective vector inside that region. However, in this situation, the objective vector inside should be preferred to the solution outside. Even though this fact, P-MOGA has been ranked as the second best method after WASF-GA in our computational tests, what shows its potential.

When the objective functions are in very different scales, we need to have approximations of the ideal and the nadir vectors for the normalization carried out in the ASF given

in (2). This fact may be a disadvantage of WASF-GA, given that other algorithms do not need to normalize and, thus, do not require estimations of the ideal and the nadir vectors, as g-NSGA-II and P-MOGA. In our computational tests, we set these boundary vectors using the best and the worst objective values reached in the Pareto optimal fronts of the problems considered. Any other available technique for estimating the ideal and the nadir vectors can be used for problems with unknown Pareto optimal fronts.

Regarding the procedure described in Subsection 3.2 for the generation of the vector of weights, we would like to remark why we have considered in WASF-GA a set of weight vectors whose inverse components form vectors well-distributed in $(0, 1)^k$. Minimizing the ASF given in (2) over the feasible set means to project the reference point onto the Pareto optimal front in the direction defined by the inverse of the weights used. Then, by using weight vectors which verify that the vectors formed by their inverse components are well-distributed in $(0, 1)^k$, the ASF used in WASF-GA will project the reference point taking into account an evenly distributed set of projection directions. Otherwise, if we had used the evenly distributed weight vectors in $(0, 1)^k$ instead, this ASF would have projected \mathbf{q} in the directions given by the inverses of these weight vectors, but we cannot assure that these projection directions are evenly distributed. Furthermore, it should be said that, in WASF-GA, the weight vectors whose inverses are well-distributed in $(0, 1)^k$ generated better results than the ones obtained when the evenly distributed weight vectors in $(0, 1)^k$ were used. However, it must be noted that this fact does not guarantee to generate evenly distributed objective vectors in the region of interest, although it may help to cover the whole region of interest.

6 Conclusion

The main purpose of preference-based EMO algorithms consists of approximating the region of the Pareto optimal front which contains the most promising Pareto optimal objective vectors and their corresponding Pareto optimal solutions for the DM. In this paper, we have proposed a preference-based EMO algorithm called *Weighting Achievement Scalarizing Function Genetic Algorithm* (WASF-GA), which considers a reference point given by a DM as preferential information. This algorithm tries to approximate the region of interest of the Pareto optimal front defined by the reference point. For an achievable reference point, this region of interest contains all the Pareto optimal objective vectors which dominate the reference point and, thus, which are the most interesting objective vectors for the DM. For an unachievable reference point, the region of interest is formed by the Pareto optimal objective vectors which dominate the reference point. In this case, the objective vectors lying in this region are likely to be more appealing for the DM than the ones outside it, given that they worsen the reference values as little as possible.

To approximate the region of interest, at each generation of WASF-GA, the population of individuals is divided into several fronts, taking into account the values that each solution reaches on an ASF for each vector of weights in a sample of the weight vector space. The key of the success of WASF-GA is to consider a sample of weight vectors which verify that the vectors formed by their inverse components are evenly distributed in the weight vector space. Taking into account the obtained results, this fact has permitted us to generate good approximations of the region of interests defined by several reference points in the problems considered.

An important feature of WASF-GA is that the DM does not need to set any extra parameter by his/her own without any understandable criterion. (S)he just has to give the reference point, and the number of objective vectors (solutions) (s)he wants to have

in the final population, which represent the number of weight vectors used in WASF-GA. However, if so desired, the number of weight vectors may be set by default as the size of the population of WASF-GA.

The usefulness of WASF-GA has been shown in several test problems. We have compared WASF-GA with four preference-based EMO algorithms which also use the reference point preferential scheme: r-NSGA-II, R-NSGA-II, g-NSGA-II and P-MOGA. The obtained results have shown that WASF-GA has been able to approximate the region of interest of several types of Pareto optimal fronts, for achievable and unachievable reference points. An analysis of the computational complexities and the qualities of the final populations generated by all the algorithms states that WASF-GA is the least computationally complex, and gets the best values of the hypervolume metric used in nearly all the cases.

One possible line of future research would be to propose an interactive method on which WASF-GA were embedded at each iteration. In this case, the number of weight vectors could be lower in order to decrease the computation time.

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 group SEJ-532).

References

- [1] J. Branke. Consideration of Partial User Preferences in Evolutionary Multiobjective Optimization. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization, Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*, pages 157–178. Springer, 2008.
- [2] J. Branke and K. Deb. Integrating user preferences into evolutionary multi-objective optimization. In Y. Jin, editor, *Knowledge Incorporation in Evolutionary Computation*, volume 167, pages 461–477. Springer-Verlag, Berlin, 2004.
- [3] J. Branke, T. Kaussler, and H. Schemeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32:499–507, 2001.
- [4] V. Chankong and Y.Y. Haimes. *Multiobjective Decision Making Theory and Methodology*. Elsevier Science Publishing, New York, 1983.
- [5] C. A. C. Coello, G. B. Lamont, and D.A.V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems. Second Edition*. Springer, New York, 2007.
- [6] C.A.C. Coello. Handling Preferences in Evolutionary Multiobjective Optimization: A Survey. In *IEEE Congress on Evolutionary Computation*, pages 30–37, 2000.
- [7] D.W. Corne and J.D. Knowles. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Annual conference on Genetic and evolutionary computation, GECCO '07*, pages 773–780, New York, 2007. ACM Press.
- [8] K. Deb. *Multi-objective Optimization using Evolutionary Algorithms*. Wiley, Chichester, 2001.

- [9] K. Deb. Salient issues of multi-objective evolutionary algorithms. In K. Deb, editor, *Multi-objective Optimization using Evolutionary Algorithms*, volume 8, pages 315–445, Chichester, 2001. Wiley.
- [10] K. Deb and A. Kumar. Interactive Evolutionary Multi-Objective Optimization and Decision-Making using Reference Direction Method. In *9th Annual Conference on Genetic and Evolutionary Computation (GECCO-2007)*, pages 781–788, 2007.
- [11] K. Deb and A. Kumar. Light beam search based multi-objective optimization using evolutionary algorithms. In *IEEE Congress on Evolutionary Computation (CEC-2007)*, pages 2125–2132, 2007.
- [12] K. Deb and K. Miettinen. Nadir point estimation using evolutionary approaches: Better accuracy and computational speed through focused search. In M. Ehrgott, B. Naujoks, T.J. Stewart, and J. Wallenius, editors, *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems*, pages 339–354, Berlin, Heidelberg, 2010. Springer-Verlag.
- [13] K. Deb., K. Miettinen, and S. Chaudhuri. 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.
- [14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002.
- [15] K. Deb and D. Saxena. Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *World Congress on Computational Intelligence*, pages 3352–3360, 2006.
- [16] K. Deb, A. Sinha, P.J. Korhonen, and J. Wallenius. An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. *IEEE Transactions on Evolutionary Computation*, 14(5):723–739, 2010.
- [17] K. Deb, J. Sundar, B. Ubay, and S. Chaudhuri. Reference point based multi-objective optimization using evolutionary algorithm. *International Journal of Computational Intelligence Research*, 2(6):273–286, 2006.
- [18] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable multi-objective optimization test problems. In *Congress on Evolutionary Computation (CEC-2002)*, pages 825–830, 2002.
- [19] Juan J. Durillo and Antonio J. Nebro. jmetal: A java framework for multi-objective optimization. *Advances in Engineering Software*, 42:760–771, 2011.
- [20] M. Ehrgott and D. Tenfelde-Podehl. Computation of ideal and nadir values and implications for their use in MCDM methods. *European Journal of Operational Research*, 151:119–139, 2003.
- [21] J. R. Figueira, A. Liefoghe, E. G. Talbi, and A. P. Wierzbicki. A parallel multiple reference point approach for multi-objective optimization. *European Journal of Operational Research*, 205:390–400, 2010.

- [22] C.M. Fonseca and P.J. Fleming. Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In *5th International Conference on Genetic Algorithms*, pages 416–423, 1993.
- [23] C.M. Fonseca, L. Paquete, and M. Lopez-Ibanez. An improved dimension-sweep algorithm for the hypervolume indicator. In *IEEE Congress on Evolutionary Computation (CEC-2006)*, pages 1157 – 1163, 2006.
- [24] M. Gong, F. Liu, W. Zhang, L. Jiao, and Q. Zhang. Interactive MOEA/D for multi-objective decision making. In *13th annual conference on Genetic and evolutionary computation, GECCO '11*, pages 721–728, 2011.
- [25] S. Huband, P. Hingston, L. Barone, and L. While. A review of multiobjective test problems and a scalable test problem toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, 2006.
- [26] C. L. Hwang and A. S. M. Masud. *Multiple Objective Decision Making – Methods and Applications: A State-of-the-Art Survey*. Springer-Verlag, Berlin, 1979.
- [27] A. Jaszkievicz and J. Branke. Interactive Multiobjective Evolutionary Algorithms. In J. Branke, K. Deb, K. Miettinen, and R. Slowinski, editors, *Multiobjective Optimization, Interactive and Evolutionary Approaches*, volume 5252 of *Lecture Notes in Computer Science*, pages 179–193. Springer, 2008.
- [28] A. Jaszkievicz and R. Slowiński. The 'Light Beam Search' Approach – An Overview of Methodology and Applications. *European Journal of Operational Research*, 113(2):300–314, 1999.
- [29] J. D. Knowles and D. W. Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary Computation*, 8(2):149–172, 2000.
- [30] P. Korhonen and J. Laakso. A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 24(2):277–287, 1986.
- [31] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, moea/d and nsga-ii. *IEEE Transactions on Evolutionary Computation*, 13(2):284–302, 2009.
- [32] M. Luque, K. Miettinen, P. Eskelinen, and F. Ruiz. Incorporating preference information in interactive reference point methods for multiobjective optimization. *Omega*, 37(2):450–462, 2009.
- [33] M. Luque, F. Ruiz, and K. Miettinen. Global formulation for interactive multiobjective optimization. *OR Spectrum*, 33(1):27–48, 2011.
- [34] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, Berkeley, University of California Pressley, 1967.
- [35] K. Miettinen. *Nonlinear Multiobjective Optimization*. Kluwer Academic Publishers, Boston, 1999.
- [36] K. Miettinen and M.M. Mäkelä. Comparative evaluation of some interactive reference point-based methods for multi-objective optimisation. *Journal of the Operational Research Society*, 50(9):949–959, 1999.

- [37] K. Miettinen and M.M. Mäkelä. On scalarizing functions in multiobjective optimization. *OR Spectrum*, 24(2):193–213, 2002.
- [38] K. Miettinen, M.M. Mäkelä, and K. Kaario. Experiments with classification-based scalarizing functions in interactive multiobjective optimization. *European Journal of Operational Research*, 175(2):931–947, 2006.
- [39] J. Molina, L. V. Santana, A. G. Hernandez-Diaz, C. A. Coello, and R. Caballero. g-dominance: Reference point based dominance for multiobjective metaheuristics. *European Journal of Operational Research*, 197:685–692, 2009.
- [40] L. Rachmawati and D. Srinivasan. Preference Incorporation in Multiobjective Evolutionary Algorithms. In *IEEE Congress on Evolutionary Computation*, pages 3385–3391, 2006.
- [41] F. Ruiz, M. Luque, and J.M. Cabello. A classification of the weighting schemes in reference point procedures for multiobjective programming. *Journal of the Operational Research Society*, 60(4):544–553, 2009.
- [42] L. Ben Said, S. Bechikh, and K. Ghedira. The r-dominance: A new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Transactions on Evolutionary Computation*, 14(5):801–818, 2010.
- [43] Y. Sawaragi, H. Nakayama, and T. Tanino. *Theory of Multiobjective Optimization*. Academic Press, Orlando, 1985.
- [44] K. Sindhya, K. Miettinen, and K. Deb. A hybrid framework for evolutionary multi-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(4):495–511, 2013.
- [45] K. Sindhya, A.B. Ruiz, and K. Miettinen. A Preference Based Interactive Evolutionary Algorithm for Multi-objective Optimization: PIE. In R. Takahashi, K. Deb, E. Wanner, and S. Greco, editors, *Evolutionary Multi-Criterion Optimization*, volume 6576 of *Lecture Notes in Computer Science*, pages 212–225, 2011.
- [46] A. Sinha, P.J. Korhonen, J. Wallenius, and K. Deb. An Interactive Evolutionary Multi-objective Optimization Method Based on Polyhedral Cones. In C. Blum and R. Battiti, editors, *Learning and Intelligence Optimization*, volume 6073 of *Lecture Notes in Computer Science*, pages 318–332. Springer, 2010.
- [47] L. Thiele, K. Miettinen, P. Korhonen, and J. Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3):411–436, 2009.
- [48] A. P. Wierzbicki. The use of reference objectives in multiobjective optimization. In G. Fandel and T. Gal, editors, *Multiple Criteria Decision Making, Theory and Applications*, pages 468–486. Springer-Verlag, Berlin, 1980.
- [49] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):pp. 80–83, 1945.
- [50] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007.

- [51] E. Zitzler, K. Deb, and L. Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, 8(2):173–195, 2000.
- [52] E. Zitzler and S. Kuenzli. Indicator-based selection in multiobjective search. In X. Yao, E. Burke, J. A. Lozano, J. Smith, J. J. Merelo-Guervos, J. A. Bullinaria, J. Rowe, P. Tino, A. Kaban, and H. P. Schwefel, editors, *8th International Conference on Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842. Springer-Verlag, Berlin, 2004.
- [53] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In K. C. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.
- [54] E. Zitzler and L. Thiele. Multiobjective optimization using evolutionary algorithms - A comparative case study. In A. Eiben, T. Bäck, M. Schoenauer, and H.P. Schwefel, editors, *Parallel Problem Solving from Nature*, volume 1498 of *Lecture Notes in Computer Science*, pages 292–301. Springer-Verlag, Berlin, 1998.
- [55] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.
- [56] X. Zou, Y. Chen, M. Liu, and L. Kang. A New Evolutionary Algorithm for Solving Many-Objective Optimization Problems. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(5):1402–1412, 2008.