An Improvement Study of the Decomposition-based Algorithm Global WASF-GA for Evolutionary Multiobjective Optimization

Sandra Gonzalez-Gallardo, Rubén Saborido, Ana B. Ruiz, and Mariano Luque

Department of Applied Economics (Mathematics), University of Málaga,
C/ Ejido 6, 29071, Málaga, Spain
sandragg@uma.es,rsain@uma.es,abruiz@uma.es,mluque@uma.es

Abstract. The convergence and the diversity of the decomposition-based evolutionary algorithm Global WASF-GA (GWASF-GA) relies on a set of weight vectors that determine the search directions in the objective space for new non-dominated solutions. Although using weight vectors whose search directions are widely distributed may lead to a well-diversified approximation of the Pareto front (PF), this fact may not perform as expected for complicated PFs (discontinuous, not convex, etc.). To handle this, we propose an adjustment of the weight vectors once GWASF-GA has been run for certain number of generations. This dynamic adjustment is aimed at re-calculating some of the weight vectors, so that search directions pointing to overcrowded regions of the PF are redirected toward parts with a lack of solutions that may be hard to be approximated. We test different parameters settings of this dynamic adjustment in optimization problems with three, five, and six objectives. We conclude that GWASF-GA performs better when adjusting the weight vectors dynamically than without applying the adjustment.

Keywords: Evolutionary multiobjective optimization · Decomposition-based algorithm · Global WASF-GA · Weight vector.

we say that $\mathbf{y}$ *dominates* $\bar{\mathbf{y}}$ if $y_i \leq \bar{y}_i$ for all $i = 1, \ldots, m$, with, at least, one strict inequality. Then, a solution $\mathbf{x} \in S$ is *Pareto optimal* if there does not exist another $\bar{\mathbf{x}} \in S$ so that $\mathbf{f}(\bar{\mathbf{x}})$ dominates $\mathbf{f}(\mathbf{x})$. All Pareto optimal solutions form the *Pareto optimal set* (PS) in the decision space and the *Pareto optimal front* (PF) in the objective space.

Evolutionary multiobjective optimization (EMO) has been proven to be a good methodology to solve multiobjective optimization problems [1]. EMO algorithms start from an initial population of solutions (or individuals) and, based on an iterative evolutionary process, produce an approximation of the PF. Among all of them, decomposition-based EMO algorithms scalarize the original multiobjective optimization problem into a set of single objective optimization problems. GWASF-GA [6] is one of these algorithms, in which an achievement scalarizing function (ASF) [7] based on the Tchebychev distance is used as fitness function to classify the individuals into different fronts. For this classification, two reference points are simultaneously considered in the ASF: the nadir and the utopian points. The *nadir point*, denoted as $\mathbf{z}^{\text{nad}} = (z_1^{\text{nad}}, \ldots, z_k^{\text{nad}})^T$, is defined by the worst possible objective function values, that is, $z_i^{\text{nad}} = \max_{\mathbf{x} \in E} f_i(\mathbf{x})$ $(i = 1, \ldots, k)$. To calculate the *utopian point*, referred to as $\mathbf{z}^{\star\star} = (z_1^{\star\star}, \ldots, z_k^{\star\star})^T$, we need the *ideal point* $\mathbf{z}^{\star} = (z_1^{\star}, \ldots, z_k^{\star})^T$, defined as $z_i^{\star} = \min_{\mathbf{x} \in E} f_i(\mathbf{x}) = \min_{\mathbf{x} \in S} f_i(\mathbf{x})$ $(i = 1, \ldots, k)$. Thus, the components of the utopian point are obtained as $z_i^{\star\star} = z_i^{\star} - \epsilon_i$ $(i = 1, \ldots, k)$, where $\epsilon_i > 0$ is a relatively small real value. At each generation of GWASF-GA, individuals are classified according to their ASF values calculated using both the nadir and the utopian points, but also depending on a set of weight vectors. Due to the mathematical properties of the ASF [7], the weight vectors determine the search directions for new nondominated solutions towards the PF. Because of this, the weight vectors should preferably define uniformly distributed search directions in the objective space.

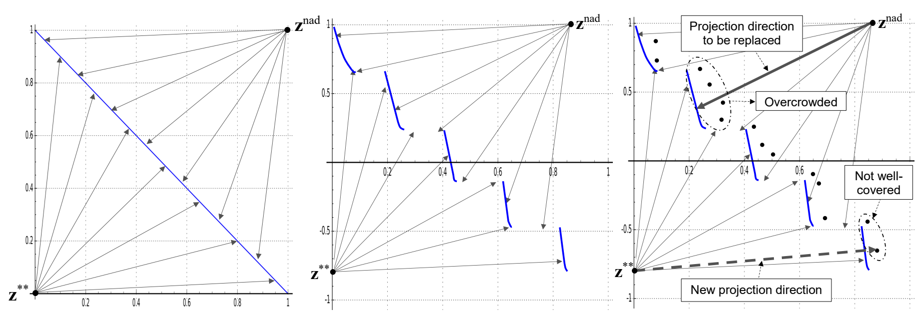
Promising results have been provided by GWASF-GA [6]. However, in cases where the PF is discontinuous or has a complex shape, GWASF-GA may not be able to generate a good approximation of the PF, for instance, if different weight vectors are generating similar solutions. This means that different weight vectors may be directing the search towards the same region of the PF. To amend this, we propose to improve the performance of GWASF-GA by dynamically adjusting the weight vectors. The idea is to re-direct some of the search directions, taken into account the current distribution of the solutions obtained, so that weight vectors generating solutions in overcrowded areas of the PF are replaced by new ones directing the search towards regions of the PF with lack of solutions. Since this improvement depends on different parameters, we carry out an experimental study to gain knowledge about their impact on the performance of GWASF-GA when solving different multiobjective optimization test problems.

Next, Section 2 motivates the weigh vectors' adjustment to improve the performance of GWASF-GA. In Section 3, we describe our proposal in details. Section 4 presents and discusses the experiments designed to study the benefits of adjusting the weight vectors and, finally, Section 5 concludes this work.

2 Motivation

Minimizing the ASF proposed in [7] over the feasible set means, in practice, to project the reference point used onto the PF in the projection direction given by the inverse components of the weight vector. Owing to this, the weight vectors are key parameters in GWASF-GA since they set the search directions for new non-dominated solutions, either from the nadir or from the utopian point. Thus, they affect the diversity and the convergence of GWASF-GA and, according to [6], the weight vectors used must define projection directions as evenly distributed as possible. Besides, since the front classification is based on the ASF values, each individual in the population produced at each generation is associated either with the nadir or with the utopian point, and with a weight vector.

In practice, a set of weight vectors producing uniformly spread projection directions of the nadir or of the utopian point allows a well-diversified population to be obtained just in case the PF is not complex, such as e.g. an hyper-plane (Figure 1 (a)). In case the PF is discontinuous (Figure 1 (b)), the same weight vectors may be generating very similar (i.e. very close) non-dominated solutions, because their projection directions may be pointing towards the same area of the PF, or may be even directing the search for new solutions towards gaps in the PF. This fact does not contribute to the convergence of the algorithm and produces a lack of the diversity among the solutions generated.



(a) Hyper-plane PF (b) Discontinuous PF (c) Our proposal

Fig. 1. Projection directions from $\mathbf{z}^{\mathrm{nad}}$ and $\mathbf{z}^{\star\star}$ in GWASF-GA.

According to this, to improve the quality (convergence and diversity) of the approximation generated by GWASF-GA, we suggest to re-calculate some of the weight vectors once GWASF-GA has converged to some extent. The new weight vectors generated should be able to fit to the features of the PF, such as e.g. discontinuities, convexity, etc. To achieve this, one possibility is to re-distribute the weight vectors according to the solutions in the current population. These individuals have survived along the generations and, therefore, are assumed to be the best individuals found so far to approximate the PF.

After performing a number of generations, we suggest to re-orientate the search for new individuals in GWASF-GA as shown in Figure 1 (c). According to the distribution of the solutions in the current population, our proposal is to dynamically replace the weight vectors whose projection directions point towards overcrowded parts (where there are more solutions), by weight vectors helping to approximate regions that are not so well-covered. With this, we try to diversify the solutions generated, helping the algorithm to converge towards regions where the density of solutions is not very high at the moment and that may be hard to be approximated. The convexity of the PF is also considered somehow when adjusting the weight vectors, since more importance is likely to be given to the use of the utopian point if the PF is concave, or of the nadir point if it is convex.

3 Improvement of GWASF-GA through a Dynamic Adjustment of the Weight Vectors

Next, we describe how to enhance the convergence and diversity of the final population generated by GWASF-GA. Let G_T denote the total number of generations to be performed. Initially, a representative set W of weight vectors in $[0, 1]^k$ is considered. They are generated as suggested in [6] (pp. 321-323) so that the projection directions they define towards the PF are evenly distributed. Let us refer to the number of weight vectors in W as N_μ.

3.1 Initial Approximation Using the Original GWASF-GA

Initially, the original GWASF-GA [6] is run as usual for a certain number of generations. Let G_p be the number of generations performed so. We assume that G_p is defined as a percentage p $(0 < p \leq 1)$ of the total number of generations, i.e. $G_p = p \cdot G_T$. Thus, to classify the individuals into different fronts according to their ASF values until generation G_p, a half of the N_μ weight vectors is used with the utopian point and the other half is associated with the nadir point. The individuals with the lowest ASF values for the nadir and for the utopian points, using each of their weight vectors, are selected in the first front; the individuals with the next lowest ASF values form the second front, and so on.

The purpose of executing these G_p generations is to let GWASF-GA produces an initial approximation of the PF. The solutions obtained at the generation G_p may have not converged closely enough to the PF, but their distribution gives us an initial insight of the true PF. Besides, regions of the PF that are complicated to converge to may still have not been well-covered at this generation. Thus, G_p cannot be too small in order not to stop GWASF-GA too prematurely, neither too large so that there is still room to improve the search directions for new solutions.

3.2 Dynamic Weight Vectors' Adjustment

The adjustment of the weight vectors is aimed at re-orientating some of the projection directions, so that the search directions pointing to overcrowded regions are re-directed towards parts of the PF with a lack of solutions. To this aim, as previously explained, useful information about the shape of the PF can be extracted from the distribution of the solutions generated so far.

Let N_a be the number of weight vectors to be re-adapted ($N_a < N_\mu/2$) and $G_a = (1-p) \cdot G_T$ the number of generations that are left for the dynamical adjustment (i.e. $G_p + G_a = G_T$). In the next G_a generations, we perform n_a adjustments of the weight vectors (with $n_a \in \{1, \ldots, G_a\}$). This implies that the weight vectors are re-distributed each $E(G_a/n_a)$ generations, starting at generation G_p (where $E(\cdot)$ denotes the integer part of a number). That is, if G_a^r denotes the generation at which the r-th adjustment is done ($r = 1, \ldots, n_a$), we have $G_a^r = G_p + (r-1) \cdot E(G_a/n_a)$. Note that the r-th adjustment of the weight vectors is done as explained hereafter once the generation G_a^r has concluded and before running the generation $G_a^r + 1$. For each $r = 1, \ldots, n_a$, let P_a^r denote the population generated at generation G_a^r and W_a^r the new set of weight vectors obtained at the r-th adjustment. Next, we describe the procedure to identify the weight vectors to be re-calculated and how to re-distribute them.

Firstly, at the r-th adjustment, we initialize $W_a^r = W_a^{r-1}$ (with $W_a^1 = W$). To detect what regions of the PF are well-covered and what are not (according to the current population P_a^r), we need a measurement of the diversity around each individual (in the objective space). For this, we define the *scattering level* of each $\mathbf{x} \in P_a^r$, and denote it by $s(\mathbf{x})$, as follows:

$$s(\mathbf{x}) = \textstyle\prod_{j=1}^{k} L_2(\mathbf{f}(\mathbf{x}), \mathbf{f}(\mathbf{x}^j)), \qquad (2)$$

where $\mathbf{x}^1, \ldots, \mathbf{x}^k \in P_a^r$ are the k solutions with the closest objective vectors to $\mathbf{f}(\mathbf{x})$ regarding the L_2-distance. The objective vectors are assumed to be normalized to avoid scale problems. The higher (respectively, the lower) $s(\mathbf{x})$ is, the less (respectively, the more) crowded is the region where $\mathbf{f}(\mathbf{x})$ lyes. Thus, we can identify areas of the PF that are poorly approximated (with a lack of solutions) by means of the solutions with the highest scattering level, and overcrowded regions as these containing the solutions with the lowest scattering level.

Secondly, the N_a solutions in P_a^r with the lowest scattering level are selected, and we denoted them by $\{\bar{\mathbf{x}}^1, \ldots, \bar{\mathbf{x}}^{N_a}\}$. So as, it can be understood that there are close enough individuals of P_a^r around each $\bar{\mathbf{x}}^j$ in the objective space ($j = 1, \ldots, N_a$), meaning that the area of the PF where $\mathbf{f}(\mathbf{x})$ lyes has been covered enough at generation G_a^r in comparison to other regions. As previously said, each solution in GWASF-GA is associated with a weight vector, as well as with the nadir or with the utopian point. Then, we can consider that the weight vector corresponding to each $\bar{\mathbf{x}}^j$ is directing the search towards an overcrowded area of the PF, where other weight vectors are also orientating the search towards (those of the solutions around $\bar{\mathbf{x}}^j$). In view of this, the N_a weight vectors corresponding to the solutions $\{\bar{\mathbf{x}}^1, \ldots, \bar{\mathbf{x}}^{N_a}\}$ are the candidates to be replaced by new ones pointing towards the least crowded areas (according to P_a^r). Thus, they are removed from the set W_a^r, in which only $N_\mu - N_a$ weight vectors are left.

Thirdly, to generate the new N_a weight vectors, we identify the N_a solutions in P_a^r with the highest scattering level, referred to as $\{\hat{\mathbf{x}}^1, \ldots, \hat{\mathbf{x}}^{N_a}\}$. The objective vectors of these solutions enable the search to be focused on parts that have not been well-approximated by the population P_a^r, since the new weight vectors $\mu^j = (\mu_1^j, \ldots, \mu_k^j)$ $(j = 1, \ldots, N_a)$ to be introduced into W_a^r are calculated as:

Case (a): If the solution $\hat{\mathbf{x}}^j$ has been obtained in the front classification process as one minimizing the ASF with the utopian point $\mathbf{z}^{\star\star}$, then:

$$\mu_i^j = \frac{1}{f_i(\hat{\mathbf{x}}^j) - z_i^{\star\star}} \text{ for each } i = 1, \ldots, k. \quad (3)$$

Case (b): In case $\hat{\mathbf{x}}^j$ has been selected in the front classification process as one minimizing the ASF with the nadir point $\mathbf{z}^{\mathrm{nad}}$, then:

$$\mu_i^j = \frac{1}{z_i^{\mathrm{nad}} - f_i(\hat{\mathbf{x}}^j)} \text{ for each } i = 1, \ldots, k. \quad (4)$$

Once all the new weight vectors are incorporated to W_a^r, GWASF-GA is run as usual using W_a^r as the set of weight vectors, until the next weight vectors' adjustment needs to be carried out at generation G_a^{r+1}. Note that, to classify the individuals into different fronts, each new μ^j is considered to select individuals with the lowest ASF values either with the utopian point (case (a)) or with the nadir point (case (b)), depending on the way it has been calculated.

According to (3) and (4), each individual $\hat{\mathbf{x}}^j$ (lying in a not-well approximated region) is used to calculate a new weight vector μ^j, taking into account whether this solution has been elicited in GWASF-GA using the utopian or the nadir point. When using the utopian point (case (a)), μ^j determines a projection direction towards the PF which directly points to $\mathbf{f}(\hat{\mathbf{x}}^j)$ from $\mathbf{z}^{\star\star}$. But if the nadir point has been employed (case (b)), the projection direction set by μ^j is orientated to $\mathbf{f}(\hat{\mathbf{x}}^j)$ from $\mathbf{z}^{\mathrm{nad}}$. In this way, the region of the PF where each $\mathbf{f}(\hat{\mathbf{x}}^j)$ lies may be better covered in subsequent generations than before, since a new weight vector is directing the search directly towards it, from the utopian or from the nadir point, assuring that the best one of them is used.

Observe that, after the first adjustment, the number of weight vectors used in the ASF for the utopian point is no longer equal to that for the nadir point, so as in the original GWASF-GA. In case the PF is convex, it is likely that the process automatically assigns a larger amount of weight vectors to the nadir point, giving more importance to the projection from it. But if it is concave, more new weight vectors may be likely to be associated with the utopian point, meaning that its projection is more suitable.

4 Experimental Study

The adjustment of the weight vectors in GWASF-GA depends on three parameters: the number of dynamic adjustments to be performed (n_a), the percentage of generations that GWASF-GA is run before the first adjustment is applied (p), and the number of weight vectors to be adjusted (N_a). In this section, we test different settings of these parameters in order to gain knowledge about the improvement performance of GWASF-GA achieved using different configurations.

4.1 Experimental Design

A total of 46 test problems are used: 18 with three objectives (DTLZ1-DTLZ4 and DTLZ7 [2], WFG1-9 [3], UF8-10 [9], and LZ09 [4]) and 14 with five and six objectives, respectively (DTLZ1-DTLZ4 and DTLZ7 [2] and WFG1-9 [3]). We set the number of decision variables to $k+4$ for DTLZ1, $k+9$ for DTLZ2-DTLZ4, and $k+19$ for DTLZ7 (k is the number of objectives). For the WFG problems, the position- and distance-related parameters are $k-1$ and 10, respectively.

In our analysis, we use a fractional factorial design to investigate the impact of n_a, p, and N_a on the performance of GWASF-GA. We analyze the approximations generated by GWASF-GA when the weight vectors are adjusted using all possible combinations of the following values: $n_a \in \{2, 4, 6\}$, $p \in \{0.6, 0.7, 0.8\}$, and $N_a \in \{5, 20, 25, 30, 50, 60, 75, 100\}$.[1] Thus, we study 72 different configurations to dynamically adjust the weight vectors and we compare their performance against the original GWASF-GA (i.e. without weight vectors' adjustment).

For each algorithm (meaning each version of GWASF-GA with each possible dynamic adjustment configuration, in addition to the original GWASF-GA), 30 independent runs are executed for each test problem, which implies more than 100,000 runs ($30 \times (72+1) \times (18+14+14)$). In all cases, we use the same evolutionary parameters: 300 individuals (i.e. $N_\mu = 300$ weight vectors), 3,000 generations, the SBX crossover operator with a distribution index $\eta_c = 20$ and a probability $P_c = 0.9$, and the polynomial mutation operator with a distribution index $\eta_m = 20$ and a probability $P_m = 1/n$, where n is the number of variables.

To run the experiments, we use the implementations of GWASF-GA and of the test problems available in jMetal [5], an object-oriented Java-based framework for multiobjective optimization using metaheuristics. We conduct our experiments in a cluster of 21 computers offering a total of 172 cores and 190 GB of memory. The cluster is managed by HTCondor, a specialized workload management system for compute-intensive jobs.[2] We use HTCondor because it provides a job queuing mechanism, scheduling policy, and resource management that allow users to submit parallel jobs to HTCondor.

4.2 Data Analysis

The hypervolume indicator [10] has been used as performance metric, being able to measure both convergence and diversity of the solutions generated in the objective space. To compute the hypervolume, a representative set of the PF is required. For the DTLZ and WFG problems, we generate it using an open-source tool.[3] For the rest of problems, we use the representative sets available in jMetal.

Given that we carry out 30 independent runs of each algorithm, we apply a Wilcoxon rank-sum test [8] to check if the hypervolume achieved by the original GWASF-GA is significantly different to that of its version adjusting the weight

[1] These values reported the best results after performing several initial tests. The results are not included due to space limitations and are available upon request.

[2] https://research.cs.wisc.edu/htcondor/index.html

[3] https://github.com/rsain/Pareto-fronts-generation

vectors and using each one of the configurations considered. For each problem, the null hypothesis is that the distribution of their hypervolume average values in the 30 runs differ by a location shift of α. Thus, we consider the difference to be significant if the obtained p-value is lower than $\alpha = 0.05$. To compute the Wilcoxon rank-sum test, we use the `wilcox.test` function from the R software.[4]

4.3 Results

Table 2 reports the number of problems where each version of GWASF-GA with the weight vectors' adjustment performs significantly better than (▲) and worse than (▽) the original GWASF-GA. Note that the amount of problems in which the performance did not significantly differ can be known using the total number of problems. The column '*Configuration*' shows the values of the parameters n_a, p, and N_a. The second, third, and fourth columns show the results obtained for the three-, five-, and six-objective problems, respectively ($k = 3, 5, 6$). In these columns, we highlight in gray color the configurations of the weight vectors' adjustment that perform better than the original algorithm in the highest number of cases, reaching an equal performance for the rest of them at the same time.

We can see that the hypervolume achieved by GWASF-GA is higher when using the weight vectors' adjustment, on average, in 15/18 (81%), 11/14 (76%), and 10/14 (74%) of the cases, respectively, for three, five, and six objectives. In addition, for each group of problems, there always exists, at least, one configuration (or more) that never performs worse than the original GWASF-GA. Therefore, we can conclude that adjusting the weight vectors enables the outcome of GWASF-GA to be enhanced, although we must say that there is no unique configuration reporting the best results for all the cases.

5 Conclusion

In this paper, we have proposed an improvement of the decomposition-based EMO algorithm GWASF-GA. This algorithm uses weight vectors to classify the individuals into several fronts, which determine the search directions for new non-dominated solutions. Thus, we have suggested to perform a dynamic adjustment of the weight vectors to enhance its convergence and diversity. Based on the population generated, we identify regions of the PF that are overcrowded and others that have been poorly approximated. Thus, some weight vectors are replaced so that search directions pointing to overcrowded regions are re-directed towards parts with a lack of solutions that may be hard to be approximated.

The performance of our proposal depends on several parameters, such as the number of weigh vectors to be replaced, the generation at which the first adjustment is carried out, and the times the weight vectors are adjusted. In the numerical experiments, we have tested different configurations of these parameters to solve three-, five-, and six-objective optimization problems. According to

[4] `https://stat.ethz.ch/R-manual/R-devel/library/stats/html/wilcox.test.html`

Fig. 2. Number of problems with three, five and six objectives ($k = 3, 5, 6$) where each version of GWASF-GA with the weight vectors' adjustment performs better than (▲) and worse than (▽) the original GWASF-GA, for the hypervolume in the 30 independent runs.

Configuration			$k=3$ 18 prob.		$k=5$ 14 prob.		$k=6$ 14 prob.	
n_a	p	N_a	▲	▽	▲	▽	▲	▽
2	0.6	5	13	2	10	2	9	3
2	0.6	20	16	0	13	0	11	3
2	0.6	25	16	0	13	0	10	3
2	0.6	30	16	0	12	0	11	0
2	0.6	50	16	0	12	0	10	2
2	0.6	60	16	0	12	0	11	2
2	0.6	75	16	0	9	0	11	3
2	0.6	100	12	2	8	4	11	2
2	0.7	5	14	1	8	2	9	3
2	0.7	20	16	0	12	0	10	3
2	0.7	25	16	0	13	0	10	3
2	0.7	30	17	0	13	0	11	3
2	0.7	50	16	0	12	0	11	3
2	0.7	60	16	0	11	0	10	2
2	0.7	75	15	0	10	1	10	2
2	0.7	100	12	1	8	4	12	2
2	0.8	5	13	0	10	1	9	3
2	0.8	20	16	0	13	0	10	3
2	0.8	25	17	0	13	0	11	3
2	0.8	30	16	0	13	0	10	3
2	0.8	50	16	0	13	0	11	2
2	0.8	60	16	0	12	0	11	2
2	0.8	75	16	0	10	0	11	2
2	0.8	100	12	3	8	4	10	3
4	0.6	5	14	2	11	2	10	3
4	0.6	20	16	0	13	0	11	3
4	0.6	25	16	0	13	0	11	3
4	0.6	30	16	0	13	0	11	3
4	0.6	50	16	0	13	0	11	2
4	0.6	60	16	0	13	0	11	2
4	0.6	75	15	0	11	1	11	1
4	0.6	100	12	2	7	4	12	1
4	0.7	5	15	1	9	1	10	3
4	0.7	20	16	0	13	0	10	2
4	0.7	25	17	0	13	0	11	2
4	0.7	30	16	0	13	0	12	2
4	0.7	50	16	0	13	0	10	2
4	0.7	60	16	0	12	0	11	2
4	0.7	75	15	0	10	0	12	2
4	0.7	100	12	2	7	4	12	1
4	0.8	5	15	1	12	2	10	3
4	0.8	20	16	0	13	0	10	3
4	0.8	25	17	0	13	0	12	2
4	0.8	30	16	0	13	0	12	2
4	0.8	50	17	0	13	0	12	2
4	0.8	60	16	0	13	0	11	2
4	0.8	75	17	0	10	0	4	11
4	0.8	100	12	2	7	4	11	1
6	0.6	5	14	1	11	1	11	3
6	0.6	20	17	0	13	0	10	2
6	0.6	25	16	0	13	0	12	2
6	0.6	30	16	0	13	0	11	2
6	0.6	50	16	0	13	0	11	1
6	0.6	60	16	0	13	0	12	1
6	0.6	75	16	0	10	0	13	1
6	0.6	100	13	3	7	5	11	1
6	0.7	5	15	2	9	1	10	3
6	0.7	20	16	0	13	0	12	2
6	0.7	25	16	0	13	0	12	2
6	0.7	30	17	0	13	0	12	2
6	0.7	50	16	0	13	0	12	1
6	0.7	60	16	0	13	0	12	1
6	0.7	75	16	0	11	0	13	1
6	0.7	100	12	3	8	5	11	1
6	0.8	5	15	1	11	0	11	3
6	0.8	20	16	0	13	0	11	2
6	0.8	25	16	0	13	0	12	2
6	0.8	30	16	0	13	0	11	2
6	0.8	50	16	0	13	0	12	1
6	0.8	60	16	0	13	0	13	1
6	0.8	75	16	0	11	1	12	0
6	0.8	100	12	3	7	4	12	1

the hypervolume, we have concluded that better results are provided when the weight vectors' adjustment is incorporated into GWASF-GA, in comparison to the outcome generated by the original algorithm. As future research, we plan to analyze the performance of our proposal as compared to other EMO algorithms.

Acknowledgements

This research is funded by the Government of Spain (project ECO2017-88883-R and network ECO2017-90573-REDT) and by the Andalusian Regional Government (PAI group SEJ-532). Sandra González-Gallardo is recipient of a technical research contract within "Sistema Nacional de Garantía Juvenil y del Programa Operativo de Empleo Juvenil 2014-2020 - Fondos FEDER", and thanks the training received from the University of Málaga PhD Programme in Economy and Business. Ana B. Ruiz is recipient of a post-doctoral fellowship of "Captación de Talento para la Investigación" at the University of Málaga.

References

1. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems. Springer, New York, 2nd edn. (2007)
2. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable multi-objective optimization test problems. In: Congress on Evolutionary Computation. pp. 825–830 (2002)
3. Huband, S., Hingston, P., Barone, L., While, L.: A review of multi-objective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation **10**(5), 477–506 (2007)
4. Li, H., Zhang, Q.: Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE Transactions on Evolutionary Computation **12**(2), 284–302 (2009)
5. Nebro, A.J., Durillo, J.J., Vergne, M.: Redesigning the jMetal multi-objective optimization framework. In: Conference on Genetic and Evolutionary Computation. pp. 1093–1100 (2015)
6. Saborido, R., Ruiz, A.B., Luque, M.: Global WASF-GA: An evolutionary algorithm in multiobjective optimization to approximate the whole Pareto optimal front. Evolutionary Computation **25**(2), 309–349 (2017)
7. Wierzbicki, A.P.: The use of reference objectives in multiobjective optimization. In: Fandel, G., Gal, T. (eds.) Multiple Criteria Decision Making, Theory and Applications. pp. 468–486. Springer (1980)
8. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
9. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the CEC 2009 special session and competition. Technical Report (CES-487, University of Essex and Nanyang Technological University) (2008)
10. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. IEEE Transactions on Evolutionary Computation **3**(4), 257–271 (1999)