Δ_p-MOEA: A New Multi-Objective Evolutionary Algorithm Based on the Δ_p Indicator

Adriana Menchaca-Mendez*
CINVESTAV-IPN,
Departamento de Computación
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO

Carlos Hernández†
CINVESTAV-IPN,
Departamento de Computación
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO

Carlos A. Coello Coello‡
CINVESTAV-IPN,
Departamento de Computación
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO

***Abstract*—In this paper, we propose a new selection scheme for Multi-Objective Evolutionary Algorithms (MOEAs) based on the Δ_p indicator. Our new selection scheme is incorporated into a MOEA giving rise to the "Δ_p-MOEA." Perhaps, one of the most important disadvantages of MOEAs based on Δ_p is the definition of the reference set. In this work, we propose to create a reference set at each generation using ϵ-dominance and the set of nondominated solutions found so far. Our new selection scheme uses two different techniques to select solutions according to the modified generational distance indicator or the modified inverted generational distance indicator. Our proposed Δ_p-MOEA is validated using standard test functions taken from the specialized literature, having three to six objective functions and it is compared with respect to two well-known MOEAs: MOEA/D using Penalty Boundary Intersection (PBI), which is based on decomposition, and SMS-EMOA-HYPE (a version of SMS-EMOA that uses a fitness assignment scheme based on the use of an approximation of the hypervolume indicator). Our preliminary results indicate that our Δ_p-MOEA is a good alternative to solve MOPs with low and high dimensionality (in objective function space) since it outperforms MOEAs such as MOEA/D and SMS-EMOA-HYPE in several problems and its computational cost is reasonably low (it is slower than MOEA/D but is faster than SMS-EMOA-HYPE).**

I. Introduction

In a wide range of applications, conflicting objectives arise. In such cases, we are interested not in a single solution but rather on the best possible trade-offs among the objectives. For this purpose, many MOEAs have been proposed that are capable of solving these problems, and such MOEAs work very well when dealing with two and three objectives. However, as the number of objectives increases, the quality of the solutions given by traditional MOEAs (i.e., based on Pareto ranking) tends to deteriorate [1].

Recently, there has been an increasing interest for the use of different selection schemes, namely the use of indicators[1] to determine the quality of a given solution. Such indicators are well-suited for the problem at hand, since they are able to assign the quality with a single real number, even with an increasing number of objectives. This feature makes them attractive to attack problems with many objectives, in which the use of Pareto dominance fails.

Here, we present a new evolutionary algorithm based on the Δ_p indicator and we compare it to some state-of-the-art algorithms on the Walking-Fish-Group (WFG) test problems [2]. The remainder of this paper is organized as follows: Section II, presents the necessary background on Multi-Objective Optimization Problems (MOPs) and the Δ_p indicator. In Section III, we discuss the key elements that need to be considered for the selection mechanism. Next, in Section IV, we present our proposed algorithm Δ_p-MOEA. Section V, contains our comparison with respect to state-of-the-art algorithms using the hypervolume indicator [3] and the Δ_p indicator [4]. Finally, Section VI contains our conclusions and some possible paths for future work.

II. Background

A. Multi-Objective Optimization Problems (MOPs)

We are interested in the general *multi-objective optimization problem*, which is defined as follows: Find $\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which optimizes

$$\vec{f}(\vec{x}) = [f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})]^T \quad (1)$$

such that $\vec{x}^* \in \Omega$, where $\Omega \subset R^n$ defines the feasible region of the problem. Assuming minimization problems, we have the following definitions.

Definition 1: We say that a vector $\vec{u} = [u_1, \ldots, u_n]^T$ dominates vector $\vec{v} = [v_1, \ldots, v_n]^T$, denoted by $\vec{u} \leq_p \vec{v}$, if and only if $f_i(\vec{u}) \leq f_i(\vec{v})$ for all $i \in \{1, ..., k\}$ and there exists an $i \in \{1, \ldots, k\}$ such that $f_i(\vec{u}) < f_i(\vec{v})$.

Definition 2: A point $\vec{x}^* \in \Omega$ is Pareto optimal if there does not exist $\vec{x} \in \Omega$ such that $\vec{x} \leq_p \vec{x}^*$.

Definition 3: A point $\vec{x} \in \Omega$ is weakly Pareto optimal if there does not exist another point $\vec{y} \in \Omega$ such that $f_i(\vec{y}) < f_i(\vec{x})$ for all $i \in \{1, ..., k\}$.

Definition 4: For a given MOP, $\vec{f}(\vec{x})$, the Pareto optimal set is defined as: $\mathcal{P}^* = \{\vec{x} \in \Omega | \neg\exists \vec{y} \in \Omega : \vec{f}(\vec{y}) \leq_p \vec{f}(\vec{x})\}$.

Definition 5: Let $\vec{f}(\vec{x})$ be a given MOP and $\mathcal{P}^*$ the Pareto optimal set. Then, the Pareto Front is defined as: $\mathcal{PF}^* = \{\vec{f}(\vec{x}) \mid \vec{x} \in \mathcal{P}^*\}$.

*adriana.menchacamendez@gmail.com
†chernandez@computacion.cs.cinvestav.mx
‡ccoello@cs.cinvestav.mx

[1] An m-ary quality indicator I is a function $I : \Omega^m \rightarrow \mathbb{R}$, which assigns each vector $(\mathcal{A}_1, \mathcal{A}_2, \cdots, \mathcal{A}_m)$ of m approximation sets a real value $I(\mathcal{A}_1, \cdots, \mathcal{A}_m)$.

B. Δ_p Indicator

The Δ_p indicator was proposed in [4]. The authors based this indicator on slight modifications of the well-known I_{GD} [5] and I_{IGD} [6] indicators. Further, the authors used the indicator to assess performance of state-of-the-art algorithms showing its capability to evaluate the quality of the solutions produced by a MOEA. Δ_p can be seen as an "averaged Hausdorff distance" between the image of $\mathcal{A}$, where $\mathcal{A}$ is an approximation of the Pareto optimal set, and the true Pareto front. Next, we present the formal definition of this indicator. Given a candidate set $\mathcal{A} = \{a_1, \ldots, a_j\}$ and a discretization of the Pareto front $F(P_s) = \{y_1, \ldots, y_z\}$, then:

$$I_{GD_p}(\mathcal{A}, F(P_s)) := \left(\frac{1}{j}\sum_{i=1}^{j} d_i^p\right)^{1/p} \quad (2)$$

where d_i denotes the minimal Euclidean Distance from $a_i \in \mathcal{A}$ to $F(P_s)$. $I_{GD_p}(\mathcal{A}, F(P_s))$ is the averaged semi-distance from the image of $\mathcal{A}$ to the discretization of the Pareto front and

$$I_{IGD_p}(\mathcal{A}, F(P_s)) := \left(\frac{1}{z}\sum_{i=1}^{z} \tilde{d}_i^p\right)^{1/p} \quad (3)$$

where $\tilde{d}_i$ denotes the minimal Euclidean Distance from $y_i \in F(P_s)$ to $\mathcal{A}$. $I_{IGD_p}(\mathcal{A}, F(P_s))$ is the averaged semi-distance of the y_i's to the image of $\mathcal{A}$. Finally, we define Δ_p as:

$$\Delta_p := \max(I_{GD_p}(\mathcal{A}, F(P_s)), I_{IGD_p}(\mathcal{A}, F(P_s))) \quad (4)$$

As we know, to compute I_{GD} and I_{IGD}, it is necesary to know the Pareto optimal front. Thus, MOEAs based on Δ_p need to define a reference set which must contain well-distributed solutions. If we are not capable of producing a good reference set, it is possible to lose diversity in the population which might lead to an incomplete Pareto front, or a poor distribution of the solutions. This can be seen as the major drawback in MOEAs based on Δ_p.

C. Related Work

Now, we briefly review the MOEAs that have adopted Δ_p. In [7], an algorithm based on GDE3 was proposed that uses a selection scheme based on Δ_p. In this case, a preference relation is introduced by: computing the contribution to I_{GD}, computing the distance to the closest element in the approximation of the Pareto front and to I_{IGD} assigning the distance from a reference point to a solution if it is the closest or assigning -1 if the solution is not the closest to any point in the reference. If the same individual is the closest for several reference points, then it takes the p-norm of the distances. This is done for each individual and then the algorithm selects the best individuals according to these contributions. In this case, the algorithm gives a preference to the I_{IGD} contributions over I_{GD}.

A different approach to use Δ_p can be found in [8] where the authors proposed an archive-based plug-in method that builds an evenly spaced approximation, using the averaged Hausdorff measure between the archive and a reference front. In this case, the approach is used to solve problems having three objectives and the reference font is constructed from a triangulated approximation of the Pareto front obtained from a previous experiment (offline), although a construction from the best known solutions (online) can be used, too. In this case any MOEA can be adopted and the archive will keep the solutions that are better according to the Δ_p indicator. In [9], the authors proposed to use an archive similar to the one adopted with SMS-EMOA. In this case, the Part and Selection Algorithm (PSA) is used to construct the reference front and both the offline and the online versions are adopted.

In further work, a different kind of multi-objective optimizer was proposed [10]. In this case, the aim was to steer the search towards an aspiration set (the reference set is given by the aspiration set). In this work, the selection is performed by the Δ_p indicator and, in order to do it, the algorithm generates one offspring at a time. Then, the contribution to the Δ_p indicator is computed in a similar way as done in SMS-EMOA and then, the individual with the minimum contribution to the indicator is deleted. A similar approach is used in [11]. However, in this case, the algorithm is able to generate its own reference set by using a family of curves that can be either convex, linear or concave.

Note that except for the approaches reported in [7], [8], all the other MOEAs based on Δ_p only generate one offspring at a time. In such approaches, if one would like to generate more offspring (for instance to benefit from different search engines), then selection would become computationally expensive. Regarding [8], the approach is not specifically designed around Δ_p, and the search mechanism is not affected by the indicator, which might leave regions of the search space that are good for Δ_p without being correctly explored. In this paper, we generalize the selection mechanism used in the previously indicated works and simplify the one used in [7].

III. A New Selection Mechanism based on the Δ_p indicator

As we mentioned before, the definition of the reference set is one of the most important aspects to consider when we intend to use the Δ_p indicator as a selection mechanism. In this work, we propose to build the reference set at each generation using the set of nondominated solutions found so far, combined with the definition of ϵ-dominance. We know that the Δ_p indicator is based on I_{GD} and I_{IGD}. However, these indicators have some disadvantages when they are used to select individuals, e.g., if we use I_{GD}, we know that this indicator only measures convergence and, therefore, its use can cause a quick loss of diversity. If we use I_{IGD}, it is possible that we cannot assign fitness to some individuals in the population. For this reason, we propose to use two different selection mechanisms which address these possible disadvantages. Our selection mechanism based on the Δ_p indicator works as follows: First, we define the reference set. After that, we calculate the Δ_p indicator and we decide if we select using a selection mechanism based on I_{GD} or if we select using a selection mechanism based on I_{IGD}. Algorithm 1 shows a general description of our proposed selection mechanism, called "Δ_p-selection".

Algorithm 1: Δ_p-Selection

Input : $\mathcal{P}$ (population), N (number of individuals to choose $N < \|\mathcal{P}\|$).
Output: $\mathcal{S}$ (selected individuals).

Obtain the set of nondominated solutions from $\mathcal{P}$ and call it $\mathcal{A}$;
Generate the reference set $\mathcal{RS}$ using $\mathcal{A}$ and ϵ-dominance as we explain in Section III-A;
Calculate $I_{GD}(\mathcal{P})$ and $I_{IGD}(\mathcal{P})$ using $\mathcal{RS}$;
if $I_{GD}(\mathcal{P}) > I_{IGD}(\mathcal{P})$ **then**
    Select the N solutions using I_{GD} as we explain in Section III-B and put the selected individuals in $\mathcal{S}$;
else
    Select the N solutions using I_{IGD} as we explain in Section III-C and put the selected individuals in $\mathcal{S}$;
end
return $\mathcal{S}$;

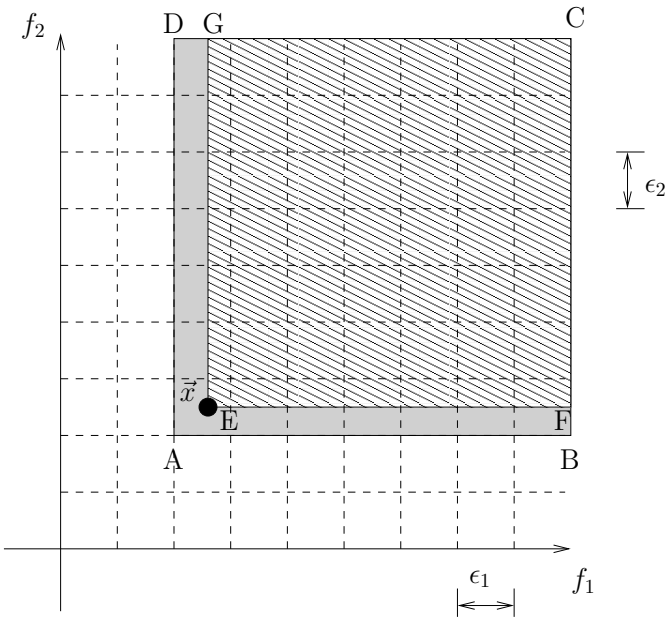

Figure 1. ϵ-dominance. The area (ABCDA) is ϵ-dominated by $\vec{x}$. Using Pareto dominance; $\vec{x}$ dominates any solution in the area EFCGE.

A. Reference Set

We define a reference set from the nondominated solutions and adopting ϵ-dominance as described next. In order to determine the ϵ-dominance between two solutions, it is necessary to calculate an identification array $\vec{b}_i$ for each solution $\vec{x}_i$. Each component of $\vec{b}_i$ is defined as follows:

$$b_{i,j} = \lfloor (f_j(\vec{x}_i) - f_j^{min})/\epsilon_j \rfloor \quad (5)$$

where f_j^{min} is the minimum value of the j-th objective and ϵ_j is a tolerance in the j-th objective. The identification arrays divide the whole objective space into hypercubes, each having ϵ_j size in the j-th objective. Figure 1 illustrates the concept of ϵ-dominance. It is important to note that all points in the same hypercube have the same identification array.

Our proposal is to define an ϵ vector, $\vec{\epsilon} = [\epsilon_1, \cdots, \epsilon_k]$, such that each nondominated solution is placed in a different hypercube. Then, the reference set will consist of the points at the bottom left corners of each hypercube which contains a nondominated solution. If more than one nondominated solution is placed in the same hypercube, the size of the reference set will be smaller than the size of the set of nondominated solutions. Let $\mathcal{A} = \{\vec{x}_1, \cdots, \vec{x}_n\}$ be the set of nondominated solutions and let $\vec{\epsilon} = [\epsilon_1, \cdots, \epsilon_k]$ be a vector of the allowable tolerance in each objective function. We thus propose to build a reference set, $\mathcal{RS} = \{\vec{r}_1, \cdots, \vec{r}_m\}$, from the identification arrays $\vec{b}$, as follows: First, we normalize the nondominated solutions. After that, we calculate the identification arrays for each nondominated solution, and finally, we obtain the bottom left corners of each hypercube that has a nondominated solution using the following equation:

$$r_{i,j} = \epsilon_j * b_j^i. \quad (6)$$

It is important to check weak dominance between solutions in the reference set: If two points are equal in $k-1$ components, where k is the number of objective functions, then we only select the individual which is closest to the origin. Figure 2 shows one example. An important question that we must ask ourselves is how we can define $\vec{\epsilon}$ and this task is not easy. This is, indeed, an important disadvantage of MOEAs based on ϵ-dominance. Recently, a new MOEA based on I_{GD} and ϵ-dominance was proposed in [12]. An interesting aspect of this work is that the authors proposed a method to set ϵ at each generation. Here, we propose to set ϵ in the same way as in this previous work, and then, we do the following: First, we divide each objective function in two equal parts:

$$\epsilon_j = (f_j^{max} - f_j^{min})/2 \quad (7)$$

where f_j^{max} and f_j^{min} are the maximun and the minimum values for the objective function j, considering only the nondominated solutions. Since we normalize according to the nondominated solutions (in this case $\epsilon_j = 0.5$), then, we proceed to build the reference set as we explained above. If the size of the reference set is less than the size of the nondominated set ($|\mathcal{RF}| < |\mathcal{A}|$), we divide each objective function in three equal parts:

$$\epsilon_j = (f_j^{max} - f_j^{min})/3 \quad (8)$$

We repeat this process (increasing the number of hypercubes in which the search space is divided) aiming that the reference set contains the same number of elements as the nondominated set: $|\mathcal{RF}| = |\mathcal{A}|$. However, if we increase the number of hypercubes three consecutive times and we can not increase the size of the reference set, we stopped. We adopted three consecutive times as our second stop condition in an arbitrary way.

B. Selection using I_{GD}

We use the selection mechanism proposed in [13]. This is due to the following: If we select optimizing I_{GD}, we can lose diversity in the population and it is possible that the MOEA cannot generate the complete Pareto front, or even, it is possible that the MOEA cannot find the true Pareto front. The selection mechanism proposed in [13] works as follows: Let $\mathcal{RS}$ be the reference set, $\mathcal{P}$ be the current population and N be the number of solutions that we want to select. Then,

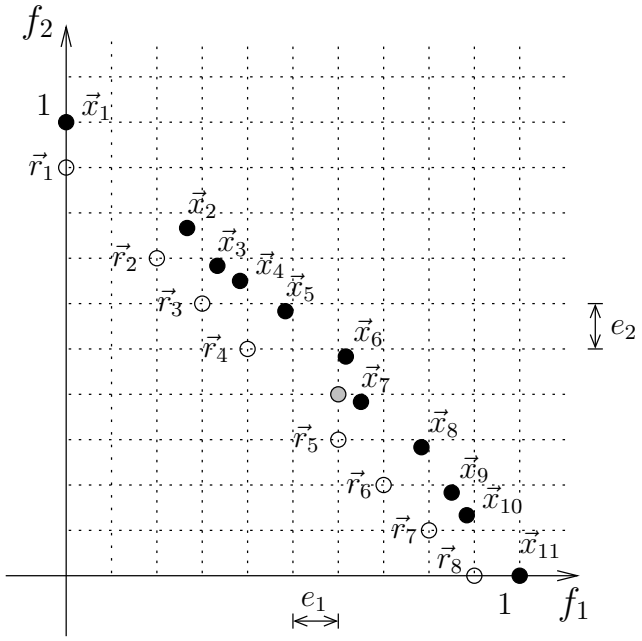

Figure 2. Let $\mathcal{A} = [\vec{x}_1, \cdots, \vec{x}_{10}]$ be a set of nondominated solutions and $\vec{\epsilon} = [0.1, 0.1]$. Then, the reference set, $\mathcal{RS}$, is composed by $\vec{r}_1 = [0, 0.9]$, $\vec{r}_2 = [0.2, 0.7]$, $\vec{r}_3 = [0.3, 0.6]$, $\vec{r}_4 = [0.4, 0.5]$, $\vec{r}_5 = [0.6, 0.3]$, $\vec{r}_6 = [0.7, 0.2]$, $\vec{r}_7 = [0.8, 0.1]$ and $\vec{r}_8 = [0.9, 0]$. We can see that the size of the reference set is less than the size of $\mathcal{A}$ because in two hypercubes there are two nondominated solutions and the reference point in gray is not selected because is equal to $\vec{r}_5$ in $k-1$ components and $\vec{r}_5$ is closer to the origin.

we calculate the distance d_i from each solution in $\mathcal{P}$ to its nearest neighbor in $\mathcal{RS}$. Also, it is necessary to save its neighboring reference point. After that, we sort $\mathcal{P}$ according to d_i and create the set $\mathcal{S} = \emptyset$. Finally, for each $\vec{x}_i \in \mathcal{P}$, we check if its nearest neighbor in $\mathcal{RS}$ is the same to other solution in $\mathcal{S}$. If not and $|\mathcal{S}| < N$, we include $\vec{x}_i$ in $\mathcal{S}$. If all solutions in $\mathcal{P}$ are considered and $|\mathcal{S}| < N$, we repeat the process but now we will allow that only one solution in $\mathcal{S}$ has the same neighbor that the solution that we want to select. We have to iterate this process until we obtain N solutions. Figure 3 shows an example of this procedure.

C. Selection using I_{IGD}

When we select individuals using I_{IGD}, we must consider that we cannot assign fitness to each solution in the population in a direct way (as in I_{GD}) because we need to calculate the minimum distance from each point in the reference set to its nearest neighbor in the population. For this reason, we propose the following selection mechanism based on I_{IGD}: Let $\mathcal{RS}$ be the reference set, $\mathcal{P}$ be the current population and N the number of solutions that we want to select. First, we create the set $\mathcal{S} = \emptyset$ and obtain the nearest neighbor from each reference point to the population. If one solution in the population is the nearest neighbor of two or more reference points, then it is only associated to its nearest reference point. Then, it is necessary to find the nearest neighbor of the reference points which don't have an associated solution, i.e., all reference points must have assigned a different solution from the population. If $|\mathcal{S}| + |\mathcal{RS}| \leq N$, then we select all solutions associated with the reference points. Otherwise, we choose the solutions which are closer to its reference point. If after this process, we have not selected all the N solutions, then, we repeat the process but this time, we do not consider the solutions already selected. Figure 4 shows an example of this procedure.

IV. Δ_p-MOEA: A NEW MOEA BASED ON THE Δ_p INDICATOR

Δ_p-MOEA is a MOEA that uses the operators of NSGA-II [14] (crossover and mutation) to create new individuals and it works as follows: First, it creates an initial population of size P. After that, it creates P new individuals and it combines the population of parents and offspring (we obtain a population of $2P$ individuals), and finally, it uses the selection mechanism proposed in Section III to select the P individuals that will take part of the next generation. It repeats this process during a pre-defined number of generations.

V. EXPERIMENTAL RESULTS

We validated our Δ_p-MOEA comparing it with respect to MOEA/D using Penalty Boundary Intersection (PBI)[2] and with respect to a version of SMS-EMOA that uses a fitness assignment scheme based on the use of an approximation of the hypervolume indicator. We called this version SMS-EMOA-HYPE[3]. In the case of SMS-EMOA-HYPE, we used the source code of HyPE available in the public domain [15] adopting 10^4 as our number of samples to assign fitness. In the case of MOEA/D, we generated the convex weights using the technique proposed in [16] and then, we applied clustering (k-means) to obtain a specific number of weights [4].

For our experiments, we used seven problems taken from the WFG toolkit [2], with $k\_factor = 2$ and $l\_factor = 10$. For each test problem, we performed 30 independent runs. For all three algorithms, we adopted the parameters suggested by the authors of NSGA-II: $p_c = 0.9$ (crossover probability), $p_m = 1/n$ (mutation probability), where n is the number of decision variables. We also used $\eta_c = 15$ and $\eta_m = 20$.[5] We performed a maximum of 30,000 fitness function evaluations (in this case, we used a population size of 100 individuals and we iterated for 300 generations).

A. Performance Indicators

We used the hypervolume indicator (I_H) to compare performance because it is "Pareto compliant" and it rewards both convergence towards the Pareto front as well as the

[2]We made the experiments using both versions: (i) MOEA/D using Penalty Boundary Intersection and (ii) MOEA/D using the Tchebycheff function. However, the first version obtained better results than the second version. For this reason, we decided to include only the comparison with respect to the first version.

[3]We designed SMS-EMOA-HYPE because our aim was to validate the effect of our selection mechanism. Therefore, we required that all the MOEAs being compared, created the individuals in the same way in order to allow for a fair comparison.

[4]The source code of the three algorithms (MOEA/D, SMS-EMOA-HYPE and Δ_p-MOEA) can be provided by the first author upon request. For MOEA/D, we used the source code available in the MOEA/D webpage.

[5]Our aim was to compare the selection mechanism used by each MOEA. Therefore, we wanted to know the behavior of each MOEA given the same configuration of parameters.

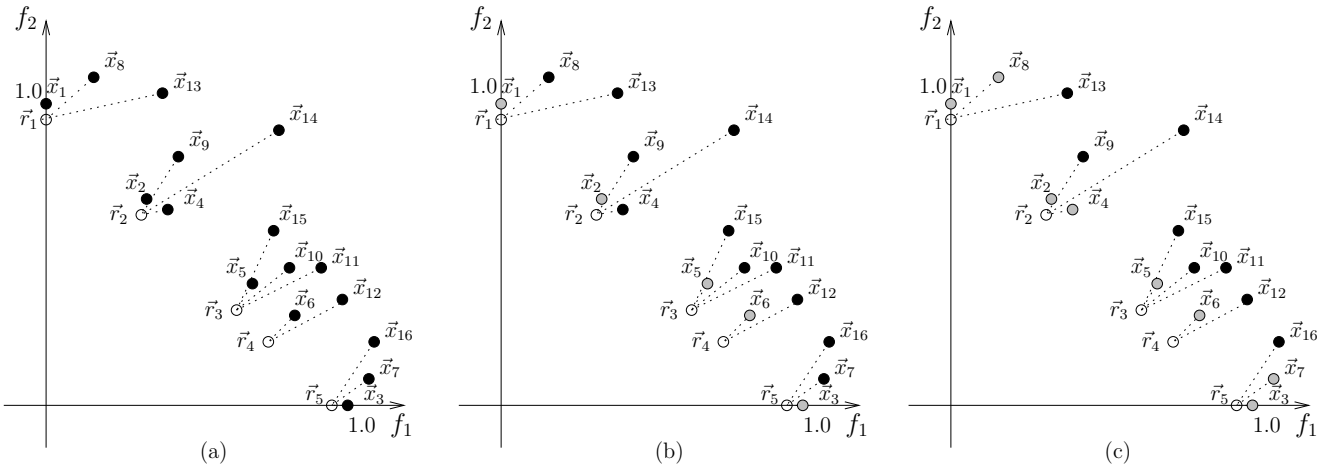

Figure 3. Selection using I_{GD}. Let's assume that we want to select eight solutions of a population of sixteen individuals (black points). Firts, in (a), we define the reference set $\mathcal{RS}$ (white points) and we calculate the distances from each solution in the population to its nearest neighbor in $\mathcal{RS}$. In (b), we perform the first iteration (each reference point can only have one point associated) and we only select five solutions (gray points). In (c), we perform the second iteration (each reference point can have only two points associated) and we can select the eight points (gray points).

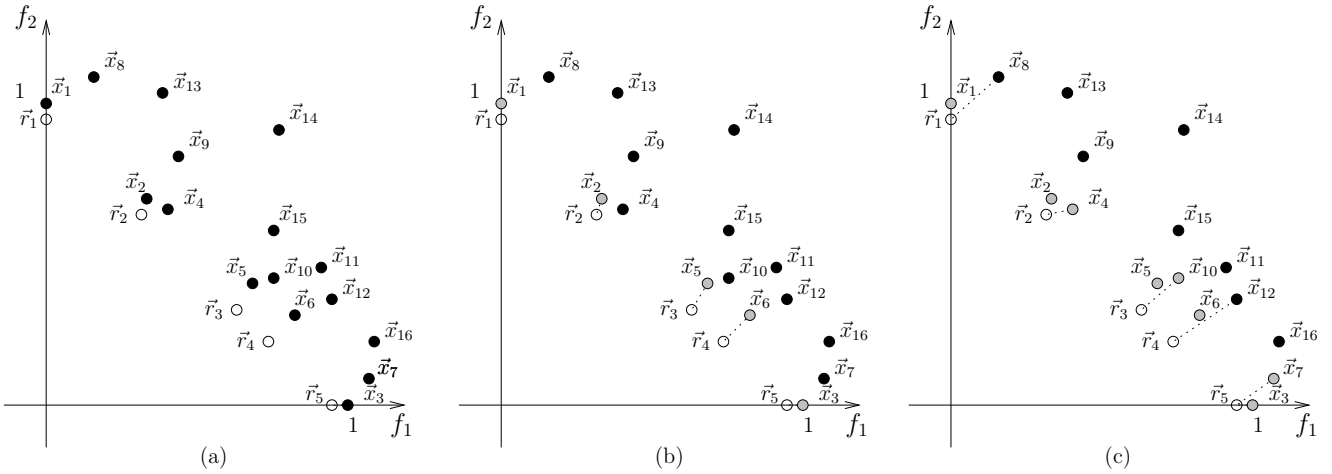

Figure 4. Selection using I_{IGD}. Let's assume that we want to select eight solutions from a population of sixteen individuals (black points). First, in (a), we define the reference set $\mathcal{RS}$ (white points). In (b), we perform the first iteration, we calculate the distances from each reference point to its nearest neighbor in $\mathcal{P}$ and, finally, we select the five nearest neighbors (gray points) because the condition $|\mathcal{S}| + |\mathcal{RS}| \leq N$ is true $(0 + 5 < 8)$. In (c), we perform the second iteration, we calculate the distances from each reference point to its nearest neighbor in $\mathcal{P}$ without considering the already selected points (gray points). We can see that the point $\vec{x}_{10}$ is the nearest neighbor of $\vec{r}_3$ and $\vec{r}_4$ but $\vec{r}_3$ is closer to $\vec{x}_{10}$ than $\vec{r}_4$. Therefore, $\vec{x}_{10}$ is associated to the reference point $\vec{r}_3$ and we associate $\vec{x}_{12}$ with $\vec{r}_4$. After that, since $|\mathcal{S}| + |\mathcal{RS}| \leq N$ is not true $(5 + 5 < 8)$, we choose the solutions which are closer to its associated reference points, and we choose $\vec{x}_4$, $\vec{x}_7$ and $\vec{x}_{10}$.

maximum spread of the solutions obtained. If Λ denotes the Lebesgue measure, I_H is defined as:

$$I_H(\mathcal{A}, \vec{y}_{ref}) = \Lambda\left(\bigcup_{\vec{y} \in \mathcal{A}} \{\vec{x} \mid \vec{y} \prec \vec{x} \prec \vec{y}_{ref}\}\right) \qquad (9)$$

where $\mathcal{A}$ is the approximation of the Pareto optimal set and $\vec{y}_{ref} \in \mathbb{R}^k$ denotes a reference point which should be dominated by all possible points. To calculate I_H, we normalized the approximations of the Pareto optimal set, generated by the MOEAs, and we used $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$, as our reference point. The normalization was performed considering all approximations generated by the different MOEAs (i.e., we put, in one set, all the non-dominated solutions found and from this set we calculated the maximum and minimum for each objective function).

Also, we decided to use Δ_p to assess performance, because this indicator also measures both convergence to the true Pareto optimal set and distribution of the solutions along the Pareto front. The definition of this indicator is shown in eq. 4. It is important to keep in mind that the results of this indicator depend strongly on the reference set that is adopted. If we use reference sets with a considerable number of non-dominated solutions and these solutions are uniformly distributed, the results are reliable. In this case, we generated 10,000 random solutions using the WFG test suite.

B. Discussion of Results

In Table I, we compare Δ_p-MOEA with respect to MOEA/D using I_H and Δ_p. Also, we present the results of the statistical analysis using Wilcoxon's rank sum. In

(a), we can see that with respect to I_H, Δ_p-MOEA is better than MOEA/D in twenty-four cases because in these problems Δ_p-MOEA obtained a better result and the null hypothesis "medians are equal" can be rejected. And, in four cases (WFG1 with 3, 4, 5 and 6 objective functions) Δ_p-MOEA is outperformed by MOEA/D. Regarding Δ_p, see (b), Δ_p-MOEA outperforms MOEA/D in twenty-two cases, it is outperformed in five cases (WFG5 with 3, 4, 5 and 6 objective functions and WFG3 with 3 objective functions) and only in one case both MOEAs have a similar behavior (the null hypothesis cannot be rejected). Table III shows the time required by these two MOEAs to obtain the approximation of the Pareto set. In this case, MOEA/D is faster than Δ_p-MOEA. In the worst case, MOEA/D requires two seconds while Δ_p-MOEA requires ten seconds (MOEA/D is five times faster than our Δ_p-MOEA). However, we should consider that Δ_p-MOEA outperforms MOEA/D in most cases.

In Table II, we compare Δ_p-MOEA with respect to SMS-EMOA-HYPE using I_H and Δ_p. Also, we present the results of the statistical analysis using Wilcoxon's rank sum. In (a), we can see that with respect to I_H, Δ_p-MOEA outperforms SMS-EMOA-HYPE in twenty-one cases, it is outperformed by SMS-EMOA-HYPE only in four cases and in three cases both MOEAs have a similar behavior. In (b), we can see the results regarding Δ_p and, in this case, Δ_p-MOEA outperforms SMS-EMOA-HYPE in thirteen cases, it is outperformed in eleven cases and in four cases both MOEAs have a similar behavior. Table III shows that our Δ_p-MOEA is faster than SMS-EMOA-HYPE. In the worst case, our Δ_p-MOEA requires ten seconds to find the approximation of the Pareto optimal set while SMS-EMOA-HYPE requires forty-eight seconds in the worst case (our Δ_p-MOEA is 4.8 times faster than SMS-EMOA-HYPE).

Therefore, we can conclude that our Δ_p-MOEA is a good option to solve MOPs with either low or high dimensionality (in objective function space), if we consider both quality of the solutions and time required to solve the problems. This is because our proposed Δ_p-MOEA obtains better results than MOEA/D in most cases, although it is slower than MOEA/D. Also, our Δ_p-MOEA performed better than SMS-EMOA-HYPE in most cases while also having a lower computational cost.

VI. Conclusions and Future Work

We have proposed a new selection mechanism based on the Δ_p indicator. This selection mechanism has two special features: first, it creates the reference set using ϵ-dominace and second, it uses two different selection schemes (one based on GD and another based on IGD). Although these two selection schemes have as their main aim to optimize their respective indicators, they also consider other aspects such as the diversity in the population. Thus, they select solutions seeking a balance between the indicator value and the diversity in the population. It is important to emphasize that our selection mechanism does not require any extra information to generate the reference set: it builds it from the nondominated set known at each generation and it has a technique to automatically set the value of ϵ. Our selection mechanism was incorporated into a MOEA that uses the operators of NSGA-II to create individuals (crossover and mutation) giving rise to a new MOEA called "Δ_p - Multi-Objective Evolutionary Algorithm (Δ_p-MOEA)". Our preliminary results indicate that our Δ_p-MOEA is a good alternative to solve MOPs with low and high dimensionality (in objective function space) since it outperforms MOEAs such as MOEA/D and SMS-EMOA-HYPE in several problems and its computational cost is reasonably low (it is slower than MOEA/D but is faster than SMS-EMOA-HYPE). It is important to mention that we used two different indicators to measure the performance of the three MOEAs: I_H and Δ_p. And, with respect to these two indicators, our Δ_p-MOEA obtained better results than the other two MOEAs in most cases. In fact, our proposed approach turned out to be better than SMS-EMOA-HYPE with respect to I_H, in spite of the fact that the latter is based on such performance indicator.

As part of our future work, we want to study in detail why our Δ_p-MOEA has difficulties in problems in which many weakly dominated solutions are generated during the search process. Also, we want to analyze alternative techniques to set the value of ϵ because the adopted technique has two important disadvantages:

- With this technique at each generation of the evolutionary process we must start the process to set ϵ, i.e., we start by dividing the search space into two equal parts for each objective function, then in three parts, and then in four parts and so on until finding the value of ϵ that allows us to have, at most, one nondominated solution per hypercube. Therefore, this process can be very expensive.
- In this technique, we use the same value of ϵ for each objective function. However, this may not be the optimal configuration: Depending on the shape of the true Pareto front, it can happen that one objective function requires small values of ϵ, while another objective function requires bigger values of ϵ.

Finally, we want to study techniques to set optimal values for all parameters used by Δ_p-MOEA, e.g., we can use self-adaptive schemes (the value of each parameter is adjusted during the search process).

Acknowledgements

The first and second authors acknowledge support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The last author gratefully acknowledges support from CONACyT project no. 221551.

References

[1] M. Farina and P. Amato, "On the Optimal Solution Definition for Many-criteria Optimization Problems," in *Proceedings of the NAFIPS-FLINT International Conference'2002*, (Piscataway, New Jersey), pp. 233–238, IEEE Service Center, June 2002.

[2] 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*, vol. 10, pp. 477–506, October 2006.

$\vec{f}$	moead-pbi I_H	dp-moea I_H	$P(H)$
WFG1 (3)	**0.9699 (0.014)**	0.8756 (0.026)	0.000 (1)
WFG2 (3)	0.1838 (0.211)	**0.8018 (0.058)**	0.000 (1)
WFG3 (3)	0.5058 (0.026)	**0.6154 (0.007)**	0.000 (1)
WFG4 (3)	0.6177 (0.013)	**0.7205 (0.007)**	0.000 (1)
WFG5 (3)	0.4767 (0.011)	**0.5351 (0.004)**	0.000 (1)
WFG6 (3)	0.4554 (0.007)	**0.5568 (0.002)**	0.000 (1)
WFG7 (3)	0.5558 (0.062)	**0.6888 (0.012)**	0.000 (1)
WFG1 (4)	**1.0996 (0.040)**	0.9017 (0.028)	0.000 (1)
WFG2 (4)	0.0029 (0.016)	**0.6594 (0.188)**	0.000 (1)
WFG3 (4)	0.3054 (0.035)	**0.5734 (0.013)**	0.000 (1)
WFG4 (4)	0.6952 (0.026)	**0.9408 (0.017)**	0.000 (1)
WFG5 (4)	0.3753 (0.016)	**0.5672 (0.006)**	0.000 (1)
WFG6 (4)	0.2949 (0.016)	**0.5559 (0.010)**	0.000 (1)
WFG7 (4)	0.3639 (0.047)	**0.7663 (0.027)**	0.000 (1)
WFG1 (5)	**1.1775 (0.041)**	0.9496 (0.032)	0.000 (1)
WFG2 (5)	0.0270 (0.059)	**0.6864 (0.156)**	0.000 (1)
WFG3 (5)	0.1419 (0.041)	**0.5090 (0.039)**	0.000 (1)
WFG4 (5)	0.6986 (0.025)	**1.0901 (0.028)**	0.000 (1)
WFG5 (5)	0.2459 (0.015)	**0.5885 (0.015)**	0.000 (1)
WFG6 (5)	0.2476 (0.015)	**0.5409 (0.021)**	0.000 (1)
WFG7 (5)	0.3198 (0.023)	**0.8453 (0.035)**	0.000 (1)
WFG1 (6)	**1.1043 (0.013)**	0.9914 (0.032)	0.000 (1)
WFG2 (6)	0.0056 (0.026)	**0.6061 (0.285)**	0.000 (1)
WFG3 (6)	0.1244 (0.053)	**0.5313 (0.057)**	0.000 (1)
WFG4 (6)	0.6232 (0.029)	**1.1847 (0.035)**	0.000 (1)
WFG5 (6)	0.1548 (0.018)	**0.6130 (0.015)**	0.000 (1)
WFG6 (6)	0.2326 (0.020)	**0.5352 (0.039)**	0.000 (1)
WFG7 (6)	0.2884 (0.022)	**0.8203 (0.049)**	0.000 (1)

(a)

$\vec{f}$	moead-pbi Δ_p	dp-moea Δ_p	$P(H)$
WFG1 (3)	2.0151 (0.041)	**2.0126 (0.148)**	0.750 (0)
WFG2 (3)	1.6571 (0.051)	**1.4322 (0.133)**	0.000 (1)
WFG3 (3)	**1.9938 (0.008)**	2.0050 (0.020)	0.048 (1)
WFG4 (3)	0.5331 (0.010)	**0.2805 (0.023)**	0.000 (1)
WFG5 (3)	**0.7405 (0.003)**	0.8021 (0.016)	0.000 (1)
WFG6 (3)	1.9609 (0.009)	**1.9388 (0.015)**	0.000 (1)
WFG7 (3)	1.5097 (0.083)	**1.4060 (0.006)**	0.000 (1)
WFG1 (4)	2.9841 (0.083)	**2.9337 (0.085)**	0.009 (1)
WFG2 (4)	2.1579 (0.005)	**1.7896 (0.148)**	0.000 (1)
WFG3 (4)	2.5695 (0.050)	**2.4824 (0.016)**	0.000 (1)
WFG4 (4)	1.3731 (0.021)	**0.9016 (0.057)**	0.000 (1)
WFG5 (4)	**1.2136 (0.005)**	1.3505 (0.016)	0.000 (1)
WFG6 (4)	2.7726 (0.031)	**2.6656 (0.014)**	0.000 (1)
WFG7 (4)	3.7478 (0.266)	**1.7887 (0.033)**	0.000 (1)
WFG1 (5)	4.2674 (0.129)	**3.6219 (0.065)**	0.000 (1)
WFG2 (5)	2.6711 (0.111)	**2.3936 (0.257)**	0.000 (1)
WFG3 (5)	3.1010 (0.065)	**3.0203 (0.023)**	0.000 (1)
WFG4 (5)	3.4391 (0.621)	**1.6450 (0.091)**	0.000 (1)
WFG5 (5)	**1.6984 (0.011)**	1.7792 (0.033)	0.000 (1)
WFG6 (5)	3.7233 (0.050)	**3.5480 (0.017)**	0.000 (1)
WFG7 (5)	6.1148 (0.132)	**2.3596 (0.107)**	0.000 (1)
WFG1 (6)	5.4592 (0.195)	**4.3805 (0.085)**	0.000 (1)
WFG2 (6)	3.1833 (0.134)	**2.7790 (0.314)**	0.000 (1)
WFG3 (6)	3.7707 (0.082)	**3.5661 (0.052)**	0.000 (1)
WFG4 (6)	6.0918 (0.519)	**2.7645 (0.146)**	0.000 (1)
WFG5 (6)	**2.2548 (0.012)**	2.3420 (0.065)	0.000 (1)
WFG6 (6)	5.0138 (0.212)	**4.5607 (0.031)**	0.000 (1)
WFG7 (6)	7.7530 (0.113)	**3.0803 (0.172)**	0.000 (1)

(b)

Table I. RESULTS OBTAINED IN THE WFG TEST PROBLEMS. WE COMPARE OUR Δ_p-MOEA WITH RESPECT TO MOEA/D, USING THE HYPERVOLUME INDICATOR (I_H) AND THE Δ_p INDICATOR. WE SHOW AVERAGE VALUES OVER 30 INDEPENDENT RUNS. THE VALUES IN PARENTHESES CORRESPOND TO THE STANDARD DEVIATIONS. THE THIRD COLUMN OF EACH TABLE SHOWS THE RESULTS OF THE STATISTICAL ANALYSIS APPLIED TO OUR EXPERIMENTS USING WILCOXON'S RANK SUM. P IS THE PROBABILITY OF OBSERVING THE GIVEN RESULT (THE NULL HYPOTHESIS IS TRUE). SMALL VALUES OF P CAST DOUBT ON THE VALIDITY OF THE NULL HYPOTHESIS. $H = 0$ INDICATES THAT THE NULL HYPOTHESIS ("MEDIANS ARE EQUAL") CANNOT BE REJECTED AT THE 5% LEVEL. $H = 1$ INDICATES THAT THE NULL HYPOTHESIS CAN BE REJECTED AT THE 5% LEVEL.

$\vec{f}$	sms-emoa-hype I_H	dp-moea I_H	$P(H)$
WFG1 (3)	0.8200 (0.036)	**0.8756 (0.026)**	0.000 (1)
WFG2 (3)	0.6334 (0.070)	**0.8018 (0.058)**	0.000 (1)
WFG3 (3)	0.5994 (0.007)	**0.6154 (0.007)**	0.000 (1)
WFG4 (3)	0.7115 (0.007)	**0.7205 (0.007)**	0.000 (1)
WFG5 (3)	**0.5397 (0.002)**	0.5351 (0.004)	0.000 (1)
WFG6 (3)	0.5514 (0.004)	**0.5568 (0.002)**	0.000 (1)
WFG7 (3)	0.5350 (0.034)	**0.6888 (0.012)**	0.000 (1)
WFG1 (4)	0.8828 (0.040)	**0.9017 (0.028)**	0.176 (0)
WFG2 (4)	0.3189 (0.215)	**0.6594 (0.188)**	0.000 (1)
WFG3 (4)	0.5270 (0.015)	**0.5734 (0.013)**	0.000 (1)
WFG4 (4)	0.9285 (0.014)	**0.9408 (0.017)**	0.004 (1)
WFG5 (4)	0.5535 (0.007)	**0.5672 (0.006)**	0.000 (1)
WFG6 (4)	**0.5630 (0.011)**	0.5559 (0.010)	0.009 (1)
WFG7 (4)	0.4332 (0.031)	**0.7663 (0.027)**	0.000 (1)
WFG1 (5)	**0.9836 (0.045)**	0.9496 (0.032)	0.001 (1)
WFG2 (5)	0.4739 (0.238)	**0.6864 (0.156)**	0.000 (1)
WFG3 (5)	0.4291 (0.031)	**0.5090 (0.039)**	0.000 (1)
WFG4 (5)	**1.0968 (0.025)**	1.0901 (0.028)	0.411 (0)
WFG5 (5)	0.5728 (0.010)	**0.5885 (0.015)**	0.000 (1)
WFG6 (5)	0.4995 (0.036)	**0.5409 (0.021)**	0.000 (1)
WFG7 (5)	0.3933 (0.023)	**0.8453 (0.035)**	0.000 (1)
WFG1 (6)	**1.0185 (0.045)**	0.9914 (0.032)	0.007 (1)
WFG2 (6)	0.3177 (0.222)	**0.6061 (0.285)**	0.000 (1)
WFG3 (6)	0.4053 (0.053)	**0.5313 (0.057)**	0.000 (1)
WFG4 (6)	1.1670 (0.038)	**1.1847 (0.035)**	0.233 (0)
WFG5 (6)	0.5286 (0.016)	**0.6130 (0.015)**	0.000 (1)
WFG6 (6)	0.4590 (0.046)	**0.5352 (0.039)**	0.000 (1)
WFG7 (6)	0.3909 (0.018)	**0.8203 (0.049)**	0.000 (1)

$\vec{f}$	sms-emoa-hype Δ_p	dp-moea Δ_p	$P(H)$
WFG1 (3)	2.2917 (0.060)	**2.0126 (0.148)**	0.000 (1)
WFG2 (3)	1.4653 (0.150)	**1.4322 (0.133)**	0.009 (1)
WFG3 (3)	**1.9659 (0.004)**	2.0050 (0.020)	0.000 (1)
WFG4 (3)	0.3429 (0.029)	**0.2805 (0.023)**	0.000 (1)
WFG5 (3)	**0.7441 (0.003)**	0.8021 (0.016)	0.000 (1)
WFG6 (3)	**1.9277 (0.016)**	1.9388 (0.015)	0.006 (1)
WFG7 (3)	1.5494 (0.109)	**1.4060 (0.006)**	0.000 (1)
WFG1 (4)	3.1491 (0.161)	**2.9337 (0.085)**	0.000 (1)
WFG2 (4)	1.8674 (0.195)	**1.7896 (0.148)**	0.641 (0)
WFG3 (4)	**2.4514 (0.004)**	2.4824 (0.016)	0.000 (1)
WFG4 (4)	1.0911 (0.066)	**0.9016 (0.057)**	0.000 (1)
WFG5 (4)	**1.2411 (0.009)**	1.3505 (0.016)	0.000 (1)
WFG6 (4)	**2.6539 (0.018)**	2.6656 (0.014)	0.001 (1)
WFG7 (4)	2.7103 (0.293)	**1.7887 (0.033)**	0.000 (1)
WFG1 (5)	3.9967 (0.256)	**3.6219 (0.065)**	0.000 (1)
WFG2 (5)	2.4102 (0.259)	**2.3936 (0.257)**	0.529 (0)
WFG3 (5)	**2.9663 (0.003)**	3.0203 (0.023)	0.000 (1)
WFG4 (5)	2.0859 (0.115)	**1.6450 (0.091)**	0.000 (1)
WFG5 (5)	**1.7061 (0.014)**	1.7792 (0.033)	0.000 (1)
WFG6 (5)	**3.5462 (0.022)**	3.5480 (0.017)	0.270 (0)
WFG7 (5)	4.2434 (0.315)	**2.3596 (0.107)**	0.000 (1)
WFG1 (6)	4.9748 (0.364)	**4.3805 (0.085)**	0.000 (1)
WFG2 (6)	2.7995 (0.317)	**2.7790 (0.314)**	0.501 (0)
WFG3 (6)	**3.4659 (0.006)**	3.5661 (0.052)	0.000 (1)
WFG4 (6)	3.6893 (0.360)	**2.7645 (0.146)**	0.000 (1)
WFG5 (6)	**2.2448 (0.022)**	2.3420 (0.065)	0.000 (1)
WFG6 (6)	**4.5361 (0.030)**	4.5607 (0.031)	0.001 (1)
WFG7 (6)	5.4801 (0.316)	**3.0803 (0.172)**	0.000 (1)

Table II. RESULTS OBTAINED IN THE WFG TEST PROBLEMS. WE COMPARE OUR Δ_p-MOEA WITH RESPECT TO SMS-EMOA-HYPE, USING THE HYPERVOLUME INDICATOR (I_H) AND THE Δ_p INDICATOR. WE SHOW AVERAGE VALUES OVER 30 INDEPENDENT RUNS. THE VALUES IN PARENTHESES CORRESPOND TO THE STANDARD DEVIATIONS. THE THIRD COLUMN OF EACH TABLE SHOWS THE RESULTS OF THE STATISTICAL ANALYSIS APPLIED TO OUR EXPERIMENTS USING WILCOXON'S RANK SUM. P IS THE PROBABILITY OF OBSERVING THE GIVEN RESULT (THE NULL HYPOTHESIS IS TRUE). SMALL VALUES OF P CAST DOUBT ON THE VALIDITY OF THE NULL HYPOTHESIS. $H = 0$ INDICATES THAT THE NULL HYPOTHESIS ("MEDIANS ARE EQUAL") CANNOT BE REJECTED AT THE 5% LEVEL. $H = 1$ INDICATES THAT THE NULL HYPOTHESIS CAN BE REJECTED AT THE 5% LEVEL.

$\vec{f}$	moead-pbi time	sms-emoa-hype time	dp-moea time
WFG1 (3)	**1.5000 (0.500)**	29.3180 (3.150)	6.7983 (1.310)
WFG2 (3)	**1.1667 (0.373)**	21.6913 (2.086)	6.6697 (1.212)
WFG3 (3)	**1.2333 (0.423)**	29.4300 (3.409)	9.0743 (0.582)
WFG4 (3)	**1.3000 (0.458)**	25.2457 (2.644)	5.4883 (0.859)
WFG5 (3)	**1.2667 (0.442)**	31.2997 (3.408)	8.9833 (0.569)
WFG6 (3)	**1.2000 (0.400)**	32.2827 (4.233)	8.7317 (0.695)
WFG7 (3)	**1.6333 (0.482)**	30.7810 (3.713)	5.7933 (0.699)
WFG1 (4)	**1.5333 (0.499)**	32.7003 (1.907)	7.3240 (1.346)
WFG2 (4)	**1.2667 (0.442)**	24.6957 (2.153)	7.4783 (1.460)
WFG3 (4)	**1.3333 (0.471)**	33.0970 (2.243)	4.0230 (0.848)
WFG4 (4)	**1.3667 (0.482)**	28.3970 (2.213)	6.3813 (1.316)
WFG5 (4)	**1.3000 (0.458)**	38.7537 (2.166)	8.9097 (2.072)
WFG6 (4)	**1.2667 (0.442)**	38.7110 (2.052)	6.2967 (1.159)
WFG7 (4)	**1.7000 (0.458)**	36.4420 (2.422)	7.6110 (1.447)
WFG1 (5)	**1.6333 (0.482)**	38.8440 (2.147)	7.0817 (1.857)
WFG2 (5)	**1.3000 (0.458)**	30.6240 (1.637)	6.1977 (1.457)
WFG3 (5)	**1.3667 (0.482)**	40.2863 (2.344)	4.7840 (0.429)
WFG4 (5)	**1.4667 (0.499)**	35.3357 (1.935)	6.9950 (1.916)
WFG5 (5)	**1.4333 (0.496)**	47.2197 (2.837)	6.3120 (0.546)
WFG6 (5)	**1.3000 (0.458)**	46.6780 (3.141)	6.7110 (1.434)
WFG7 (5)	**1.9667 (0.180)**	43.8047 (2.095)	8.0043 (2.185)
WFG1 (6)	**1.7333 (0.442)**	38.1870 (4.864)	5.6143 (0.557)
WFG2 (6)	**1.3667 (0.482)**	36.7027 (2.644)	4.7930 (0.738)
WFG3 (6)	**1.4333 (0.496)**	39.4787 (6.629)	5.3673 (0.709)
WFG4 (6)	**1.5667 (0.496)**	41.1373 (3.706)	5.6143 (0.512)
WFG5 (6)	**1.5000 (0.500)**	38.2700 (7.106)	5.4603 (0.617)
WFG6 (6)	**1.3333 (0.471)**	37.7870 (6.954)	5.2073 (0.685)
WFG7 (6)	**2.1667 (0.373)**	37.2300 (6.830)	5.8410 (0.762)

Table III. TIME REQUIRED (IN SECONDS) BY MOEA/D, SMS-EMOA-HYPE AND OUR PROPOSED Δ_p-MOEA FOR THE TEST PROBLEMS ADOPTED. ALL ALGORITHMS WERE COMPILED USING THE GNU C COMPILER AND THEY WERE EXECUTED ON A COMPUTER WITH A 2.66GHZ PROCESSOR AND 4GB IN RAM.

[3] B. Naujoks, N. Beume, and M. Emmerich, "Multi-objective Optimization using S-metric Selection: Application to three-dimensional Solution Spaces," in *2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, vol. 2, (Edinburgh, Scotland), pp. 1282–1289, IEEE Service Center, September 2005.

[4] O. Schutze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the averaged hausdorff distance as a performance measure in evolutionary multiobjective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 16, pp. 504–522, Aug 2012.

[5] D. A. V. Veldhuizen, *Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[6] C. A. Coello Coello and N. Cruz Cortés, "Solving Multiobjective Optimization Problems using an Artificial Immune System," *Genetic Programming and Evolvable Machines*, vol. 6, pp. 163–190, June 2005.

[7] C. A. Rodríguez Villalobos and C. A. Coello Coello, "A new multiobjective evolutionary algorithm based on a performance assessment indicator," in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation*, GECCO '12, (New York, NY, USA), pp. 505–512, ACM, 2012.

[8] G. Rudolph, H. Trautmann, S. Sengupta, and O. Schütze, "Evenly spaced pareto front approximations for tricriteria problems based on triangulation," in *Evolutionary Multi-Criterion Optimization* (R. Purshouse, P. Fleming, C. Fonseca, S. Greco, and J. Shaw, eds.), vol. 7811 of *Lecture Notes in Computer Science*, pp. 443–458, Springer Berlin Heidelberg, 2013.

[9] C. Dominguez-Medina, G. Rudolph, O. Schutze, and H. Trautmann, "Evenly spaced pareto fronts of quad-objective problems using psa partitioning technique," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pp. 3190–3197, June 2013.

[10] G. Rudolph, O. Schütze, C. Grimme, and H. Trautmann, "An aspiration set emoa based on averaged hausdorff distances," in *Learning and Intelligent Optimization* (P. M. Pardalos, M. G. Resende, C. Vogiatzis, and J. L. Walteros, eds.), Lecture Notes in Computer Science, pp. 153–156, Springer International Publishing, 2014.

[11] S. Zapotecas Martínez, V. Sosa Hernández, H. Aguirre, K. Tanaka, and C. Coello Coello, "Using a family of curves to approximate the pareto front of a multi-objective optimization problem," in *Parallel Problem Solving from Nature – PPSN XIII* (T. Bartz-Beielstein, J. Branke, B. Filipič, and J. Smith, eds.), vol. 8672 of *Lecture Notes in Computer Science*, pp. 682–691, Springer International Publishing, 2014.

[12] A. Menchaca-Mendez and C. A. Coello Coello, "GDE-MOEA : A New MOEA based on the Maximin Fitness Function and Euclidean Distances between Solutions," in *2015 IEEE Congress on Evolutionary Computation (CEC'2015)*, (Sendai, Japan), IEEE Press, 25-28 May 2015.

[13] A. Menchaca-Mendez and C. A. Coello Coello, "GD-MOEA: A New Multi-Objective Evolutionary Algorithm Based on the Generational Distance Indicator," in *Evolutionary Multi-Criterion Optimization, 8th International Conference, EMO 2015* (A. Gaspar-Cunha, C. H. Antunes, and C. Coello Coello, eds.), pp. 156–170, Guimarães, Portugal: Springer. Lecture Notes in Computer Science Vol. 9018, March 29 - April 1 2015.

[14] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 182–197, April 2002.

[15] J. Bader and E. Zitzler, "HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, Spring, 2011.

[16] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM J. on Optimization*, vol. 8, pp. 631–657, Mar. 1998.