MD-MOEA : A New MOEA based on the Maximin Fitness Function and Euclidean Distances between Solutions

Adriana Menchaca-Mendez
CINVESTAV-IPN,
Departamento de Computación
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO
email: adriana.menchacamendez@gmail.com

Carlos A. Coello Coello
CINVESTAV-IPN,
Departamento de Computación
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO
email: ccoello@cs.cinvestav.mx

***Abstract*—In this paper, we propose a new selection mechanism based on the maximin fitness function and a technique based on Euclidean distances between solutions to improve the diversity of the population in objective function space. Our new selection mechanism is incorporated into a multi-objective evolutionary algorithm (MOEA) which uses the operators of NSGA-II (crossover and mutation) to generate new individuals, giving rise to the so-called "Maximin-Distances Multi-Objective Evolutionary Algorithm (MD-MOEA)". Our MD-MOEA is validated using standard test functions taken from the specialized literature, having three to six objective functions. MD-MOEA is compared with respect to MC-MOEA (which is based on the maximin fitness function and a clustering technique), 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 based on the use of an approximation of the hypervolume indicator). Our preliminary results indicate that our MD-MOEA is a good alternative to solve multi-objective optimization problems having both low dimensionality and high dimensionality in objective function space because it obtains better results than MC-MOEA and MOEA/D in most cases and it is competitive with respect to SMS-EMOA-HYPE (in fact, it outperforms SMS-EMOA-HYPE in problems of high dimensionality) but at a much lower computational cost.**

I. Introduction

In the real word, there are many optimization problems which involve multiple objective functions which must be satisfied simultaneously. They are called *multiobjective optimization problems (MOPs)* and usually their objectives are in conflict with each other. In MOPs, the notion of optimality refers to the best possible trade-offs among the objectives. Consequently, these MOPs have several solutions (the so-called *Pareto optimal set* whose image is called the *Pareto front*). The use of evolutionary algorithms for solving MOPs has become very popular and they are generically called Multi-Objective Evolutionary Algorithms (MOEAs). MOEAs have two main goals [1]: (i) to find solutions that are, as close as possible, to the true Pareto front and, (ii) to produce solutions that are spread along the Pareto front as uniformly as possible. Based on their selection mechanism, MOEAs can be classified in two groups: (i) those that incorporate the concept of Pareto optimality, and (ii) those that do not use Pareto dominance to select individuals. Since Pareto-based MOEAs have several limitations, mainly when solving MOPs with many objective functions,[1] MOEAs of type (ii) have become relatively popular in recent years.

SMS-EMOA [3], [4] and MOEA/D [5] are two well-known MOEAs of type (ii). However, both have some disadvantages. SMS-EMOA uses Pareto ranking (as in NSGA-II) as well as the contribution to the hypervolume indicator in order to decide which individual will be removed. It is worth noticing, however, that when we only obtain one front after performing Pareto ranking, SMS-EMOA needs to calculate the contribution of all individuals in the population in order to decide which will be removed. Since calculating the contribution to the hypervolume is an NP-hard problem [6], this MOEA is not practical when we want to solve MOPs with many objective functions (more than six). There are several proposals to address this problem. For example: to approximate the hypervolume or the calculation of the contribution to the hypervolume [7], [8], [9], [10]. However, MOEAs that approximate the hypervolume lose quality in their solutions in a significant manner. Other authors have proposed an efficient algorithm to calculate the contributions to the hypervolume but only for low dimensionality (two and three objective functions) [11]. It is also worth mentioning the new competition scheme for selection mechanisms based on the hypervolume indicator that was proposed in [12]. With this scheme, the authors were able to have a significant reduction of the running time of the original SMS-EMOA without losing quality in the solutions that it produces. However, this selection mechanism also needs to calculate contributions to the hypervolume and, therefore, it is still impractical for MOPs with many objective functions.

Regarding MOEA/D, it decomposes the MOP into N scalar optimization subproblems and then it solves these subproblems simultaneously using an evolutionary algorithm. At each generation, the best solution found so far for each subproblem survives. Although this MOEA has a low computational cost and is able to find an approximation of the Pareto front with a good distribution (this depends of the approach used to decompose the MOP), MOEA/D always needs to generate a set of well-distributed convex weights and perhaps this is its most important disadvantage because this task is difficult in several cases. There are some proposals

[1]The number of nondominated solutions grows exponentially as we increase the number of objective functions, and this rapidly dilutes the selection pressure of a MOEA [2].

to generate these weights [13], [14]. However, none of these techniques ensures obtaining a uniformly distributed set of weights for high dimensionality and they also have some disadvantages. For example, in [13], when we increase the number of objective functions, the number of weights grows according to $\binom{n+p-1}{p}$, where n is the number of objective functions and $p = \frac{1}{\delta}$ (δ is the stepsize). Then, if we use $\delta = 0.1$, this technique generates 66, 286, 1001, 3003, 8008 and 19448 weights for 3, 4, 5, 6, 7 and 8 objective functions, respectively. Since using population sizes as large as these values suggest is not practical, it is necessary to use another technique (for example, clustering) to select a subset of these weights for running the algorithm.

Another approach, called MC-MOEA [15], [16] which is based on the maximin fitness function [17], [18], [19], was recently proposed. This is clearly a MOEA of type (ii). The maximin fitness function has interesting properties. For example, based on the fitness value of an individual, we can know if an individual is non-dominated or dominated. Also, its fitness value is a metric of the distance to the non-dominated front. Additionally, the fitness value of an individual is penalized if it is clustered together with other individuals. Another important thing is that the complexity of calculating the maximin fitness function is linear with respect to the number of objective functions, which makes this approach a good choice for dealing with many-objective optimization problems. In [16], the authors mention some disadvantages of the maximin fitness function and they make some proposals to address them. One of these disadvantages is that although the maximin fitness function penalizes individuals that are clustered together, the solutions generated by MOEAs based on this sort of approach don't have a good distribution along the Pareto front. At this point, it is important to mention that improving the diversity of the solutions generated by a MOEA implies not to affect its convergence properties. In this paper, we address this issue by proposing a technique to correct the possible errors generated (with respect to the diversity) when we use the maximin fitness function to select the individuals which will be part of the next generation. The proposed technique is based only on Euclidean distances between solutions in objective function space and it has linear complexity with respect to the number of objective functions. Thus, both our final selection mechanism based on the maximin fitness function and our technique to improve diversity are linear with respect to the number of objectives.

The remainder of this paper is organized as follows. Section II states the problem of our interest. Section III describes the maximin fitness function. The previous related work is discussed in Section IV. Our proposal is discussed in Section V. Our experimental validation and the results obtained are shown in Section VI. Finally, we provide our conclusions and some possible paths for future work in Section VII.

II. Problem Statement

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

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

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 any $\vec{x} \in \Omega$ 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: $\mathcal{PF}^* = \{\vec{f}(\vec{x}) \mid \vec{x} \in \mathcal{P}^*\}$.

III. Maximin Fitness Function

The maximin fitness function was proposed by Balling [17] and it works as follows. Let's consider a MOP with K objective functions and an evolutionary algorithm whose population size is P. Let f_k^i be the normalized value of the kth objective for the ith individual in a particular generation. Assuming minimization problems, we have that the jth individual weakly dominates the ith individual if:

$$min_k(f_k^i - f_k^j) \geq 0 \tag{2}$$

The ith individual, in a particular generation, will be weakly dominated by another individual, in the generation, if:

$$max_{j \neq i}(min_k(f_k^i - f_k^j)) \geq 0 \tag{3}$$

Then, the maximin fitness function of individual i is defined as:

$$fitness^i = max_{j \neq i}(min_k(f_k^i - f_k^j)) \tag{4}$$

where the min is taken over all the objective functions, and the max is taken over all the individuals in the population, except for the same individual i. From eq. (4), we can say the following:

1) Any individual whose maximin fitness is greater than zero is a dominated individual,
2) Any individual whose maximin fitness is less than zero is a non-dominated individual.
3) Finally, any individual whose maximin fitness is equal to zero is a weakly-dominated individual.

Some interesting properties of the maximin fitness function are the following:

1) The maximin fitness function penalizes clustering of non-dominated individuals. See Figure 1(b).
2) The maximin fitness of dominated individuals is a metric of the distance to the non-dominated front. See Figure 1(c).

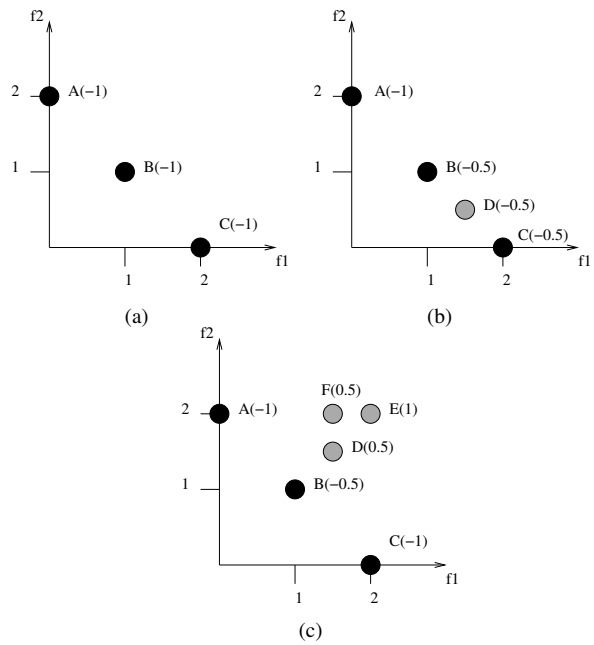

Fig. 1. Properties of the maximin fitness function. In (a) all individuals have the same fitness because they are non-dominated to each other and they are well distributed. In (b), we can see that if we incorporate individual D, individuals B, C and also D are penalized because they are close from each other. In (c), we can see that the fitness of individuals D, E and F is controlled by the non-dominated individual B, and their fitness is a metric of the distance to the individual B. Also, we can see that the fitness of individual B is affected by the dominated individual D because they are close.

3) The max function in the maximin fitness of a dominated individual is always controlled by a non-dominated individual and is indifferent to clustering. The max function in the maximin fitness of a non-dominated individual may be controlled by a dominated or a non-dominated individual. See Figure 1(c).

The author of the maximin fitness function proposes in [18] the following modified maximin fitness function:

$$fitness^i = max_{j \neq i, j \in \mathbb{ND}}(min_k(f_k^i - f_k^j)) \qquad (5)$$

where $\mathbb{ND}$ is the set of non-dominated individuals. Using eq. (5) to assign the fitness of each individual, we guarantee that the fitness of a non-dominated individual is controlled only by non-dominated individuals and then we only penalize clustering between non-dominated individuals. For example, if we use the modified maximin fitness function in Figure 1(c), individual B would not be penalized and it would retain a fitness equal to -1.

IV. Previous Related Work

The maximin fitness function has been incorporated in genetic algorithms [18], [19], [16], particle swarm optimizers [20], [21], ant colony optimizers [22] and differential evolution [15]. In most of these papers, only low dimensionality MOPs were considered (i.e., MOPs with 2 objectives) and no extra diversity mechanism was adopted.

In [15], two important disadvantages of the maximin fitness function were identified. The main disadvantage arises from the following question: Is it better to prefer weakly dominated individuals than dominated individuals? The answer was that it is not good to prefer weakly dominated individuals or individuals which are close to being weakly dominated (even if they are weakly dominated by any dominated individual). For example, if we assign the fitness of each individual using the maximin fitness function, and then we sort the individuals according to their fitness values, we can obtain many (even only) weakly Pareto points. In order to address this problem, the following constraint was proposed in [15]: *Any individual that we want to select must not be similar (in objective function space) to another (selected) individual*.

The second disadvantage has to do with the poor diversity obtained in objective function space when we use the maximin fitness to select individuals. In [15], the authors showed that the maximin fitness function has difficulties in some cases. For example, in Figure 1(b), individuals B, C and D have the same maximin fitness. Then, we cannot know which of these three individuals is the best choice to form part of the next generation. In order to address this problem, in [15] it was proposed to use a clustering technique to correct the possible errors produced in the selection process when using the maximin fitness as follows: If we want to select S individuals from a population of size P, then, we choose the best S individuals with respect to their maximin fitness value, and we use them as centers of the clusters. Then, we proceed to place each individual in the nearest cluster. Finally, for each of the resulting clusters, we recompute the center and we choose the individual closest to it. This technique is only effective in cases when more than S individuals are non-dominated. The clustering technique does not iterate many times to improve the distribution of the centers, because we choose the initial centers according to the maximin fitness value and we only want to do a small correction based on the idea on which the maximin fitness function penalizes clustering. Because of this, in [16] the authors studied the effect of using different selection operators based on either the original maximin fitness function or the modified maximin fitness function and the above clustering technique.

V. Our Proposed Selection Mechanism

We based our selection mechanism on the mechanisms proposed in [15], [16] but instead of using a clustering technique to improve de diversity in the population (objective function space) we propose here the use of a technique based on Euclidean distances between solutions. We choose this type of distance because we want that the solutions are uniformly distributed in objective function space. Also, it is important to mention that if all the objectives are equally important, we need to calculate the Euclidean distance on the normalized values of the objective functions. Our selection mechanism works as follows: Let's assume that we want to select S individuals from a population of size P. First, we assign fitness to each individual using the modified maximin fitness function (see equation (5)). Then, we proceed to select individuals according to their fitness, verifying similarity between selected individuals (see Algorithm 1, lines 1 to 20). If we already selected the S individuals but there are still non-dominated individuals who have not participated in the selection process, then we proceed to use the technique

proposed in the next Subsection to improve diversity (see Algorithm 1, lines 22 to 41). The process to verify similarity is shown in Algorithm 2, where min_dif is the minimum difference allowed between solutions with respect to all objective functions and K is the number of objective functions.

A. Improving Diversity

Our aim now will be to maximize the minimum distance between solutions in objective function space. For this sake, we will do the following: Let $\mathbb{X}$ be the population from which we want to select S individuals and let $\mathbb{S}$ be the set of already selected individuals. For each nondominated individual $\mathbb{X}[i]$ who has not participated in the selection process (because its fitness is low), we obtain its nearest neighbor from $\mathbb{S}$ ($\mathbb{S}[X]$) and we choose a random individual[2] from $\mathbb{S}$ ($\mathbb{S}[R]$, such that $X \neq R$). Then, $\mathbb{X}[i]$ will compete with $\mathbb{S}[R]$ and $\mathbb{S}[X]$ to survive. We use $\mathbb{S}[X]$ with the idea of improving the diversity locally: If we move $\mathbb{S}[X]$ to $\mathbb{X}[i]$, do we increase the distance with respect to its nearest neighbor in $\mathbb{S}$? And, we use $\mathbb{S}[R]$ because we consider the scenario in which the solution $\mathbb{S}[X]$ is in an unexplored region and, therefore, it is not a good idea to delete $\mathbb{S}[X]$ or $\mathbb{X}[i]$. Therefore, we propose that first, $\mathbb{X}[i]$ competes with the randomly chosen solution $\mathbb{S}[R]$: If the Euclidean distance from $\mathbb{X}[i]$ to its nearest neighbor in $\mathbb{S}$ is greater than the Euclidean distance from $\mathbb{S}[R]$ to its nearest neighbor in $\mathbb{S}$, we replace $\mathbb{S}[R]$ with $\mathbb{X}[i]$. If $\mathbb{X}[i]$ loses the competition, then $\mathbb{X}[i]$ competes with its nearest neighbor to survive. If the Euclidean distance from $\mathbb{X}[i]$ to its nearest neighbor in $\mathbb{S}$ (without considering $\mathbb{S}[X]$) is greater than the Euclidean distance from $\mathbb{S}[X]$ to its nearest neighbor in $\mathbb{S}$, then we replace $\mathbb{S}[X]$ with $\mathbb{X}[i]$.

In Figure 2(a), we can see an example of selection using only the maximin fitness function; the black points are the selected individuals. Figures 2(b,c) show the selection process using the maximin fitness function and the clustering technique proposed in [15]. In (b), we can see the clusters constructed and, in (c), we can see the selected individuals. Figures 2(d,e) show the selection process using the maximin fitness function and our technique based on Euclidean distances. Since individuals C and D are not considered in (a), in (d), C competes with A and B, and C replaces B. In (e), D competes with A and C, and D replaces C. As we can observe in Figure 2, our selection mechanism obtains better results than the other two.

B. Maximin-Distances Multi-Objective Evolutionary Algorithm (MD-MOEA)

In order to validate our selection mechanism, we designed a multi-objective evolutionary algorithm that uses the operators of NSGA-II (crossover and mutation) to create new individuals. This is because our main aim is to validate the effect of our proposed selection mechanism comparing it with respect to other three selection mechanisms based on different techniques: (1) the first is based on the same maximin fitness function, (2) the second is based on the approximation of the hypervolume indicator and, (3) the

[2]We assume that the probability of choosing an individual in a crowded region is higher than the probability of choosing an individual in an unexplored region.

Algorithm 1: Maximin selection

Input : $\mathbb{X}$ (Population), S (number of individuals to choose $S < \|\mathbb{X}\|$).
Output: $\mathbb{S}$ (Selected individuals).

1 AssignFitness($\mathbb{X}$);
2 $numNonDom \leftarrow$ Number of nondominated solutions in $\mathbb{X}$;
/*Sorting with respect to the maximin fitness */
3 Sort($\mathbb{X}$);
4 $s \leftarrow 1$, $i \leftarrow 1$, $\mathbb{S} \leftarrow \emptyset$;
/*Fill up the new population with the best copies according to the maximin fitness, verifying that there is not a similar one */
5 **while** $s \leq S$ *AND* $i \leq \|\mathbb{X}\|$ **do**
6 **if** $\mathbb{X}[i]$ *is not similar to any individual in* $\mathbb{S}$ **then**
/*Select individual i */
7 $\mathbb{S} \leftarrow \mathbb{S} \cup \mathbb{X}[i]$;
8 $s \leftarrow s + 1$;
9 **end**
10 $i \leftarrow i + 1$;
11 **end**
12 **if** $s \leq S$ **then**
/*Choose the remaining individuals considering only the maximin fitness */
13 $i \leftarrow 1$;
14 **while** $s \leq S$ **do**
15 **if** $\mathbb{X}[i]$ *has not been selected* **then**
16 $\mathbb{S} \leftarrow \mathbb{S} \cup \mathbb{X}[i]$;
17 $s \leftarrow s + 1$;
18 **end**
19 $i \leftarrow i + 1$;
20 **end**
21 **else**
/*Improve diversity according to the Euclidean distances between solutions. */
22 **while** $i < numNonDom$ **do**
23 **if** $\mathbb{X}[i]$ *is not similar to any individual in* $\mathbb{S}$ **then**
24 $X \leftarrow$ Index of nearest neighbor to $\mathbb{X}[i]$ in $\mathbb{S}$;
25 $dItoX \leftarrow$ Distance from $\mathbb{X}[i]$ to $\mathbb{S}[X]$;
26 $R \leftarrow$ Obtain a random index between 1 and S such that $R \neq X$;
27 $Y \leftarrow$ Index of nearest neighbor to $\mathbb{S}[R]$ in $\mathbb{S}$;
28 $dRtoY \leftarrow$ Distance from $\mathbb{S}[R]$ to $\mathbb{S}[Y]$;
29 **if** $dItoX > dRtoY$ **then**
30 Replace $\mathbb{S}[R]$ with $\mathbb{X}[i]$;
31 **else**
32 $Z \leftarrow$ Index of nearest neighbor to $\mathbb{X}[X]$ in $\mathbb{S}$;
33 $dXtoZ \leftarrow$ Distance from $\mathbb{S}[X]$ to $\mathbb{S}[Z]$;
34 $W \leftarrow$ Index of nearest neighbor to $\mathbb{X}[i]$, without regard to X, in $\mathbb{S}$;
35 $dItoW \leftarrow$ Distance from $\mathbb{X}[i]$ to $\mathbb{S}[W]$;
36 **if** $dItoW > dXtoZ$ **then**
37 Replace $\mathbb{S}[X]$ with $\mathbb{X}[i]$;
38 **end**
39 **end**
40 **end**
41 **end**
42 **end**
43 **return** $\mathbb{S}$;

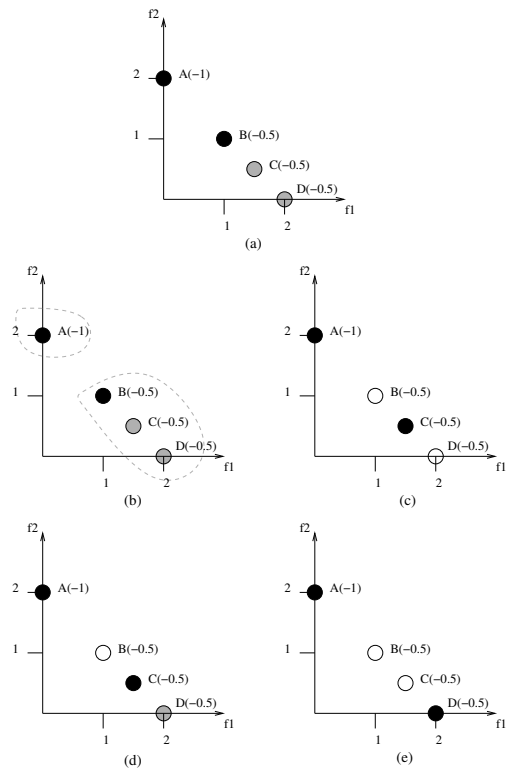

Fig. 2. Let's assume that we want to select two individuals. If we use only the maximin fitness function and we assume that A, B, C and D is the ordering of the solutions after sorting them with respect to their fitness value, then we select individuals A and B and individuals C and D are not considered (see (a)). This is clearly not a good selection. If we use the clustering technique proposed in [15], we take A and B as initial centers of the clusters and we obtain two clusters: the first one only has A and the second has B, C and D, see (b). When we recalculate the centers of the clusters and choose the closest solution to the centers, we select A and C, see (c). If we use our proposed technique, first we select A and B ($\mathbb{S} = A, B$). After that, we consider individual C; its nearest neighbor is B and we choose A as a random solution. First, C competes with A and C loses because the distance from A to B is greater than the distance from C to B. Then, C competes with B and C wins because the distance from C to A is greater than the distance from B to A. Finally, we consider D, and D loses with A but it wins with C. Then, we select A and D, see (e). We can say that (e) is a better choice than (a) and (c) in the selection process.

Algorithm 2: Verify similarity

Input : x (individual), $\mathbb{S}$ (population).
Output: Returns 1, if the individual x is similar to any individual in the population $\mathbb{S}$; otherwise, it returns 0.

```
for i ← 1 to ||S|| do
    for k ← 1 to K do
        if |x.f[k] − S[i].f[k]| < min_dif then
            return 1;
        end
    end
end
return 0;
```

third is based on decomposition. For this sake, we used the following MOEAs: MC-MOEA [16] (the version in which the modified maximin fitness is used all the time), SMS-EMOA-HYPE (a version of SMS-EMOA [4] that uses a fitness assignment based on the approximation of the hypervolume indicator, proposed in [8]) and MOEA/D [5] (using PBI to decompose the MOP). All of these MOEAs use the same operators to create new individuals, which allows a fair comparison.

Our MOEA is called **"Maximin-Distances Multi-Objective Evolutionary Algorithm (MD-MOEA)"** 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 to obtain a population of size $2P$. Then, we use the selection mechanism described in Algorithm 1 to choose the P individuals that will take part of the following generation. Finally, this process is repeated for a (pre-defined) number of generations.

VI. Experimental Results

As we mentioned before, we validated our selection mechanism by comparing our MD-MOEA with respect to MC-MOEA, MOEA/D and SMS-EMOA-HYPE. In the case of SMS-EMOA-HYPE, we used the source code of HyPE available in the public domain [8] adopting 10^4 as our number of samples to assign fitness in the original SMS-EMOA. In the case of MOEA/D, we generated the convex weights using the technique proposed in [13] and after that, we applied clustering (k-means) to obtain a specific number of weights.[3]

For our experiments, we used seven problems taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [23]. We used $k = 5$ for DTLZ1, DTLZ3 and DTLZ6 and $k = 10$ for the remaining test problems. Also, we used seven problems taken from the WFG toolkit [24], with $k_factor = 2$ and $l_factor = 10$. For each test problem, we performed 30 independent runs. For all four 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 $\eta_m = 20$, respectively. We performed a maximum of 50,000 fitness function evaluations (in this case, we used a population size of 100 individuals and we iterated for 500 generations). Only in DTLZ3 we performed 100,000 evaluations (we used a population size of 100 individuals and we iterated for 1000 generations).

A. Performance Indicators

We adopted only the hypervolume indicator (I_H) to validate our results because it rewards both convergence towards the Pareto front as well as the maximum spread of the solutions obtained. Furthermore, I_H is the only unary indicator which is known to be "Pareto compliant" [25]. To calculate the hypervolume indicator, we used the following reference points: $y_{ref} = [y_1, \cdots, y_M]$ such that $y_i = 0.7$ for DTLZ1, $y_{ref} = [y_1, \cdots, y_M]$ such that $y_i = 1.1$ for DTLZ(2-6), $y_{ref} = [y_1, \cdots, y_M]$ such that $y_M = 6.1$ and $y_{i \neq M} = 1.1$ for DTLZ7. In the case of the WFG test problems, we generated the reference point using the highest value found for each objetive function taking into account all the outputs of the four algorithms (i.e., MC-MOEA, MOEA/D, SMS-EMOA-HYPE and MD-MOEA).

B. Discussion of Results

Table I shows the results with respect to I_H in problems with three objective functions as well as the results of the

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

$\vec{f}$	mc-moea I_H	md-moea I_H	$P(H)$
DTLZ1 (3)	**0.311634 (0.001536)**	0.311213 (0.001637)	0.228 (0)
DTLZ2 (3)	0.696078 (0.008462)	**0.716900 (0.006070)**	0.000 (1)
DTLZ3 (3)	0.666299 (0.022591)	**0.720263 (0.007612)**	0.000 (1)
DTLZ4 (3)	0.703924 (0.008896)	**0.714999 (0.007642)**	0.000 (1)
DTLZ5 (3)	0.424854 (0.005358)	**0.433524 (0.005550)**	0.000 (1)
DTLZ6 (3)	0.381283 (0.026286)	**0.407466 (0.024926)**	0.000 (1)
DTLZ7 (3)	1.851081 (0.202273)	**1.854329 (0.208323)**	0.411 (0)
WFG1 (3)	**17.622577 (1.379125)**	17.365532 (1.698654)	0.129 (0)
WFG2 (3)	0.115221 (0.009956)	**0.116298 (0.010288)**	0.027 (1)
WFG3 (3)	0.407785 (0.006808)	**0.448216 (0.002539)**	0.000 (1)
WFG4 (3)	20.858572 (0.557784)	**25.854932 (0.333751)**	0.000 (1)
WFG5 (3)	7.997923 (0.203347)	**9.265981 (0.077585)**	0.000 (1)
WFG6 (3)	0.920378 (0.012003)	**0.977053 (0.011069)**	0.000 (1)
WFG7 (3)	17.286919 (0.727687)	**18.961153 (0.389552)**	0.000 (1)

(a)

$\vec{f}$	moead I_H	md-moea I_H	$P(H)$
DTLZ1 (3)	0.303053 (0.000450)	**0.311213 (0.001637)**	0.000 (1)
DTLZ2 (3)	0.708105 (0.000205)	**0.716900 (0.006070)**	0.000 (1)
DTLZ3 (3)	0.702575 (0.004807)	**0.720263 (0.007612)**	0.000 (1)
DTLZ4 (3)	0.708295 (0.000133)	**0.714999 (0.007642)**	0.000 (1)
DTLZ5 (3)	0.416468 (0.000529)	**0.433524 (0.005550)**	0.000 (1)
DTLZ6 (3)	0.352563 (0.026612)	**0.407466 (0.024926)**	0.000 (1)
DTLZ7 (3)	1.607354 (0.202304)	**1.854329 (0.208323)**	0.000 (1)
WFG1 (3)	16.211830 (0.312977)	**17.365532 (1.698654)**	0.063 (0)
WFG2 (3)	0.088530 (0.006124)	**0.116298 (0.010288)**	0.000 (1)
WFG3 (3)	0.388538 (0.014871)	**0.448216 (0.002539)**	0.000 (1)
WFG4 (3)	22.853573 (0.541417)	**25.854932 (0.333751)**	0.000 (1)
WFG5 (3)	8.501790 (0.147219)	**9.265981 (0.077585)**	0.000 (1)
WFG6 (3)	0.845541 (0.009395)	**0.977053 (0.011069)**	0.000 (1)
WFG7 (3)	16.227431 (1.859549)	**18.961153 (0.389552)**	0.000 (1)

(b)

$\vec{f}$	sms-emoa hype I_H	md-moea I_H	$P(H)$
DTLZ1 (3)	**0.314237 (0.000752)**	0.311213 (0.001637)	0.000 (1)
DTLZ2 (3)	**0.741222 (0.002104)**	0.716900 (0.006070)	0.000 (1)
DTLZ3 (3)	0.000000 (0.000000)	**0.720263 (0.007612)**	0.000 (1)
DTLZ4 (3)	**0.743932 (0.002429)**	0.714999 (0.007642)	0.000 (1)
DTLZ5 (3)	**0.438030 (0.000195)**	0.433524 (0.005550)	0.000 (1)
DTLZ6 (3)	0.363403 (0.062094)	**0.407466 (0.024926)**	0.003 (1)
DTLZ7 (3)	1.830760 (0.195849)	**1.854329 (0.208323)**	0.004 (1)
WFG1 (3)	**17.841521 (1.282806)**	17.365532 (1.698654)	0.021 (1)
WFG2 (3)	**0.116636 (0.010844)**	0.116298 (0.010288)	0.046 (1)
WFG3 (3)	0.446381 (0.004118)	**0.448216 (0.002539)**	0.046 (1)
WFG4 (3)	**27.035334 (0.224630)**	25.854932 (0.333751)	0.000 (1)
WFG5 (3)	**9.462112 (0.048270)**	9.265981 (0.077585)	0.000 (1)
WFG6 (3)	**0.987239 (0.007085)**	0.977053 (0.011069)	0.000 (1)
WFG7 (3)	18.288252 (0.873704)	**18.961153 (0.389552)**	0.001 (1)

(c)

TABLE I. RESULTS OBTAINED IN THE DTLZ AND WFG TEST PROBLEMS WITH THREE OBJECTIVE FUNCTIONS. WE COMPARE OUR MD-MOEA WITH RESPECT TO MC-MOEA, MOEA/D AND SMS-EMOA-HYPE, USING THE HYPERVOLUME INDICATOR (I_H). 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 WILCOXONS 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.

statistical analysis that we made to validate our experiments, for which we used Wilcoxon's rank sum. In (a), we can see that our MD-MOEA outperformed MC-MOEA in twelve cases and we can reject the null hypothesis (medians are equal) in eleven of them. On the other hand, only in two cases MC-MOEA obtained better results than our MD-MOEA and in these two cases we cannot reject the null hypothesis. That means that in eleven problems our MD-MOEA significantly outperforms MC-MOEA and in the remaining three both MC-MOEA and MD-MOEA have a similar behavior. Thus, we can say that our MD-MOEA is better than MC-MOEA. In (b), we can observe that our MD-MOEA is better than MOEA/D because it outperforms MOEA/D in all cases and only in one of them, we cannot reject the null hypothesis. Finally, in (c), we can see that our MD-MOEA outperformed SMS-EMOA-HYPE in five cases and it was outperformed in nine cases (in all cases the null hypothesis can be rejected). However, it is important to note that in DTLZ3, SMS-EMOA-HYPE was unable to converge to the true Pareto front. In contrast, our MD-MOEA was able to reach the true Pareto front in that problem.

Table II shows a scalability study of our proposed approach (using up to six objective functions). We chose four problems from the DTLZ test suite: DTLZ1, DTLZ3, DTLZ5 and DTLZ7[4] and two problems from the WFG toolkit: WFG3 and WFG7.[5] In (a), we can see that our proposed MD-MOEA still outperforms MC-MOEA because it obtained better results in all cases and only in three cases we cannot reject the null hypothesis. In (b), we can see that MOEA/D outperformed our MD-MOEA in five of the eighteen MOPs adopted. An important observation is that this happens in MOPs with degenerate Pareto fronts such as DTLZ5 and DTLZ6 and in MOPs with a disconnected Pareto front such as DTLZ7. In all problems, we can reject the null hypothesis. Therefore; we can say that our MD-MOEA is better than MOEA/D in most cases. Finally, in (c), we can see that our MD-MOEA outperformed SMS-EMOA-HYPE in most cases (fourteen out of eighteen), and only in three of them, we cannot reject the null hypothesis. Therefore, we claim that our proposed MD-MOEA maintains its good behavior as we increase the number of objective functions. In fact, our proposed MD-MOEA outperforms SMS-EMOA-HYPE in most cases when we use four or more objective functions. It is important to observe that SMS-EMOA-HYPE outperformed our proposed MD-MOEA in the same problems as MOEA/D did (DTLZ5 and DTLZ7).

Another important thing that we must consider is the running time required by the four algorithms. In Table III, we can see that our proposed MD-MOEA requires at most five seconds to solve problems with three objective functions while SMS-EMOA-HYPE requires up to 168 seconds. Then, although SMS-EMOA-HYPE obtains better results than our MD-MOEA in several three-objective MOPs, our

[4]We chose these problems because their Pareto fronts have different forms: linear (DTLZ1), concave (DTLZ3), degenerate (DTLZ5) and disconnected (DTLZ7).

[5]We chose these problems because MD-MOEA outperformed SMS-EMOA-HYPE in them, when using three objective functions and we wanted to see if this behavior would hold if we increased the number of objectives.

$\vec{f}$	mc-moea I_H	md-moea I_H	$P(H)$
DTLZ1 (4)	0.209699 (0.050196)	**0.227252 (0.001531)**	0.695 (0)
DTLZ3 (4)	0.857388 (0.036140)	**0.940285 (0.015157)**	0.000 (1)
DTLZ5 (4)	0.205774 (0.028565)	**0.232610 (0.033320)**	0.003 (1)
DTLZ7 (4)	0.554462 (0.066262)	**0.571259 (0.078772)**	0.004 (1)
DTLZ1 (5)	0.077149 (0.074751)	**0.161622 (0.000807)**	0.000 (1)
DTLZ3 (5)	0.999126 (0.042205)	**1.132718 (0.014944)**	0.000 (1)
DTLZ5 (5)	0.164502 (0.026344)	**0.165387 (0.026130)**	0.994 (0)
DTLZ7 (5)	0.064793 (0.013324)	**0.069806 (0.028375)**	0.045 (1)
DTLZ1 (6)	0.033675 (0.044938)	**0.113517 (0.000686)**	0.000 (1)
DTLZ3 (6)	1.060825 (0.096036)	**1.272017 (0.022573)**	0.000 (1)
DTLZ5 (6)	0.138851 (0.044069)	**0.195823 (0.027087)**	0.000 (1)
DTLZ7 (6)	0.003920 (0.001524)	**0.004332 (0.002933)**	0.529 (0)
WFG3 (4)	0.133793 (0.006850)	**0.186307 (0.001452)**	0.000 (1)
WFG7 (4)	126.673434 (4.468933)	**137.524340 (5.091123)**	0.000 (1)
WFG3 (5)	0.006545 (0.003399)	**0.034708 (0.000433)**	0.000 (1)
WFG7 (5)	887.581970 (39.672574)	**990.070618 (33.543485)**	0.000 (1)
WFG3 (6)	0.000055 (0.000053)	**0.004540 (0.000107)**	0.000 (1)
WFG7 (6)	7058.029077 (371.172528)	**7873.374654 (435.168613)**	0.000 (1)

(a)

$\vec{f}$	moead I_H	md-moea I_H	$P(H)$
DTLZ1 (4)	0.207214 (0.000826)	**0.227252 (0.001531)**	0.000 (1)
DTLZ3 (4)	0.849726 (0.008275)	**0.940285 (0.015157)**	0.000 (1)
DTLZ5 (4)	**0.395004 (0.003938)**	0.232610 (0.033320)	0.000 (1)
DTLZ7 (4)	0.510402 (0.136447)	**0.571259 (0.078772)**	0.000 (1)
DTLZ1 (5)	0.138851 (0.001183)	**0.161622 (0.000807)**	0.000 (1)
DTLZ3 (5)	0.907270 (0.016862)	**1.132718 (0.014944)**	0.000 (1)
DTLZ5 (5)	**0.384474 (0.004732)**	0.165387 (0.026130)	0.000 (1)
DTLZ7 (5)	**0.090559 (0.025208)**	0.069806 (0.028375)	0.000 (1)
DTLZ1 (6)	0.090773 (0.000779)	**0.113517 (0.000686)**	0.000 (1)
DTLZ3 (6)	0.836798 (0.127543)	**1.272017 (0.022573)**	0.000 (1)
DTLZ5 (6)	**0.386218 (0.005819)**	0.195823 (0.027087)	0.000 (1)
DTLZ7 (6)	**0.017423 (0.001653)**	0.004332 (0.002933)	0.000 (1)
WFG3 (4)	0.130169 (0.006990)	**0.186307 (0.001452)**	0.000 (1)
WFG7 (4)	65.878334 (9.319913)	**137.524340 (5.091123)**	0.000 (1)
WFG3 (5)	0.018217 (0.001263)	**0.034708 (0.000433)**	0.000 (1)
WFG7 (5)	292.456527 (30.794113)	**990.070618 (33.543485)**	0.000 (1)
WFG3 (6)	0.001995 (0.000324)	**0.004540 (0.000107)**	0.000 (1)
WFG7 (6)	2333.063125 (246.601646)	**7873.374654 (435.168613)**	0.000 (1)

(b)

$\vec{f}$	sms-emoa hype I_H	md-moea I_H	$P(H)$
DTLZ1 (4)	0.227248 (0.003551)	**0.227252 (0.001531)**	0.411 (0)
DTLZ3 (4)	0.000000 (0.000000)	**0.940285 (0.015157)**	0.000 (1)
DTLZ5 (4)	**0.407141 (0.006175)**	0.232610 (0.033320)	0.000 (1)
DTLZ7 (4)	0.471140 (0.233999)	**0.571259 (0.078772)**	0.258 (0)
DTLZ1 (5)	0.148712 (0.038414)	**0.161622 (0.000807)**	0.001 (1)
DTLZ3 (5)	0.020370 (0.073259)	**1.132718 (0.014944)**	0.000 (1)
DTLZ5 (5)	**0.397983 (0.012657)**	0.165387 (0.026130)	0.000 (1)
DTLZ7 (5)	0.054747 (0.052799)	**0.069806 (0.028375)**	0.082 (0)
DTLZ1 (6)	0.106482 (0.013763)	**0.113517 (0.000686)**	0.000 (1)
DTLZ3 (6)	0.014137 (0.053109)	**1.272017 (0.022573)**	0.000 (1)
DTLZ5 (6)	**0.398966 (0.011553)**	0.195823 (0.027087)	0.000 (1)
DTLZ7 (6)	**0.013773 (0.011388)**	0.004332 (0.002933)	0.000 (1)
WFG3 (4)	0.183098 (0.002400)	**0.186307 (0.001452)**	0.000 (1)
WFG7 (4)	98.364141 (7.586617)	**137.524340 (5.091123)**	0.000 (1)
WFG3 (5)	0.032837 (0.000599)	**0.034708 (0.000433)**	0.000 (1)
WFG7 (5)	497.644458 (43.994542)	**990.070618 (33.543485)**	0.000 (1)
WFG3 (6)	0.004313 (0.000114)	**0.004540 (0.000107)**	0.000 (1)
WFG7 (6)	3756.921293 (286.752275)	**7873.374654 (435.168613)**	0.000 (1)

(c)

TABLE II. RESULTS OBTAINED IN THE DTLZ AND WFG TEST PROBLEMS WITH FOUR, FIVE AND SIX OBJECTIVE FUNCTIONS. WE COMPARE OUR MD-MOEA WITH RESPECT TO MC-MOEA, MOEA/D AND SMS-EMOA-HYPE, USING THE HYPERVOLUME INDICATOR (I_H). 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 WILCOXONS 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.

MD-MOEA requires much less time while obtaining competitive results. As we increase the number of objectives, our proposed MD-MOEA requires at most seven seconds while SMS-EMOA-HYPE requires up to 445 seconds. Furthermore, MD-MOEA outperformed SMS-EMOA-HYPE in most of these problems with more than three objectives. With respect to MC-MOEA, our approach requires a similar running time. MOEA/D is faster than our proposed MD-MOEA but not for a significant time difference.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a new selection mechanism based on the modified maximin fitness function and the use of Euclidean distances between solutions (in objective function space). Our idea is to use this modified maximin fitness function to select individuals and, after that, correct the possible errors in the selection process with respect to diversity in the population using the Euclidean distances between solutions. Our new selection mechanism has a linear complexity with respect to the number of objective functions and it is therefore suitable for solving many-objective optimization problems. Our preliminary results indicate that our proposed MD-MOEA is able to outperform MOEAs such as MC-MOEA and MOEA/D and that it is competitive with respect to a version of SMS-EMOA that uses a fitness assignment mechanism based on the approximation of the hypervolume (SMS-EMOA-HYPE). Indeed, as we increase the number of objective functions, our proposed MD-MOEA outperforms SMS-EMOA-HYPE in most cases. Finally, it is important to mention that our proposed approach requires much less running time than SMS-EMOA-HYPE and it does not require any additional information such as MOEA/D and SMS-EMOA.

As part of our future work, we are interested in further studying the reasons for which our proposed mechanism has difficulties to deal with MOPs having degenerate Pareto fronts (e.g., DTLZ5 and DTLZ6) and disconnected Pareto fronts (e.g., DTLZ7). Also, we want to study the possibility of hybridizing our selection mechanism with some performance indicator such as the hypervolume or $R2$.

REFERENCES

[1] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*. New York: Springer, second ed., September 2007. ISBN 978-0-387-33254-3.

[2] 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.

[3] M. Emmerich, N. Beume, and B. Naujoks, "An EMO Algorithm Using the Hypervolume Measure as Selection Criterion," in *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO*

$\vec{f}$	mc-moea time	moead time	sms-emoa-hype time	md-moea time
DTLZ1 (3)	≈ 0 s	≈ **0 s**	≈ 47 s	≈ 0 s
DTLZ2 (3)	≈ 1 s	≈ **0 s**	≈ 106 s	≈ 2 s
DTLZ3 (3)	≈ 2 s	≈ **0 s**	≈ 135 s	≈ 2 s
DTLZ4 (3)	≈ 2 s	≈ **0 s**	≈ 107 s	≈ 2 s
DTLZ5 (3)	≈ 1 s	≈ **0 s**	≈ 64 s	≈ 1 s
DTLZ6 (3)	≈ 1 s	≈ **0 s**	≈ 59 s	≈ 1 s
DTLZ7 (3)	≈ 2 s	≈ **0 s**	≈ 98 s	≈ 1 s
DTLZ1 (4)	≈ 2 s	≈ **0 s**	≈ 59 s	≈ 1 s
DTLZ3 (4)	≈ 3 s	≈ **0 s**	≈ 165 s	≈ 3 s
DTLZ5 (4)	≈ 2 s	≈ **0 s**	≈ 143 s	≈ 3 s
DTLZ7 (4)	≈ 2 s	≈ **0 s**	≈ 185 s	≈ 2 s
DTLZ1 (5)	≈ 2 s	≈ **0 s**	≈ 79 s	≈ 1 s
DTLZ3 (5)	≈ 4 s	≈ **0 s**	≈ 177 s	≈ 4 s
DTLZ5 (5)	≈ 3 s	≈ **0 s**	≈ 229 s	≈ 4 s
DTLZ7 (5)	≈ 2 s	≈ **0 s**	≈ 296 s	≈ 2 s
DTLZ1 (6)	≈ 2 s	≈ **0 s**	≈ 98 s	≈ 2 s
DTLZ3 (6)	≈ 4 s	≈ **0 s**	≈ 185 s	≈ 4 s
DTLZ5 (6)	≈ 3 s	≈ **0 s**	≈ 336 s	≈ 4 s
DTLZ7 (6)	≈ 3 s	≈ **0 s**	≈ 377 s	≈ 2 s
WFG1 (3)	≈ 3 s	≈ **1 s**	≈ 147 s	≈ 4 s
WFG2 (3)	≈ 2 s	≈ **1 s**	≈ 98 s	≈ 1 s
WFG3 (3)	≈ 3 s	≈ **1 s**	≈ 148 s	≈ 4 s
WFG4 (3)	≈ 3 s	≈ **1 s**	≈ 107 s	≈ 4 s
WFG5 (3)	≈ 3 s	≈ **1 s**	≈ 153 s	≈ 4 s
WFG6 (3)	≈ 3 s	≈ **1 s**	≈ 168 s	≈ 4 s
WFG7 (3)	≈ 4 s	≈ **1 s**	≈ 151 s	≈ 5 s
WFG3 (4)	≈ 3 s	≈ **1 s**	≈ 247 s	≈ 3 s
WFG7 (4)	≈ 5 s	≈ **1 s**	≈ 252 s	≈ 5 s
WFG3 (5)	≈ 4 s	≈ **1 s**	≈ 378 s	≈ 2 s
WFG7 (5)	≈ 6 s	≈ **1 s**	≈ 358 s	≈ 7 s
WFG3 (6)	≈ 4 s	≈ **1 s**	≈ 445 s	≈ 2 s
WFG7 (6)	≈ 7 s	≈ **2 s**	≈ 408 s	≈ 6 s

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

2005 (C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, eds.), (Guanajuato, México), pp. 62–76, Springer. Lecture Notes in Computer Science Vol. 3410, March 2005.

[4] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, pp. 1653–1669, 16 September 2007.

[5] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, pp. 712–731, December 2007.

[6] K. Bringmann and T. Friedrich, "Approximating the volume of unions and intersections of high-dimensional geometric objects," *Computational Geometry*, vol. 43, no. 67, pp. 601 – 610, 2010.

[7] H. Ishibuchi, N. Tsukamoto, Y. Sakane, and Y. Nojima, "Indicator-Based Evolutionary Algorithm with Hypervolume Approximation by Achievement Scalarizing Functions," in *Proceedings of the 12th annual conference on Genetic and Evolutionary Computation (GECCO'2010)*, (Portland, Oregon, USA), pp. 527–534, ACM Press, July 7–11 2010. ISBN 978-1-4503-0072-8.

[8] 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.

[9] K. Bringmann and T. Friedrich, "Approximating the least hypervolume contributor: NP-hard in general, but fast in practice," *Theoretical Computer Science*, vol. 425, pp. 104–116, March 30 2012.

[10] K. Bringmann and T. Friedrich, "Convergence of Hypervolume-Based Archiving Algorithms II: Competitiveness," in *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, (Philadelphia, USA), pp. 457–464, ACM Press, July 2012. ISBN: 978-1-4503-1177-9.

[11] M. T. Emmerich and C. M. Fonseca, "Computing Hypervolume Contributions in Low Dimensions: Asymptotically Optimal Algorithm and Complexity Results," in *Evolutionary Multi-Criterion Optimization, 6th International Conference, EMO 2011* (R. H. Takahashi, K. Deb, E. F. Wanner, and S. Grecco, eds.), (Ouro Preto, Brazil), pp. 121–135, Springer. Lecture Notes in Computer Science Vol. 6576, April 2011.

[12] A. Menchaca-Mendez and C. A. Coello Coello, "A New Selection Mechanism Based on Hypervolume and its Locality Property," in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, (Cancún, México), pp. 924–931, IEEE Press, 20-23 June 2013.

[13] 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.

[14] Y. yan Tan, Y. chang Jiao, H. Li, and X. kuan Wang, "Moea/d + uniform design: A new version of moea/d for optimization problems with many objectives," *Computers & Operations Research*, vol. 40, no. 6, pp. 1648 – 1660, 2013.

[15] A. Menchaca-Mendez and C. A. Coello Coello, "Solving Multi-Objective Optimization Problems using Differential Evolution and a Maximin Selection Criterion," in *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*, (Brisbane, Australia), pp. 3143–3150, IEEE Press, June 10-15 2012.

[16] A. Menchaca-Mendez and C. A. Coello Coello, "Selection Operators based on Maximin Fitness Function for Multi-Objective Evolutionary Algorithms," in *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, (Sheffield, UK), Springer. Lecture Notes in Computer Science Vol. 6576, March 2013.

[17] R. Balling, "Pareto sets in decision-based design," *Journal of Engineering Valuation and Cost Analysis*, vol. 3, pp. 189–198, 2000.

[18] R. Balling and S. Wilson, "The Maximin Fitness Function for Multi-objective Evolutionary Computation: Application to City Planning," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)* (L. Spector, E. D. Goodman, A. Wu, W. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, eds.), (San Francisco, California), pp. 1079–1084, Morgan Kaufmann Publishers, 2001.

[19] R. Balling, "The Maximin Fitness Function; Multiobjective City and Regional Planning," in *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003* (C. M. Fonseca, P. J. Fleming, E. Zitzler, K. Deb, and L. Thiele, eds.), (Faro, Portugal), pp. 1–15, Springer. Lecture Notes in Computer Science. Volume 2632, April 2003.

[20] X. Li, "Better Spread and Convergence: Particle Swarm Multiobjective Optimization Using the Maximin Fitness Function," in *Genetic and Evolutionary Computation–GECCO 2004. Proceedings of the Genetic and Evolutionary Computation Conference. Part I* (K. D. et al., ed.), (Seattle, Washington, USA), pp. 117–128, Springer-Verlag, Lecture Notes in Computer Science Vol. 3102, June 2004.

[21] X. Li, J. Branke, and M. Kirley, "On Performance Metrics and Particle Swarm Methods for Dynamic Multiobjective Optimization Problems," in *2007 IEEE Congress on Evolutionary Computation (CEC'2007)*, (Singapore), pp. 576–583, IEEE Press, September 2007.

[22] H. Li, X. Huang, and Q. Feng, "Optimizing expressway maintenance planning by coupling ant algorithm and geography information system transportation in hubei province, china," in *Geoscience and Remote Sensing Symposium (IGARSS), 2011 IEEE International*, pp. 2977 –2979, july 2011.

[23] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable Test Problems for Evolutionary Multiobjective Optimization," in *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications* (A. Abraham, L. Jain, and R. Goldberg, eds.), pp. 105–145, USA: Springer, 2005.

[24] 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.

[25] 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*, vol. 7, pp. 117–132, April 2003.