GDE-MOEA : A New MOEA based on the Generational Distance indicator and ϵ-dominance

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 ϵ-dominance which is called "ϵ-selection". An interesting feature of this selection scheme is that it does not require to set the value of ϵ ahead of time. Our ϵ-selection is incorporated into the GD-MOEA algorithm, giving rise to the so-called "Generational Distance & ϵ-dominance Multi-Objective Evolutionary Algorithm (GDE-MOEA)". Our proposed GDE-MOEA is validated using standard test functions taken from the specialized literature, having three to six objective functions. GDE-MOEA is compared with respect to the original GD-MOEA, which is based on the generational distance indicator and a technique based on Euclidean distances to improve the diversity in the population. Additionally, our proposed approach is compared with respect to 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 proposed GDE-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 GD-MOEA and MOEA/D in most cases and it is competitive with respect to SMS-EMOA-HYPE but at a much lower computational cost.**

I. INTRODUCTION

In the real world, there are many optimization problems which involve multiple objective functions. These objective functions are usually in conflict with each other and they have to be satisfied simultaneously. These types of problems are called *multi-objective optimization problems (MOPs)*. When we solve MOPs, we want to find the best possible trade-offs among the objectives, therefore, 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. There are several indicators to assess the quality of the approximation of the Pareto optimal set generated by a MOEA, e.g., hypervolume, $R2$-indicator, Δ_p-indicator, ϵ-indicator, etc. [1]. However, very few indicators are "Pareto Compliant" [1].

MOEAs can be classified in two groups according to their selection mechanism: (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,[2] MOEAs of type (ii) have become relatively popular in recent years, e.g., MOEAs based on performance indicators. MOEAs based on the hypervolume indicator (I_H) have been popular because I_H is the only unary indicator which is known to be "Pareto compliant". Additionally, Fleischer proved in [3] that, given a finite search space and a reference point, maximizing I_H is equivalent to finding the Pareto optimal set. Some examples of MOEAs based on I_H are described in [4], [5], [6], [7], [8]. Perhaps the best-known indicator-based MOEA is SMS-EMOA [6]. An important disadvantage of this type of MOEAs is that the problem of computing I_H is *#P-hard* [9]. For this reason, they are impractical when we want to solve MOPs having four or more objective functions.

In 2012, Brockhoff et al. [10] conducted a study about the properties of the $R2$-indicator (I_{R2}). After that, a number of proposals of MOEAs based on I_{R2} were introduced [11], [12], [13], [14]. These MOEAs can solve MOPs with many objective functions at an affordable computational cost. However, they need to generate a set of well-distributed convex weights and this task becomes more difficult as we increase the number of objective functions. The same applies to MOEAs based on decomposition (which decompose the MOP into a number of scalar optimization subproblems) like the well-known MOEA/D [15]. Another popular indicator in recent years is the Δ_p-indicator ($I_{\Delta_{pi}}$). It was introduced by Schütze et al. [16] in 2012 and some MOEAs based on it have already been proposed [17], [18], [19]. $I_{\Delta_{pi}}$ is composed of slight modifications of two well-known indicators: generational distance (I_{GD}) [20] and inverted generational distance (I_{IGD}) [21]. Therefore, it is necessary to know the true Pareto front to calculate $I_{\Delta_{pi}}$ and perhaps

The second author acknowledges support from CONACyT project no. 221551.

[1]An indicator $I : \Omega \to \mathbb{R}$ is **Pareto compliant** if for all $\mathcal{A}, \mathcal{B} \in \Omega : \mathcal{A} \preceq B \Rightarrow I(\mathcal{A}) \geq I(\mathcal{B})$, assuming that greater indicator values correspond to higher quality.

[2]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].

this is the most important disadvantage of MOEAs based on $I_{\Delta_{pi}}$: not being able to produce a good reference set could cause that the algorithm cannot generate the complete Pareto front, or that it generates poorly distributed solutions. In extreme cases, the algorithm could not even converge to the true Pareto front.

Recently, in [22], the authors proposed a MOEA based on I_{GD} which was called "GD-MOEA". GD-MOEA uses the set of nondominated solutions at each generation as a reference set to calculate the fitness of the dominated individuals. They argued that in this case it is not necessary to have a well-distributed reference set because GD-MOEA uses I_{GD} only as a convergence strategy and it uses two other techniques to maintain diversity in the population. According to the experimental results presented, this MOEA could outperform MOEA/D in several standard test problems and it was competitive with respect to a version of SMS-EMOA that uses a fitness assignment scheme based on the approximation of I_H. We found these results to be very interesting because they imply that when many (even all) solutions are nondominated, the diversity mechanism plays the most important role in the search, i.e., in many-objective optimization problems, the diversity mechanism can determine if the MOEA is able to converge to the true Pareto front. For this reason, we are interested in ϵ-dominance [23] as a strategy to maintain diversity in the population. In 2005, Deb et al. proposed the ϵ-MOEA. This MOEA uses an archive with a fixed size in which the nondominated solutions are stored. The idea is that the search space is divided in hypercubes of size equal to ϵ and only one nondominated individual can reside in each hypercube. ϵ-MOEA was able to outperform NSGA-II, C-NSGA-II, PESA and SPEA2 and it is also computationally efficient. The most important disadvantage of this MOEA is related to setting the value of ϵ: for defining the right ϵ value, it is necessary to known the true Pareto front as well as the number of nondominated solutions that we want to store.

In this paper, we propose a technique based on ϵ-dominance to maintain diversity in the population and we use it instead of the technique based on Euclidean distances adopted by GD-MOEA. An interesting aspect of our proposed approach is that the value of ϵ is not an input to the algorithm. We called the new algorithm "GDE-MOEA". The remainder of this paper is organized as follows. Section II states the problem of our interest. Section III describes the concept of ϵ-dominance. The original GD-MOEA is explained in Section IV. Our ϵ-selection mechanism is presented in Section V. The complete GDE-MOEA is discussed in Section VI. Our experimental validation and the results obtained are shown in Section VII. Finally, we provide our conclusions and some possible paths for future work in Section VIII.

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

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

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.

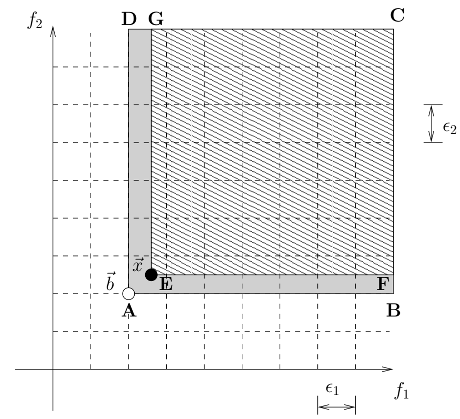

Fig. 1. ϵ-dominance. Vector $\vec{b}$ is the identification array of solution $\vec{x}$ and it defines the area (ABCDA) which is ϵ-dominated by $\vec{x}$. Using Pareto dominance, $\vec{x}$ dominates any solution in the area EFCGE.

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

III. ϵ-Dominance

To define ϵ-dominance between solutions, it is necessary to calculate an identification array $\vec{b}_i$ for each solution $\vec{x}_i$ as follows: $\vec{b}_i = [b_1, b_2, \cdots, b_k]$ where $b_j = (\lfloor (f_j(\vec{x}_i) - f_j^{min})/\epsilon_j \rfloor) * \epsilon_j$. f_j^{min} is the minimum value of the j-th objective and ϵ_j is the allowable 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 a same hypercube have the same identification array.

IV. GD-MOEA: A MOEA based on the Generational Distance Indicator

GD-MOEA is a MOEA based on I_{GD} which also uses a technique based on Euclidean distances to improve the diversity in the population. It works as follows: First, it creates an initial population of size P. After that, it creates P new individuals using the operators of NSGA-II (crossover and mutation). It combines the population of parents and offspring to obtain a population of size $2P$. Then, it selects the P individuals that will take part of the following generation.

Finally, it repeats this process for a (pre-defined) number of generations. Its selection process works as follows: Suppose that $\mathcal{P}$ is the population which contains the parents and offspring of the current generation; therefore, $|\mathcal{P}| = 2P$. Then, GD-MOEA wants to select P individuals from $\mathcal{P}$. First, it has to obtain the nondominated individuals in $\mathcal{P}$ and it puts them in $\mathcal{S}$. The remaining individuals (dominated individuals) are placed in $\mathcal{B}$.

If $P > \|\mathcal{S}\|$, it selects the remaining r individuals (where $r = P - \|\mathcal{S}\|$) from $\mathcal{B}$ as follows: It calculates the Euclidean distance, d_i, from each dominated individual in $\mathcal{B}$ to its nearest neighbor in $\mathcal{S}$, and also, it is necessary to save its neighboring nondominated individual. After that, it has to sort $\mathcal{B}$ regarding d_i and it creates another set called "$\mathcal{S}' = \emptyset$". Finally, for each $\vec{x}_i \in \mathcal{B}$, it checks if its nearest neighbor in $\mathcal{S}$ is equal to the nearest neighbor in $\mathcal{S}$ of some individual in $\mathcal{S}'$. If the answer is no and $\|\mathcal{S}'\| < r$, then, it puts $\vec{x}_i$ in $\mathcal{S}'$. If all individuals in $\mathcal{B}$ are considered and $\|\mathcal{S}'\| < r$, it repeats the last process but now it will allow that only one individual in $\mathcal{S}'$ has the same neighbor that the individual that it wants to select. It iterates until it obtains r individuals.

If $P < \|\mathcal{S}\|$, it chooses s individuals from $\mathcal{S}$ randomly. These individuals are placed in $\mathcal{S}$ and it puts the remaining nondominated individuals in a new set called "$\mathcal{B}$". For each solution $\vec{x} \in \mathcal{B}$, it checks if it is similar (in objective function space) to any selected individual in $\mathcal{S}$.[3] If it is similar, the solution is not considered to be selected, else GD-MOEA obtains its nearest neighbor from $\mathcal{S}$, $\vec{x}_{near}$, and it chooses a random individual from $\mathcal{S}$, $\vec{x}_{random}$ such that $\vec{x}_{near} \neq \vec{x}_{random}$, and then, these three solutions compete to survive. First, $\vec{x}$ competes with $\vec{x}_{random}$. If the Euclidean distance from $\vec{x}$ to its nearest neighbor in $\mathcal{S}$ is greater than the Euclidean distance from $\vec{x}_{random}$ to its nearest neighbor in $\mathcal{S}$, $\vec{x}$ replaces $\vec{x}_{random}$. If $\vec{x}$ loses the competition, $\vec{x}$ competes with its nearest neighbor to survive. If the Euclidean distance from $\vec{x}$ to its nearest neighbor in $\mathcal{S}$ (without considering $\vec{x}_{near}$) is greater than the Euclidean distance from $\vec{x}_{near}$ to its nearest neighbor in $\mathbb{S}$, then $\vec{x}$ replaces $\vec{x}_{near}$.

V. Our proposal: an ϵ-selection mechanism

As we know, when we want to solve many-objective optimization problems the number of nondominated solutions grows exponentially, and then, at early stages of the search of a MOEA it is not possible to know which individuals should be selected because most of them are nondominated. In these cases, we think that the diversity techniques play an important role in the search because if we can explore the whole space, it is possible to find solutions which guide us to the true Pareto front. Since ϵ-dominance divides the whole space in a number of hypercubes, we can control that only one nondominated solution resides in each hypercube: if two solutions $\vec{x}_1$ and $\vec{x}_2$ have the same identification array ($\vec{b}_1 = \vec{b}_2$) then we prefer the nearest solution to the identification array. The main disadvantage when we work with ϵ-dominance is to determine the value of ϵ_j for each objective function ($\vec{\epsilon} = [\epsilon_1, \cdots, \epsilon_k]$). To address this disadvantage, we propose the following: Suppose that we want to select S individuals from a population of nondominated individuals called $\mathcal{ND}$. Then, we divide each objective function in two equal parts:

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

Where f_j^{max} and f_j^{min} are the maximun and minimum value for the objective function j, considering all solutions in $\mathcal{ND}$, respectively. Then, we proceed to select the individuals (we select a single nondominated individual for each hypercube in the search space) and we put them in a set called $\mathcal{S}$. After that, if we have not selected the total number of individuals ($|\mathcal{S}| < S$) then we divide each objective function in three equal parts:

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

And, we still select the remaining individuals: We repeat this process until selecting S nondominated individuals. It is important to verify that individuals of different hypercubes are not similar (we check similarity in the same way that GD-MOEA). If we do not check similarity, we could obtain only weakly Pareto optimal points or the convergence to the true Pareto front could be slower.

VI. GDE-MOEA: A new MOEA based on Generational Distance Indicator and on ϵ-dominance

In this paper, we propose a new MOEA based on GD-MOEA but instead of using its technique based on Euclidean distances to improve the diversity in the population, we propose to use the ϵ-selection mechanism proposed in the above Section, i.e., we use I_{GD}-selection as a convergence strategy and we use the ϵ-selection mechanism to explore the whole search space at early stages of the search and to improve the distribution of solutions along the Pareto front at the end of the search. The new MOEA is called "Generational Distance & ϵ-dominance Multi-Objective Evolutionary Algorithm (GDE-MOEA)". Our GDE-MOEA works as follows: First, it creates an initial population of size P. After that, it creates P new individuals using the operators of NSGA-II (crossover and mutation). It combines the population of parents and offspring to obtain a population of size $2P$. Then, it selects the P individuals that will take part of the following generation. The complete selection process that we propose is shown in Algorithm 1. Finally, it repeats this process for a (pre-defined) number of generations. It is important to mention that our selection mechanism is applied on the objective function space and that the population has to be normalized.

VII. Experimental Results

We validated our proposed GDE-MOEA by comparing it with respect to GD-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 [24] adopting 10^4 as our number of samples to assign fitness in the original SMS-EMOA. This is because our main aim is to validate the effect of our selection mechanism. Therefore, all MOEAs used for the comparison must create the individuals in the same way in order to allow a fair comparison. In the case of MOEA/D, we generated the convex weights using

[3] GD-MOEA considers that one individual $\vec{x}$ is similar to another individual $\vec{y}$, if it is similar in any objective function: $\vec{x}.f_i - \vec{y}.f_i < \epsilon$, where ϵ is a small value.

Algorithm 1: GDE-Selection

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

Put in $\mathcal{S}$ the nondominated individuals of $\mathcal{P}$;
if $s > \|\mathcal{S}\|$ **then**
    /* I_{GD}-selection */
    Put in $\mathcal{B}$ the dominated individuals of $\mathcal{P}$;
    Calculate the Euclidean distance d_i from each individual $\vec{x}_i \in \mathcal{B}$ to its nearest neighbor in $\mathcal{S}$ and we also save its closest nondominated neighbor;
    Sort $\mathcal{B}$ with respect to d (ascending order);
    $\mathcal{S}' \leftarrow \emptyset$, $r \leftarrow s - \|\mathcal{S}\|$, $contIndAux \leftarrow 0$, $i \leftarrow 1$;
    while $\|\mathcal{S}'\| < r$ **do**
        $contInd \leftarrow 0$;
        foreach $\vec{s} \in \mathcal{S}'$ **do**
            if $\vec{s}.neighbor = \mathcal{B}.\vec{x}_i.neighbor$ **then**
                $contInd \leftarrow contInd + 1$;
            end
        end
        if $contInd \leq contIndAux$ **then**
            Put $\mathcal{B}.\vec{x}_i$ in $\mathcal{S}'$;
        end
        repeat
            $i \leftarrow i + 1$;
        until $\mathcal{B}.\vec{x}_i \notin \mathcal{S}'$;
        if $i = \|\mathcal{B}\|$ **then**
            $i \leftarrow 0$, $contIndAux \leftarrow contIndAux + 1$;
        end
    end
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$;
else
    /* ϵ-selection */
    if $s < \|\mathcal{S}\|$ **then**
        /* Set the number of divisions */
        $n \leftarrow 1$;
        /* Initialize the set of selected individuals */
        $\mathcal{S} \leftarrow \emptyset$;
        while $|\mathcal{S}| < S$ **do**
            $n \leftarrow n + 1$;
            Set the vector $\vec{\epsilon}$: $\epsilon_j \leftarrow (f_j^{max} - f_j^{min})/n$ (where j indicates the objective function);
            Update the identification array for each individual in $\mathcal{ND}$ and for each individual in $\mathcal{S}$;
            foreach $\vec{x}_i \in \mathcal{ND}$ *and* $S < |\mathcal{S}|$ **do**
                if $\vec{x}_i$ *is not similar to any individual in* $\mathcal{S}$ **then**
                    $flag \leftarrow 0$;
                    foreach $\vec{s}_i \in \mathcal{S}$ **do**
                        if $\vec{s}_i.\vec{b} = \vec{x}_i.\vec{b}$ **then**
                            if $\vec{x}_i$ *is nearest to* $\vec{b}$ *than* $\vec{s}_i$ **then**
                                $\vec{x}_i$ replaces $\vec{s}_i$;
                                $flag \leftarrow 1$;
                            end
                        end
                    end
                    if $flag = 0$ **then**
                        Put $\vec{x}_i$ in $\mathcal{S}$: $\mathcal{S} \leftarrow \mathcal{S} \cup \vec{x}_i$;
                    end
                end
            end
        end
    end
end
return $\mathcal{S}$;

the technique proposed in [25] and after that, we applied clustering (k-means) to obtain a specific number of weights.[4]

For our experiments, we used seven problems taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [26]. 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 [27], 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).

A. Performance Indicators

We adopted two indicators to validate our results: the hypervolume indicator (I_H) and the two set coverage indicator (I_{SC}). Both of them are "Pareto compliant". I_H rewards both convergence towards the Pareto front as well as the maximum spread of the solutions obtained. And, I_{SC} only measures convergence. If Λ denotes the Lebesgue measure, I_H is defined as:

$$I_H(\mathcal{A}, \vec{y}_{ref}) = \Lambda \left(\bigcup_{\vec{y} \in \mathcal{A}} \{\vec{x} \mid \vec{y} \prec \vec{x} \prec \vec{y}_{ref}\} \right) \tag{4}$$

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 nondominated solutions found and from this set we calculate the maximum and minimum for each objective function).

Let $\mathcal{A}, \mathcal{B}$ be two approximations of the Pareto optimal set, I_{SC} is defined as follows:

$$I_{SC}(\mathcal{A}, \mathcal{B}) = \frac{|\vec{b} \in \mathcal{B} \text{ such that } \exists \vec{a} \in \mathcal{A} \text{ with } \vec{a} \prec \vec{b}|}{|\mathcal{B}|}$$

If all points in $\mathcal{A}$ dominate or are equal to all points in $\mathcal{B}$, then by definition $I_{SC} = 1$. $I_{SC} = 0$ implies that no element in $\mathcal{B}$ is dominated by any element of $\mathcal{A}$. In general, both $I_{SC}(\mathcal{A}, \mathcal{B})$ and $I_{SC}(\mathcal{B}, \mathcal{A})$ have to be considered.

B. Discussion of Results

Table I shows the results with respect to I_H as well as the results of the statistical analysis (using Wilcoxon's rank sum) that we made to validate our experiments. Tables I(a) and I(d) show the comparison with respect to "GD-MOEA"

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

and we can see that in the DTLZ test problems GD-MOEA outperforms our GDE-MOEA in ten cases, GDE-MOEA outperforms GD-MOEA in fourteen cases and they have a similar behavior (we cannot reject the null hypothesis: "medians are equal") in four cases. In the WFG test problems, we can see that our GDE-MOEA outperforms GD-MOEA in twenty-two cases and it is only outperformed in six cases. Therefore, we can say that our GDE-MOEA outperforms GD-MOEA in most cases. Also, if we look at Tables II(a) and II(d), we can see that our GDE-MOEA is faster than GD-MOEA in most cases. Only in four cases, GD-MOEA required less time than our GDE-MOEA to obtain the approximation of the Pareto front but in the fifty-two remaining cases our GDE-MOEA is faster. In fact, if we consider the worst time required by each algorithm, we can say that our GDE-MOEA is 1.10 times faster than GD-MOEA.

Tables I(b) and I(e) show the comparison with respect to "MOEA/D". In the DTLZ test problems, our GDE-MOEA outperforms MOEA/D in eighteen cases, it is outperformed in eight cases and they obtain similar results in two cases. In the WFG test problems, our GDE-MOEA outperforms MOEA/D in twenty-four cases and it is outperformed in four cases. Therefore, we can say that our GDE-MOEA outperforms MOEA/D in most cases. With respect to the time required by each algorithm, we can see in Tables II(b) and II(e) that MOEA/D is faster than our GDE-MOEA in most problems (fourty-eight of fifty-six cases). However, it is important to keep in mind that our GDE-MOEA outperforms MOEA/D in several cases with a significant difference and MOEA/D is only 1.29 times faster than GDE-MOEA (considering the worst case for each algorithm).

Tables I(c) and I(f) show the results with respect to "SMS-EMOA-HYPE" and we can see that SMS-EMOA-HYPE outperforms our GDE-MOEA in twenty-one DTLZ test problems, they have a similar behavior in three cases and our GDE-MOEA outperforms SMS-EMOA in four DTLZ test problems. Regarding the WFG test problems, our GDE-MOEA outperforms SMS-EMOA-HYPE in five cases, they have a similar behavior in six cases and it is outperformed in seventeen cases. However, if we look at the time required by each algorithm, see Tables II(c) II(f), we can note that our GDE-MOEA is faster than SMS-EMOA-HYPE in all cases and the difference is significant: GDE-MOEA requires at most 2.35 seconds to solve a MOP with six objective functions while SMS-EMOA-HYPE needs up to 445.73 seconds to solve a MOP with six objective functions, i.e., GDE-MOEA is 189.67 times faster than SMS-EMOA-HYPE.

Finally, for a more detailed comparison between the original version of GD-MOEA and our GDE-MOEA, we decided to use I_{SC}. Table III shows the results obtained by the original GD-MOEA and our GDE-MOEA with respect to I_{SC} and we can corroborate that GDE-MOEA is better than GD-MOEA in most cases (even if we only consider convergence). Only in three cases both algorithms cannot cover any solution found by the other algorithm. However, in forty-four cases, GDE-MOEA covered a higher percentage of solutions found by GD-MOEA than the percentage of solutions found by GD-MOEA which managed to cover only some of the solutions found by GDE-MOEA. And, only in nine cases the percentage of solutions found by GD-MOEA that cover some solution found by GDE-MOEA is higher than the percentage of solutions found by GDE-MOEA which covers some solution found by GD-MOEA.

An interesting thing that we must note is that our GDE-MOEA has serious difficulties to solve the DTLZ1 test problem. Although this test problem has a linear Pareto front is well-known that when we solve this test problem many weakly dominated solutions are generated during the search process. This indicates that our technique still has room for improvement.

$\vec{f}$	gd-moea I_{SC} $(\mathcal{A},\mathcal{B})$	gde-moea I_{SC} $(\mathcal{B},\mathcal{A})$	$\vec{f}$	gd-moea I_{SC} $(\mathcal{A},\mathcal{B})$	gde-moea I_{SC} $(\mathcal{B},\mathcal{A})$
DTLZ1 (3)	0.014000	**0.139333**	WFG1 (3)	0.000000	**0.003000**
DTLZ2 (3)	0.007667	**0.271667**	WFG2 (3)	0.429667	**0.734667**
DTLZ3 (3)	0.106667	**0.171333**	WFG3 (3)	0.000667	**0.378667**
DTLZ4 (3)	0.010667	**0.290333**	WFG4 (3)	0.000000	**0.772000**
DTLZ5 (3)	0.130000	**0.334667**	WFG5 (3)	0.000000	**0.204333**
DTLZ6 (3)	**0.777667**	0.524667	WFG6 (3)	0.000333	**0.394333**
DTLZ7 (3)	0.144667	**0.188667**	WFG7 (3)	0.002667	**0.358667**
DTLZ1 (4)	**0.503000**	0.120000	WFG1 (4)	0.000000	**0.000333**
DTLZ2 (4)	0.004333	**0.142000**	WFG2 (4)	0.210333	**0.798667**
DTLZ3 (4)	0.165667	**0.359667**	WFG3 (4)	0.002333	**0.640333**
DTLZ4 (4)	0.003667	**0.155667**	WFG4 (4)	0.000333	**0.381667**
DTLZ5 (4)	0.129333	**0.390000**	WFG5 (4)	0.000000	**0.110333**
DTLZ6 (4)	0.077667	**0.914667**	WFG6 (4)	0.014000	**0.303000**
DTLZ7 (4)	0.156333	**0.177333**	WFG7 (4)	0.004333	**0.013667**
DTLZ1 (5)	**0.993667**	0.043000	WFG1 (5)	0.000000	0.000000
DTLZ2 (5)	0.000000	**0.062333**	WFG2 (5)	0.142000	**0.805333**
DTLZ3 (5)	**0.537000**	0.518333	WFG3 (5)	0.145667	**0.487000**
DTLZ4 (5)	0.000000	**0.075333**	WFG4 (5)	0.000000	**0.243000**
DTLZ5 (5)	0.128000	**0.461000**	WFG5 (5)	0.000000	**0.056667**
DTLZ6 (5)	0.011333	**0.987000**	WFG6 (5)	**0.311000**	0.084667
DTLZ7 (5)	0.059667	**0.096333**	WFG7 (5)	**0.004000**	0.000000
DTLZ1 (6)	**0.915667**	0.711333	WFG1 (6)	0.000000	0.000000
DTLZ2 (6)	0.000000	**0.051333**	WFG2 (6)	0.406333	**0.599667**
DTLZ3 (6)	0.080667	**0.943000**	WFG3 (6)	0.128000	**0.375333**
DTLZ4 (6)	0.000000	**0.069333**	WFG4 (6)	0.000000	**0.105000**
DTLZ5 (6)	0.048333	**0.604000**	WFG5 (6)	0.000000	**0.023333**
DTLZ6 (6)	0.001333	**0.988333**	WFG6 (6)	**0.315333**	0.064333
DTLZ7 (6)	**0.202333**	0.101333	WFG7 (6)	0.000000	0.000000

TABLE III. RESULTS OBTAINED IN THE DTLZ AND WFG TEST PROBLEMS BY GD-MOEA AND OUR GDE-MOEA, USING THE TWO SET COVERAGE INDICATOR (I_{SC}). $\mathcal{A}$ IS THE SET COMPOSED BY ALL SOLUTIONS FOUND BY GD-MOEA CONSIDERING ALL 30 INDEPENDENT RUNS AND $\mathcal{B}$ IS THE SET COMPOSED BY ALL SOLUTIONS FOUND BY GDE-MOEA CONSIDERING ALL 30 INDEPENDENT RUNS.

VIII. CONCLUSIONS AND FUTURE WORK

We have proposed a new selection mechanism based on ϵ-dominance and we called it "ϵ-selection". This mechanism has two aims: the first is to achieve that a MOEA can explore the whole search space at the beginning of the search process and, the second is to improve the distribution of the solutions along the approximation of the Pareto front towards the end of the search. An interesting feature of our ϵ-selection mechanism is that the user does not need to set the value of ϵ. Although it seems that our strategy to set ϵ can affect the efficiency of our GDE-MOEA because we need to iterate until selecting the total number of individuals, in practice, we could see that our GDE-MOEA was faster than GD-MOEA

$\vec{f}$	gd-moea I_H	gde-moea I_H	$P(H)$
DTLZ1 (3)	**1.0842 (0.005)**	1.0622 (0.017)	0.000 (1)
DTLZ2 (3)	0.7197 (0.006)	**0.7527 (0.003)**	0.000 (1)
DTLZ3 (3)	**1.3278 (0.002)**	1.3264 (0.005)	0.251 (0)
DTLZ4 (3)	0.8347 (0.008)	**0.8650 (0.002)**	0.000 (1)
DTLZ5 (3)	0.6638 (0.015)	**0.6686 (0.003)**	0.706 (0)
DTLZ6 (3)	1.1250 (0.011)	**1.1292 (0.004)**	0.206 (0)
DTLZ7 (3)	0.5078 (0.063)	**0.5100 (0.064)**	0.428 (0)
DTLZ1 (4)	**1.2077 (0.199)**	0.2282 (0.443)	0.000 (1)
DTLZ2 (4)	**0.9087 (0.013)**	0.7387 (0.077)	0.000 (1)
DTLZ3 (4)	1.4608 (0.006)	**1.4626 (0.003)**	0.035 (1)
DTLZ4 (4)	**0.9027 (0.015)**	0.8357 (0.049)	0.000 (1)
DTLZ5 (4)	1.1401 (0.018)	**1.1761 (0.004)**	0.000 (1)
DTLZ6 (4)	1.2517 (0.036)	**1.3606 (0.007)**	0.000 (1)
DTLZ7 (4)	0.5328 (0.038)	**0.5612 (0.046)**	0.000 (1)
DTLZ1 (5)	**0.4102 (0.490)**	0.0000 (0.000)	0.000 (1)
DTLZ2 (5)	**1.0680 (0.027)**	0.7464 (0.092)	0.000 (1)
DTLZ3 (5)	1.5942 (0.013)	**1.6085 (0.001)**	0.000 (1)
DTLZ4 (5)	**1.0309 (0.017)**	0.9261 (0.059)	0.000 (1)
DTLZ5 (5)	1.2534 (0.088)	**1.3313 (0.021)**	0.000 (1)
DTLZ6 (5)	1.1999 (0.067)	**1.4741 (0.017)**	0.000 (1)
DTLZ7 (5)	0.4457 (0.046)	**0.4809 (0.022)**	0.000 (1)
DTLZ1 (6)	**0.0150 (0.049)**	0.0000 (0.000)	0.041 (1)
DTLZ2 (6)	**1.2215 (0.027)**	0.8923 (0.117)	0.000 (1)
DTLZ3 (6)	1.7127 (0.049)	**1.7685 (0.002)**	0.000 (1)
DTLZ4 (6)	**1.2839 (0.059)**	1.1521 (0.144)	0.000 (1)
DTLZ5 (6)	0.9733 (0.115)	**1.1284 (0.076)**	0.000 (1)
DTLZ6 (6)	0.9630 (0.039)	**1.5303 (0.078)**	0.000 (1)
DTLZ7 (6)	0.5515 (0.061)	**0.6250 (0.020)**	0.000 (1)

(a)

moead-pbi I_H	gde-moea I_H	$P(H)$
1.0710 (0.003)	1.0622 (0.017)	0.228 (0)
0.7146 (0.000)	**0.7527 (0.003)**	0.000 (1)
1.3128 (0.001)	**1.3264 (0.005)**	0.000 (1)
0.8194 (0.000)	**0.8650 (0.002)**	0.000 (1)
0.6190 (0.004)	**0.6686 (0.003)**	0.000 (1)
1.0210 (0.011)	**1.1292 (0.004)**	0.000 (1)
0.4506 (0.027)	**0.5100 (0.064)**	0.000 (1)
1.1858 (0.005)	0.2282 (0.443)	0.000 (1)
0.8653 (0.001)	0.7387 (0.077)	0.000 (1)
1.4555 (0.001)	**1.4626 (0.003)**	0.000 (1)
0.8598 (0.001)	0.8357 (0.049)	0.000 (1)
1.0325 (0.026)	**1.1761 (0.004)**	0.000 (1)
1.2891 (0.006)	**1.3606 (0.007)**	0.000 (1)
0.3430 (0.007)	**0.5612 (0.046)**	0.000 (1)
1.2463 (0.011)	0.0000 (0.000)	0.000 (1)
0.9644 (0.004)	0.7464 (0.092)	0.000 (1)
1.5816 (0.008)	**1.6085 (0.001)**	0.000 (1)
0.9328 (0.003)	0.9261 (0.059)	0.011 (1)
1.1882 (0.022)	**1.3313 (0.021)**	0.000 (1)
1.4415 (0.012)	**1.4741 (0.017)**	0.000 (1)
0.0985 (0.067)	**0.4809 (0.022)**	0.000 (1)
1.3059 (0.013)	0.0000 (0.000)	0.000 (1)
1.0007 (0.011)	0.8923 (0.117)	0.000 (1)
1.7610 (0.005)	**1.7685 (0.002)**	0.000 (1)
1.0289 (0.007)	**1.1521 (0.144)**	0.000 (1)
0.9985 (0.018)	**1.1284 (0.076)**	0.000 (1)
1.5454 (0.017)	1.5303 (0.078)	0.233 (0)
0.0260 (0.006)	**0.6250 (0.020)**	0.000 (1)

(b)

sms-emoa-hype I_H	gde-moea I_H	$P(H)$
1.1011 (0.006)	1.0622 (0.017)	0.000 (1)
0.7482 (0.002)	**0.7527 (0.003)**	0.000 (1)
1.3298 (0.000)	1.3264 (0.005)	0.000 (1)
0.8642 (0.002)	**0.8650 (0.002)**	0.074 (0)
0.6784 (0.000)	0.6686 (0.003)	0.000 (1)
1.1237 (0.013)	**1.1292 (0.004)**	0.355 (0)
0.5419 (0.035)	0.5100 (0.064)	0.006 (1)
1.2586 (0.057)	0.2282 (0.443)	0.000 (1)
1.0144 (0.003)	0.7387 (0.077)	0.000 (1)
1.4636 (0.000)	1.4626 (0.003)	0.004 (1)
1.0153 (0.004)	0.8357 (0.049)	0.000 (1)
1.1826 (0.002)	1.1761 (0.004)	0.000 (1)
1.3577 (0.005)	**1.3606 (0.007)**	0.013 (1)
0.5309 (0.038)	**0.5612 (0.046)**	0.000 (1)
1.2371 (0.348)	0.0000 (0.000)	0.000 (1)
1.2793 (0.005)	0.7464 (0.092)	0.000 (1)
1.6087 (0.000)	1.6085 (0.001)	0.371 (0)
1.2567 (0.005)	0.9261 (0.059)	0.000 (1)
1.3891 (0.002)	1.3313 (0.021)	0.000 (1)
1.5292 (0.004)	1.4741 (0.017)	0.000 (1)
0.5017 (0.048)	0.4809 (0.022)	0.001 (1)
1.5008 (0.235)	0.0000 (0.000)	0.000 (1)
1.5706 (0.005)	0.8923 (0.117)	0.000 (1)
1.7710 (0.000)	1.7685 (0.002)	0.000 (1)
1.6172 (0.003)	1.1521 (0.144)	0.000 (1)
1.3218 (0.005)	1.1284 (0.076)	0.000 (1)
1.6757 (0.006)	1.5303 (0.078)	0.000 (1)
0.4905 (0.117)	**0.6250 (0.020)**	0.000 (1)

(c)

$\vec{f}$	gd-moea I_H	gde-moea I_H	$P(H)$
WFG1 (3)	0.8516 (0.037)	**0.9095 (0.054)**	0.000(1)
WFG2 (3)	0.5334 (0.124)	**0.6643 (0.104)**	0.000(1)
WFG3 (3)	0.5987 (0.008)	**0.6101 (0.003)**	0.000(1)
WFG4 (3)	0.6562 (0.008)	**0.7454 (0.002)**	0.000(1)
WFG5 (3)	0.5310 (0.004)	**0.5362 (0.003)**	0.000(1)
WFG6 (3)	0.5248 (0.008)	**0.5447 (0.002)**	0.000(1)
WFG7 (3)	0.6644 (0.012)	**0.6989 (0.010)**	0.000(1)
WFG1 (4)	0.6518 (0.115)	**0.8576 (0.056)**	0.000(1)
WFG2 (4)	0.1651 (0.189)	**0.5304 (0.126)**	0.000(1)
WFG3 (4)	0.4531 (0.025)	**0.5692 (0.012)**	0.000(1)
WFG4 (4)	**0.8266 (0.015)**	0.7492 (0.041)	0.000(1)
WFG5 (4)	0.5302 (0.006)	**0.5585 (0.005)**	0.000(1)
WFG6 (4)	0.3600 (0.044)	**0.5087 (0.021)**	0.000(1)
WFG7 (4)	**0.8191 (0.019)**	0.5904 (0.060)	0.000(1)
WFG1 (5)	0.4915 (0.039)	**0.7824 (0.049)**	0.000(1)
WFG2 (5)	0.1603 (0.146)	**0.5505 (0.160)**	0.000(1)
WFG3 (5)	0.2793 (0.066)	**0.5263 (0.032)**	0.000(1)
WFG4 (5)	**0.8997 (0.030)**	0.7646 (0.063)	0.000(1)
WFG5 (5)	0.4703 (0.019)	**0.5415 (0.047)**	0.000(1)
WFG6 (5)	0.2704 (0.053)	**0.4385 (0.042)**	0.000(1)
WFG7 (5)	**0.8501 (0.022)**	0.3793 (0.071)	0.000(1)
WFG1 (6)	0.5715 (0.043)	**0.7742 (0.088)**	0.000(1)
WFG2 (6)	0.1217 (0.145)	**0.2751 (0.212)**	0.001(1)
WFG3 (6)	0.1443 (0.055)	**0.4780 (0.049)**	0.000(1)
WFG4 (6)	**0.9290 (0.035)**	0.8766 (0.050)	0.000(1)
WFG5 (6)	0.3477 (0.047)	**0.5064 (0.071)**	0.000(1)
WFG6 (6)	0.2171 (0.053)	**0.3569 (0.060)**	0.000(1)
WFG7 (6)	**0.8067 (0.036)**	0.3271 (0.067)	0.000(1)

(d)

moead-pbi I_H	gde-moea I_H	$P(H)$
0.9172 (0.015)	0.9095 (0.054)	0.011 (1)
0.1610 (0.204)	**0.6643 (0.104)**	0.000 (1)
0.4993 (0.026)	**0.6101 (0.003)**	0.000 (1)
0.6031 (0.013)	**0.7454 (0.002)**	0.000 (1)
0.4732 (0.010)	**0.5362 (0.003)**	0.000 (1)
0.4545 (0.007)	**0.5447 (0.002)**	0.000 (1)
0.5025 (0.057)	**0.6989 (0.010)**	0.000 (1)
1.1003 (0.040)	0.8576 (0.056)	0.000 (1)
0.0041 (0.022)	**0.5304 (0.126)**	0.000 (1)
0.2888 (0.035)	**0.5692 (0.012)**	0.000 (1)
0.6749 (0.026)	**0.7492 (0.041)**	0.000 (1)
0.3724 (0.016)	**0.5585 (0.005)**	0.000 (1)
0.2902 (0.016)	**0.5087 (0.021)**	0.000 (1)
0.3354 (0.043)	**0.5904 (0.060)**	0.000 (1)
1.1742 (0.041)	0.7824 (0.049)	0.000 (1)
0.0173 (0.045)	**0.5505 (0.160)**	0.000 (1)
0.1587 (0.039)	**0.5263 (0.032)**	0.000 (1)
0.6687 (0.025)	**0.7646 (0.063)**	0.000 (1)
0.2432 (0.014)	**0.5415 (0.047)**	0.000 (1)
0.2492 (0.014)	**0.4385 (0.042)**	0.000 (1)
0.2560 (0.017)	**0.3793 (0.071)**	0.000 (1)
1.0998 (0.013)	0.7742 (0.088)	0.000 (1)
0.0067 (0.028)	**0.2751 (0.212)**	0.000 (1)
0.1348 (0.050)	**0.4780 (0.049)**	0.000 (1)
0.5984 (0.029)	**0.8766 (0.050)**	0.000 (1)
0.1633 (0.018)	**0.5064 (0.071)**	0.000 (1)
0.2354 (0.020)	**0.3569 (0.060)**	0.000 (1)
0.2108 (0.015)	**0.3271 (0.067)**	0.000 (1)

(e)

sms-emoa-hype I_H	gde-moea I_H	$P(H)$
0.9888 (0.049)	0.9095 (0.054)	0.000 (1)
0.6581 (0.053)	**0.6643 (0.104)**	0.153 (0)
0.6066 (0.007)	**0.6101 (0.003)**	0.014 (1)
0.7114 (0.005)	**0.7454 (0.002)**	0.000 (1)
0.5395 (0.003)	0.5362 (0.003)	0.000 (1)
0.5479 (0.004)	0.5447 (0.002)	0.000 (1)
0.5717 (0.029)	**0.6989 (0.010)**	0.000 (1)
1.1455 (0.028)	0.8576 (0.056)	0.000 (1)
0.5083 (0.213)	**0.5304 (0.126)**	0.830 (0)
0.5381 (0.016)	**0.5692 (0.012)**	0.000 (1)
0.9615 (0.008)	0.7492 (0.041)	0.000 (1)
0.5660 (0.005)	0.5585 (0.005)	0.000 (1)
0.5669 (0.010)	0.5087 (0.021)	0.000 (1)
0.4874 (0.037)	**0.5904 (0.060)**	0.000 (1)
1.2570 (0.031)	0.7824 (0.049)	0.000 (1)
0.5071 (0.240)	**0.5505 (0.160)**	0.652 (0)
0.5193 (0.029)	**0.5263 (0.032)**	0.332 (0)
1.1509 (0.019)	0.7646 (0.063)	0.000 (1)
0.5915 (0.011)	0.5415 (0.047)	0.000 (1)
0.5649 (0.016)	0.4385 (0.042)	0.000 (1)
0.3723 (0.023)	**0.3793 (0.071)**	0.841 (0)
1.3561 (0.031)	0.7742 (0.088)	0.000 (1)
0.4691 (0.260)	0.2751 (0.212)	0.002 (1)
0.5209 (0.033)	0.4780 (0.049)	0.000 (1)
1.2922 (0.025)	0.8766 (0.050)	0.000 (1)
0.6043 (0.015)	0.5064 (0.071)	0.000 (1)
0.5646 (0.027)	0.3569 (0.060)	0.000 (1)
0.3183 (0.017)	**0.3271 (0.067)**	0.970 (0)

(f)

TABLE I. RESULTS OBTAINED IN THE DTLZ AND WFG TEST PROBLEMS. WE COMPARE OUR GDE-MOEA WITH RESPECT TO GD-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.

and SMS-EMOA-HYPE. Therefore, we can say that we address the main disadvantage of the selection mechanisms based on ϵ-dominance. Our ϵ-selection mechanism has a linear complexity with respect to the number of objective

$\vec{f}$	gd-moea time	gde-moea time
DTLZ1 (3)	0.7970 (0.033)	**0.5000 (0.049)**
DTLZ2 (3)	1.1680 (0.027)	**0.5133 (0.009)**
DTLZ3 (3)	0.6343 (0.026)	**0.4830 (0.031)**
DTLZ4 (3)	1.1877 (0.031)	**0.5410 (0.007)**
DTLZ5 (3)	1.0603 (0.054)	**0.6017 (0.018)**
DTLZ6 (3)	0.8973 (0.044)	**0.4783 (0.016)**
DTLZ7 (3)	0.9127 (0.052)	**0.5320 (0.012)**
DTLZ1 (4)	0.8627 (0.047)	**0.7510 (0.050)**
DTLZ2 (4)	1.1453 (0.041)	**0.6033 (0.031)**
DTLZ3 (4)	0.8247 (0.053)	**0.5550 (0.037)**
DTLZ4 (4)	1.1830 (0.054)	**0.6507 (0.043)**
DTLZ5 (4)	1.1030 (0.042)	**0.6080 (0.017)**
DTLZ6 (4)	1.4060 (0.109)	**0.4997 (0.019)**
DTLZ7 (4)	0.9217 (0.035)	**0.6197 (0.038)**
DTLZ1 (5)	0.9467 (0.047)	**0.6330 (0.051)**
DTLZ2 (5)	1.1317 (0.014)	**0.7160 (0.066)**
DTLZ3 (5)	1.0660 (0.087)	**0.6430 (0.053)**
DTLZ4 (5)	1.1523 (0.022)	**0.7703 (0.043)**
DTLZ5 (5)	1.1407 (0.055)	**0.7240 (0.048)**
DTLZ6 (5)	1.7253 (0.041)	**0.6187 (0.030)**
DTLZ7 (5)	1.0603 (0.097)	**0.6920 (0.015)**
DTLZ1 (6)	1.4997 (0.284)	**0.7063 (0.021)**
DTLZ2 (6)	1.1307 (0.065)	**0.7850 (0.022)**
DTLZ3 (6)	1.5993 (0.152)	**0.7353 (0.061)**
DTLZ4 (6)	1.2000 (0.030)	**0.9027 (0.068)**
DTLZ5 (6)	1.3610 (0.106)	**0.7990 (0.089)**
DTLZ6 (6)	1.7547 (0.213)	**0.7507 (0.038)**
DTLZ7 (6)	1.3053 (0.073)	**0.7590 (0.012)**

(a)

moead-pbi time	gde-moea time
0.4993 (0.016)	0.5000 (0.049)
0.5783 (0.010)	**0.5133 (0.009)**
0.5195 (0.012)	**0.4830 (0.031)**
0.6037 (0.008)	**0.5410 (0.007)**
0.5922 (0.007)	0.6017 (0.018)
0.5007 (0.018)	**0.4783 (0.016)**
0.5397 (0.008)	**0.5320 (0.012)**
0.5230 (0.008)	0.7510 (0.050)
0.6147 (0.012)	**0.6033 (0.031)**
0.5533 (0.020)	0.5550 (0.037)
0.6440 (0.011)	0.6507 (0.043)
0.6128 (0.009)	**0.6080 (0.017)**
0.5351 (0.010)	**0.4997 (0.019)**
0.5860 (0.008)	0.6197 (0.038)
0.5532 (0.005)	0.6330 (0.051)
0.6453 (0.010)	0.7160 (0.066)
0.5785 (0.012)	0.6430 (0.053)
0.6949 (0.004)	0.7703 (0.043)
0.6455 (0.004)	0.7240 (0.048)
0.5784 (0.008)	0.6187 (0.030)
0.6289 (0.004)	0.6920 (0.015)
0.5816 (0.011)	0.7063 (0.021)
0.6750 (0.003)	0.7850 (0.022)
0.6162 (0.017)	0.7353 (0.061)
0.7485 (0.003)	0.9027 (0.068)
0.6683 (0.011)	0.7990 (0.089)
0.6308 (0.006)	0.7507 (0.038)
0.6589 (0.012)	0.7590 (0.012)

(b)

sms-emoa-hype time	gde-moea time
47.0000 (2.620)	**0.5000 (0.049)**
106.1333 (4.105)	**0.5133 (0.009)**
135.9667 (21.629)	**0.4830 (0.031)**
107.1667 (3.822)	**0.5410 (0.007)**
64.3333 (5.430)	**0.6017 (0.018)**
59.0667 (9.747)	**0.4783 (0.016)**
98.4333 (9.106)	**0.5320 (0.012)**
59.6667 (3.280)	**0.7510 (0.050)**
156.0333 (6.555)	**0.6033 (0.031)**
165.9333 (18.995)	**0.5550 (0.037)**
157.2667 (9.602)	**0.6507 (0.043)**
143.1667 (4.796)	**0.6080 (0.017)**
129.1000 (7.648)	**0.4997 (0.019)**
185.6667 (16.067)	**0.6197 (0.038)**
79.1333 (5.632)	**0.6330 (0.051)**
188.3333 (8.231)	**0.7160 (0.066)**
177.1000 (24.347)	**0.6430 (0.053)**
190.5000 (6.845)	**0.7703 (0.043)**
229.3333 (14.328)	**0.7240 (0.048)**
225.3667 (11.056)	**0.6187 (0.030)**
296.9333 (23.678)	**0.6920 (0.015)**
98.9333 (6.904)	**0.7063 (0.021)**
233.3667 (11.182)	**0.7850 (0.022)**
185.3000 (22.371)	**0.7353 (0.061)**
234.6333 (10.581)	**0.9027 (0.068)**
336.9000 (18.293)	**0.7990 (0.089)**
340.4333 (16.669)	**0.7507 (0.038)**
377.9000 (42.232)	**0.7590 (0.012)**

(c)

$\vec{f}$	gd-moea time	gde-moea time
WFG1 (3)	1.5650 (0.044)	**1.2473 (0.076)**
WFG2 (3)	1.4520 (0.129)	**1.1443 (0.109)**
WFG3 (3)	1.4500 (0.088)	**1.0930 (0.064)**
WFG4 (3)	2.1980 (0.026)	**1.0647 (0.054)**
WFG5 (3)	2.0107 (0.018)	**1.2197 (0.143)**
WFG6 (3)	1.5693 (0.022)	**1.0500 (0.041)**
WFG7 (3)	2.4217 (0.019)	**1.3157 (0.110)**
WFG1 (4)	1.3950 (0.065)	**1.3227 (0.040)**
WFG2 (4)	1.6833 (0.279)	**1.4180 (0.120)**
WFG3 (4)	1.4767 (0.197)	**1.1403 (0.054)**
WFG4 (4)	2.3590 (0.019)	**1.2107 (0.075)**
WFG5 (4)	1.8057 (0.026)	**1.2380 (0.143)**
WFG6 (4)	1.2520 (0.015)	**1.2023 (0.100)**
WFG7 (4)	2.4613 (0.019)	**1.4623 (0.133)**
WFG1 (5)	1.7730 (0.156)	**1.5307 (0.090)**
WFG2 (5)	1.5060 (0.034)	**1.5017 (0.141)**
WFG3 (5)	1.7977 (0.117)	**1.4197 (0.195)**
WFG4 (5)	2.4873 (0.022)	**1.3110 (0.053)**
WFG5 (5)	1.4357 (0.020)	**1.3887 (0.137)**
WFG6 (5)	**1.3510 (0.014)**	1.5437 (0.140)
WFG7 (5)	2.5517 (0.058)	**2.0677 (0.224)**
WFG1 (6)	**1.7980 (0.206)**	1.8623 (0.189)
WFG2 (6)	1.9143 (0.197)	**1.5703 (0.163)**
WFG3 (6)	**1.6420 (0.278)**	1.7260 (0.140)
WFG4 (6)	2.5670 (0.032)	**1.7070 (0.158)**
WFG5 (6)	**1.3950 (0.017)**	1.7223 (0.226)
WFG6 (6)	1.4527 (0.027)	**1.3977 (0.284)**
WFG7 (6)	2.6047 (0.037)	**2.3543 (0.211)**

(d)

moead-pbi time	gde-moea time
1.1427 (0.019)	1.2473 (0.076)
0.9272 (0.024)	1.1443 (0.109)
0.9738 (0.018)	1.0930 (0.064)
0.9919 (0.007)	1.0647 (0.054)
0.9594 (0.007)	1.2197 (0.143)
0.9478 (0.010)	1.0500 (0.041)
1.1988 (0.026)	1.3157 (0.110)
1.1697 (0.017)	1.3227 (0.040)
0.9473 (0.021)	1.4180 (0.120)
1.0207 (0.011)	1.1403 (0.054)
1.0258 (0.009)	1.2107 (0.075)
0.9848 (0.009)	1.2380 (0.143)
0.9774 (0.007)	1.2023 (0.100)
1.2529 (0.014)	1.4623 (0.133)
1.2474 (0.015)	1.5307 (0.090)
1.0083 (0.020)	1.5017 (0.141)
1.0908 (0.010)	1.4197 (0.195)
1.1067 (0.005)	1.3110 (0.053)
1.0683 (0.006)	1.3887 (0.137)
1.0342 (0.024)	1.5437 (0.140)
1.4166 (0.021)	2.0677 (0.224)
1.3214 (0.012)	1.8623 (0.189)
1.0430 (0.021)	1.5703 (0.163)
1.1115 (0.011)	1.7260 (0.140)
1.1695 (0.009)	1.7070 (0.158)
1.1185 (0.009)	1.7223 (0.226)
1.0602 (0.024)	1.3977 (0.284)
1.8199 (0.145)	2.3543 (0.211)

(e)

sms-emoa-hype time	gde-moea time
147.0000 (3.670)	**1.2473 (0.076)**
98.4333 (6.786)	**1.1443 (0.109)**
148.7333 (3.941)	**1.0930 (0.064)**
107.5000 (4.233)	**1.0647 (0.054)**
153.0667 (8.246)	**1.2197 (0.143)**
168.9333 (8.330)	**1.0500 (0.041)**
151.5667 (6.530)	**1.3157 (0.110)**
233.7333 (8.434)	**1.3227 (0.040)**
170.6333 (11.232)	**1.4180 (0.120)**
247.1000 (7.939)	**1.1403 (0.054)**
157.8333 (6.455)	**1.2107 (0.075)**
206.7667 (19.689)	**1.2380 (0.143)**
216.9667 (17.647)	**1.2023 (0.100)**
252.1333 (8.429)	**1.4623 (0.133)**
335.0667 (7.607)	**1.5307 (0.090)**
269.5667 (20.717)	**1.5017 (0.141)**
378.1667 (6.362)	**1.4197 (0.195)**
220.6667 (13.553)	**1.3110 (0.053)**
276.2000 (31.841)	**1.3887 (0.137)**
274.2667 (47.308)	**1.5437 (0.140)**
358.9667 (10.005)	**2.0677 (0.224)**
383.8000 (42.576)	**1.8623 (0.189)**
377.4333 (29.319)	**1.5703 (0.163)**
445.7333 (46.018)	**1.7260 (0.140)**
316.2000 (12.098)	**1.7070 (0.158)**
246.7000 (6.435)	**1.7223 (0.226)**
259.2333 (5.024)	**1.3977 (0.284)**
408.2667 (40.609)	**2.3543 (0.211)**

(f)

TABLE II. TIME REQUIRED (IN SECONDS) BY GD-MOEA, MOEA/D, SMS-EMOA-HYPE AND OUR PROPOSED GDE-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.

functions. Therefore, it is suitable for solving many-objective optimization problems. We decided to incorporate our ϵ-selection into the GD-MOEA algorithm giving rise to a new MOEA called "Generational Distance & ϵ-dominance Multi-Objective Evolutionary Algorithm (GDE-MOEA)". Our preliminary results indicate that our GDE-MOEA outperforms both the original GD-MOEA and MOEA/D. Also, 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) but GDE-

MOEA is 189.67 times faster than SMS-EMOA-HYPE. For these reasons, we can say that our proposed GDE-MOEA is a good option to solve MOPs having both low and high dimensionality in objective function space, if we consider both the quality in the approximation of the Pareto optimal set and the running time required to obtain it.

As part of our future work, we want to improve the technique used to avoid selecting weakly dominated solutions. This is because in some cases GDE-MOEA chooses a weakly dominated solution instead of a nondominated solution. For example, we noted that our GDE-MOEA had serious difficulties to solve the DTLZ1 test problem and it is well-known that when we solve this test problem many weakly dominated solutions are generated during the search process. Also, we want to extend our experimental study, e.g., we want to use MOPs with more than 6 objective functions and other standard test functions.

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. Fleischer, "The Measure of Pareto Optima. Applications to Multi-objective Metaheuristics," 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. 519–533, Springer. Lecture Notes in Computer Science. Volume 2632, April 2003.

[4] J. Knowles and D. Corne, "Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 100–116, April 2003.

[5] E. Zitzler and S. Künzli, "Indicator-based Selection in Multiobjective Search," in *Parallel Problem Solving from Nature - PPSN VIII* (X. Y. et al., ed.), (Birmingham, UK), pp. 832–842, Springer-Verlag. Lecture Notes in Computer Science Vol. 3242, September 2004.

[6] 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.

[7] C. Igel, N. Hansen, and S. Roth, "Covariance Matrix Adaptation for Multi-objective Optimization," *Evolutionary Computation*, vol. 15, pp. 1–28, Spring 2007.

[8] 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.

[9] 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.

[10] D. Brockhoff, T. Wagner, and H. Trautmann, "On the Properties of the $R2$ Indicator," in *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, (Philadelphia, USA), pp. 465–472, ACM Press, July 2012. ISBN: 978-1-4503-1177-9.

[11] H. Trautmann, T. Wagner, and D. Brockhoff, "R2-emoa: Focused multiobjective search using r2-indicator-based selection," in *Learning and Intelligent Optimization* (G. Nicosia and P. Pardalos, eds.), Lecture Notes in Computer Science, pp. 70–74, Springer Berlin Heidelberg, 2013.

[12] R. Hernández Gómez and C. A. Coello Coello, "MOMBI: A New Metaheuristic for Many-Objective Optimization Based on the $R2$ Indicator," in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, (Cancún, México), pp. 2488–2495, IEEE Press, 20-23 June 2013. ISBN 978-1-4799-0454-9.

[13] D. ung H. Phan and J. Suzuki, "R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization," in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, (Cancún, México), pp. 1836–1845, IEEE Press, 20-23 June 2013. ISBN 978-1-4799-0454-9.

[14] T. Wagner, H. Trautmann, and D. Brockhoff, "Preference Articulation by Means of the $R2$ Indicator," in *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013* (R. C. Purshouse, P. J. Fleming, C. M. Fonseca, S. Greco, and J. Shaw, eds.), pp. 81–95, Sheffield, UK: Springer. Lecture Notes in Computer Science Vol. 7811, March 19-22 2013.

[15] 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.

[16] O. Schütze, X. Esquivel, A. Lara, and C. A. Coello Coello, "Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, pp. 504–522, August 2012.

[17] K. Gerstl, G. Rudolph, O. Schütze, and H. Trautmann, "Finding Evenly Spaced Fronts for Multiobjective Control via Averaging Hausdorff-Measure," in *The 2011 8th International Conference on Electrical Engineering, Computer Science and Automatic Control (CCE'2011)*, (Mérida, Yucatán, México), pp. 975–980, IEEE Press, October 2011.

[18] C. A. Rodríguez Villalobos and C. A. Coello Coello, "A New Multi-Objective Evolutionary Algorithm Based on a Performance Assessment Indicator," in *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, (Philadelphia, USA), pp. 505–512, ACM Press, July 2012. ISBN: 978-1-4503-1177-9.

[19] C. Domínguez-Medina, G. Rudolph, O. Schüetze, and H. Trautmann, "Evenly Spaced Pareto Fronts of Quad-objective Problems using PSA Partitioning Technique," in *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, (Cancún, México), pp. 3190–3197, IEEE Press, 20-23 June 2013. ISBN 978-1-4799-0454-9.

[20] 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.

[21] 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.

[22] 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*, Springer. Lecture Notes in Computer Science Vol. 6576, April 2015.

[23] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, "Combining Convergence and Diversity in Evolutionary Multi-objective Optimization," *Evolutionary Computation*, vol. 10, pp. 263–282, Fall 2002.

[24] 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.

[25] 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.

[26] 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.

[27] 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.