GD-MOEA: A New Multi-Objective Evolutionary Algorithm based on the Generational Distance Indicator

Adriana Menchaca-Mendez and Carlos A. Coello Coello*

CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
México D.F. 07300, MÉXICO
`adriana.menchacamendez@gmail.com, ccoello@cs.cinvestav.mx`

Abstract. In this paper, we propose a new selection mechanism for Multi-Objective Evolutionary Algorithms (MOEAs), which is based on the generational distance indicator and uses a technique that relies on Euclidean distances to maintain diversity in the population (in objective function space). Our proposed selecion mechanism is incorporated into a MOEA which adopts the operators of NSGA-II (crossover and mutation) to generate new individuals. The new MOEA is called "Generational Distance - Multi-Objective Evolutionary Algorithm (GD-MOEA)." Our GD-MOEA is validated using standard test problems taken from the specialized literature, having three to six objective functions. GD-MOEA is compared with respect to MOEA/D using Penalty Boundary Intersection (PBI), which is based on decomposition, and to 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 if we consider both quality in the solutions and the running time required to generate them, our GD-MOEA is a good alternative to solve multi-objective optimization problems having both low dimensionality and high dimensionality in objective function space.

1 Introduction

Many real-world applications involve the solution of problems that have multiple (conflicting) objective functions which have to be simultaneously optimized. These are the so-called "Multi-objective Optimization Problems (MOPs)". Since their objective functions are in conflict with each other, the notion of optimality refers to finding the best possible trade-offs among the objective functions. Consequently, there is no single optimal solution but a set of solutions, which is called *Pareto optimal set*, whose image is known as the *Pareto front*. Since the use of mathematical programming techniques to solve MOPs has several limitations, the use of evolutionary algorithms has become very popular in this area

---

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

in recent years, giving rise to the so-called Multi-Objective Evolutionary Algorithms (MOEAs) [6]. MOEAs have two main goals: (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 different indicators to assess the quality of the approximation of the Pareto optimal set generated by a MOEA, e.g., error ratio, generational distance, inverted generational distance, spacing, hypervolume, $R2$-indicator, Δ_p-indicator, ϵ-indicator, two set coverage, etc. [6]. However, very few performance indicators are "Pareto Compliant".[1] In recent years, MOEAs based on indicators have become popular because the use of Pareto-based selection has several limitations. Perhaps, the most remarkable is its poor scalability regarding the number of objective functions of a MOP.[2]

MOEAs based on the hypervolume indicator (I_H) have been relatively popular (see for example [16, 28, 2, 15, 17]) mainly because I_H is the only unary indicator which is known to be "Pareto compliant" [29]. However, I_H has an important disadvantage: its high computational cost (the problem of computing I_H is *NP-hard* [3]). Therefore, this type of MOEAs is impractical when we want to solve MOPs having four or more objective functions. On the other hand, after the study on the properties of the $R2$-indicator (I_{R2}) presented by Brockhoff et al. [4], a number of proposals of MOEAs based on I_{R2} have been introduced [21, 13, 22, 26]. Although I_{R2}-based MOEAs can solve MOPs with many objective functions at an affordable computational cost, this type of algorithms also has an important disadvantage: 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 the well-known MOEA/D [27] which decomposes the MOP into N scalar optimization subproblems and solves them simultaneously using an evolutionary algorithm. Recently, the Δ_p-indicator (I_{Δ_p}) was introduced [20] and some MOEAs based on it have already been proposed [12, 19, 10]. The Δ_p-indicator is composed of slight modifications of two well-known indicators: generational distance (I_{GD}) [23] and inverted generational distance (I_{IGD}) [5]. It is well-known that for computing I_{GD} and I_{IGD}, it is necesary to know the true Pareto front. Therefore, the most important disadvantage of MOEAs based on I_{Δ_p} is perhaps that they need a reference set which must contain well-distributed solutions. Not being able to produce a good reference set could produce a diversity loss in the population which might cause that the algorithm cannot generate the complete Pareto front, or that it generates poorly distributed solutions. In extreme cases, the lack of an appropriate reference set could prevent convergence.

---

[1] Let Ω be the set of all feasible solutions and $\mathcal{A}$ and $\mathcal{B}$ two approximations of the Pareto optimal set, such that, $\mathcal{A} \preceq B$ denotes that every point $\boldsymbol{b} \in \mathcal{B}$ is weakly dominated by at least one point $\boldsymbol{a} \in \mathcal{A}$. An indicator $I : \Omega \to \mathbb{R}$ is **Pareto compliant** if for all $\mathcal{A}, \mathcal{B} \in \Omega : \mathcal{A} \preceq \mathcal{B} \Rightarrow I(\mathcal{A}) \geq I(\mathcal{B})$, assuming that greater indicator values correspond to higher quality.

[2] The quick increase in the number of non-dominated solutions as we increase the number of objective functions, rapidly dilutes the effect of the selection mechanism of a MOEA [11].

In this paper, we propose a new MOEA based on I_{GD} and we use the technique proposed in [18], which is based on Euclidean distances, to maintain diversity in objective function space. The idea is to use the non-dominated set produced at each generation as a reference set to calculate I_{GD}, even if it is not well-distributed, since at the beginning, the aim is to achieve convergence to the true Pareto front. Then, when we have produced many non-dominated solutions, the aim will be to improve their distribution. In this way, we can address the disadvantages of MOEAs based on I_H, I_{R2} and I_{Δ_p}: Our new selection mechanism has linear complexity with respect to the number of objective functions because computing the I_{GD} and maintaining diversity by means of computing Euclidean distances have linear complexity with respect to the number of objective functions. Further, it is not necessary to generate a set of well-distributed convex weights, and also, it is not necessary to generate a well-distributed reference set.

The remainder of this paper is organized as follows. Section 2 describes the generational distance indicator. The technique to maintain diversity in the population is described in Section 3. Our proposal is presented in Section 4. The experimental validation and the results obtained are shown in Section 5. Finally, we provide our conclusions and some possible paths for future work in Section 6.

2 Generational Distance Indicator

The generational distance indicator (I_{GD}) reports how far, on average, $\mathcal{A}$ is from $\mathcal{PF}$ [7, 24, 25], where $\mathcal{PF}$ is the true Pareto front and $\mathcal{A}$ is an approximation of the true Pareto front. I_{GD} is Pareto non-compliant and it is defined as:

$$I_{GD} = \frac{1}{|\mathcal{A}|}\left(\sum_{i=1}^{|\mathcal{A}|} d_i^p\right)^{\frac{1}{p}} \qquad (1)$$

where $|\mathcal{A}|$ is the number of vectors in $\mathcal{A}$, $p = 2$ and d_i is the Euclidean phenotypic distance between each member, i, of $\mathcal{A}$ and the closest member in $\mathcal{PF}$ to that member, i. If $I_{GD} = 0$, $\mathcal{A} \subseteq \mathcal{PF}$. Figure 1 shows how this indicator is calculated.

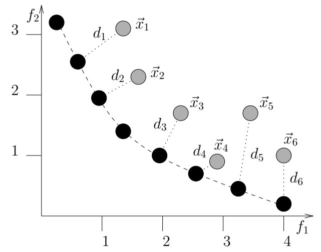


Fig. 1. The black points are the reference set. The approximation of the Pareto optimal set, $\mathcal{A}$, is composed by all point in gray. d_i is the Euclidean distance between $\boldsymbol{x}_i$ and its closest neighbor in $\mathcal{PF}$. Therefore: $I_{GD} = \frac{1}{6}\left(d_1^2 + d_2^2 + d_3^2 + d_4^2 + d_5^2 + d_6^2\right)^{\frac{1}{2}}$.

3 A distribution technique based on Euclidean distances

In [18], Menchaca and Coello proposed a technique based on Euclidean distances to improve the diversity in objective function space. This technique works as follows: Let's suppose that we have already a set of non-dominated solutions which we call "$\mathcal{S}$". If we want to improve its diversity using another set of non-dominated solutions which is called "$\mathcal{B}$", then, the solutions in $\mathcal{B}$ compete with the solutions in $\mathcal{S}$, considering that the size of $\mathcal{S}$ is fixed, as follows: For each solution $\boldsymbol{x} \in \mathcal{B}$, we obtain its nearest neighbor from $\mathcal{S}$, $\boldsymbol{x}_{near}$, and we choose a random individual from $\mathcal{S}$, $\boldsymbol{x}_{random}$ such that $\boldsymbol{x}_{near} \neq \boldsymbol{x}_{random}$, and then, these three solutions compete to survive. First, $\boldsymbol{x}$ competes with $\boldsymbol{x}_{random}$, if the Euclidean distance from $\boldsymbol{x}$ to its nearest neighbor in $\mathcal{S}$ is greater than the Euclidean distance from $\boldsymbol{x}_{random}$ to its nearest neighbor in $\mathcal{S}$, $\boldsymbol{x}$ replaces $\boldsymbol{x}_{random}$. If $\boldsymbol{x}$ loses the competition, $\boldsymbol{x}$ competes with its nearest neighbor to survive. If the Euclidean distance from $\boldsymbol{x}$ to its nearest neighbor in $\mathcal{S}$ (without considering $\boldsymbol{x}_{near}$) is greater than the Euclidean distance from $\boldsymbol{x}_{near}$ to its nearest neighbor in $\mathbb{S}$, then $\boldsymbol{x}$ replaces $\boldsymbol{x}_{near}$.

The authors mentioned that $\boldsymbol{x}_{near}$ is used with the idea of improving the diversity locally. If we move $\boldsymbol{x}_{near}$ to $\boldsymbol{x}$, do we increase the Euclidean distance from $\boldsymbol{x}_{near}$ to its nearest neighbor in $\mathcal{S}$? And, $\boldsymbol{x}_{rand}$ is used to avoid that solutions in unexplored regions are eliminated, e.g., if $\boldsymbol{x}$ and $\boldsymbol{x}_{near}$ are in an unexplored region, it is not good to delete one of the two solutions. Figure 2 illustrates how this technique works.

4 Our proposal

In this work, we propose a new selection mechanism for MOEAs. The idea is to use I_{GD} as a convergence strategy and to use the above distribution technique to maintain diversity in the population when many (even all) solutions are non-dominated. Our selection mechanism works as follows: If we want to select s individuals of a population $\mathcal{P}$, such that $s < \|\mathcal{P}\|$, we have to obtain the non-dominated individuals in $\mathcal{P}$ and put them in $\mathcal{S}$. The remaining individuals (dominated individuals) are placed in $\mathcal{B}$. If $s > \|\mathcal{S}\|$, we select the remaining r individuals (where $r = s - \|\mathcal{S}\|$) from $\mathcal{B}$ as follows: We calculate 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 non-dominated individual. After that, we have to sort $\mathcal{B}$ regarding d_i and we must create another set called "$\mathcal{S}' = \emptyset$". Finally, for each $\boldsymbol{x}_i \in \mathcal{B}$, we have to check 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, we must put $\boldsymbol{x}_i$ in $\mathcal{S}'$. If all individuals in $\mathcal{B}$ are considered and $\|\mathcal{S}'\| < r$, we must repeat the last process but now we will allow that only one individual in $\mathcal{S}'$ has the same neighbor that the individual that we want to select. We have to iterate until we obtain r individuals. Figure 3 shows how this procedure works.

If $s < \|\mathcal{S}\|$, we choose s individuals from $\mathcal{S}$ randomly. These individuals remain in $\mathcal{S}$ and we put the remaining non-dominated individuals in a new set

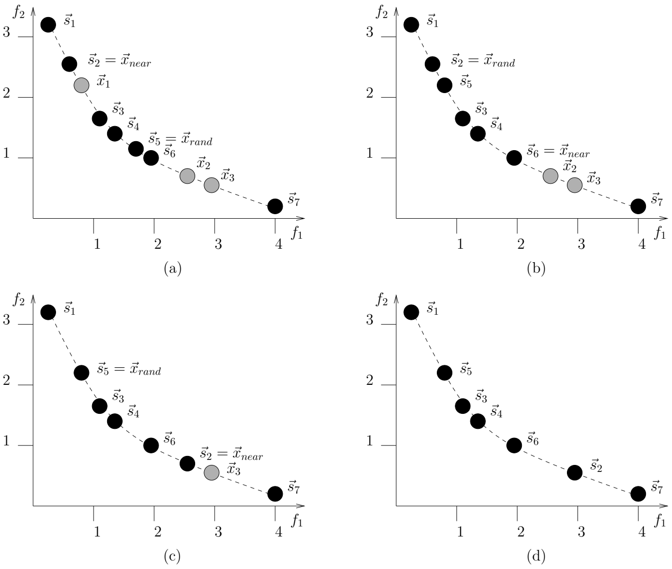


Fig. 2. All points in black are in $\mathcal{S}$ and all points in gray are in $\mathcal{B}$. In (a), we consider solution $\boldsymbol{x}_1$, its nearest neighbor in $\mathcal{S}$ is $\boldsymbol{s}_2$ and we choose $\boldsymbol{s}_5$ as a random solution. First, $\boldsymbol{x}_1$ competes with $\boldsymbol{s}_5$ and $\boldsymbol{s}_5$ loses because the distance from $\boldsymbol{x}_1$ to $\boldsymbol{s}_2$ is greater than the distance from $\boldsymbol{s}_5$ to $\boldsymbol{s}_6$; therefore, $\boldsymbol{x}_1$ replaces $\boldsymbol{s}_5$. In (b), we consider solution $\boldsymbol{x}_2$, its nearest neighbor in $\mathcal{S}$ is $\boldsymbol{s}_6$ and we choose $\boldsymbol{s}_2$ as a random solution. First, $\boldsymbol{x}_2$ and $\boldsymbol{s}_2$ compete and $\boldsymbol{x}_2$ wins because the distance from $\boldsymbol{x}_2$ to $\boldsymbol{s}_6$ is greater than the distance from $\boldsymbol{s}_2$ to $\boldsymbol{s}_5$; therefore, $\boldsymbol{x}_2$ replaces $\boldsymbol{s}_2$. Finally, in (c), we consider solution $\boldsymbol{x}_3$, its nearest neighbor in $\mathcal{S}$ is $\boldsymbol{s}_2$ and we choose $\boldsymbol{s}_5$ as a random solution. First, $\boldsymbol{x}_3$ competes with $\boldsymbol{s}_5$ and $\boldsymbol{s}_5$ wins because the distance from $\boldsymbol{x}_3$ to $\boldsymbol{s}_2$ is less than the distance from $\boldsymbol{s}_5$ to $\boldsymbol{s}_3$; therefore, $\boldsymbol{x}_3$ competes with $\boldsymbol{s}_2$ and $\boldsymbol{x}_3$ wins because the distance from $\boldsymbol{x}_3$ to $\boldsymbol{x}_6$ is greater than the distance from $\boldsymbol{s}_2$ to $\boldsymbol{s}_6$. Thus, $\boldsymbol{x}_3$ replaces $\boldsymbol{s}_2$. In (d), we can see the new $\mathcal{S}$.

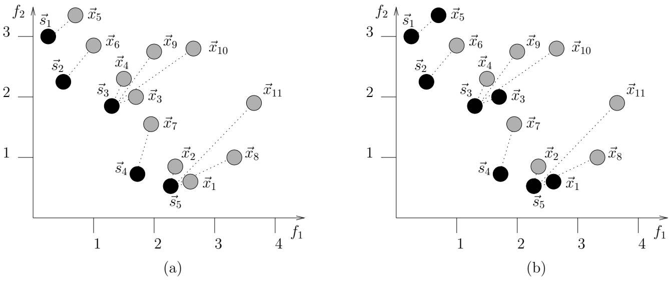
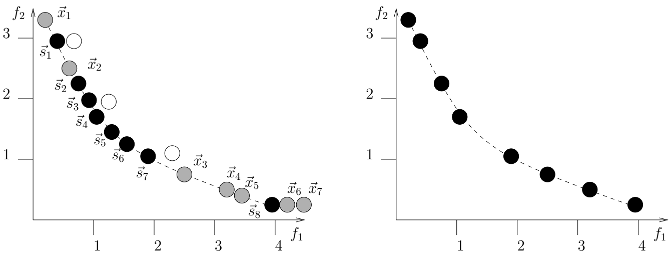


Fig. 3. Let's assume that we want to select eight individuals from the population. In (a), the non-dominated individuals are identified (black points) and then $\mathcal{S} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4, \boldsymbol{s}_5\}$. After that, we calculate d_i for each dominated individual (gray points), we store its nearest neighbor in $\mathcal{S}$ and we sort them with respect to d_i, such that $\boldsymbol{x}_i.d_i \leq \boldsymbol{x}_{i+1}.d_{i+1}$. In (b), we proceed to select the remaining 3 individuals. First, we select individual $\boldsymbol{x}_1$ ($\mathcal{S}' = \{\boldsymbol{x}_1\}$). After that, individual $\boldsymbol{x}_2$ is considered but it is not selected because its nearest neighbor in $\mathcal{S}$ is the same that the nearest neighbor of $\boldsymbol{x}_1$. Then, we consider individual $\boldsymbol{x}_3$ and we select it ($\mathcal{S}' = \{\boldsymbol{x}_1, \boldsymbol{x}_3\}$). Finally, individual $\boldsymbol{x}_4$ is considered and it is not selected because its nearest neighbor is the same that the nearest neighbor of individual $\boldsymbol{x}_3$. Then, we consider individual $\boldsymbol{x}_5$ and we select it ($\mathcal{S}' = \{\boldsymbol{x}_1, \boldsymbol{x}_3, \boldsymbol{x}_5\}$). Therefore, the selected individuals are $\mathcal{S} \cup \mathcal{S}'$ (black points).

called "$\mathcal{B}$". After that, we use the above distribution technique but before each solution $\boldsymbol{x}_i \in \mathcal{B}$ can compete for its survival, we must check if it is similar (in objective function space) to any selected individual in $\mathcal{S}$. We consider that one individual $\boldsymbol{x}$ is similar to another individual $\boldsymbol{y}$, if it is similar in any objective function: $\boldsymbol{x}.f_i - \boldsymbol{y}.f_i < \epsilon$, where ϵ is a small value. In this way, we avoid that weakly non-dominated individuals are selected. If we do not apply this constraint, we can obtain many weakly Pareto optimal solutions, which could prevent convergence. Figure 4 shows how this diversity technique works. The complete selection process is shown in Algorithm 1.



(a) (b)

Fig. 4. Let's assume that we want to select eight individuals from the population. In (a), we select randomly eight non-dominated individuals and we put them in $\mathcal{S}$ (black points), and the remaining non-dominated individuals are placed in $\mathcal{B}$ (gray points). After that, we apply the distribution technique described in Section 3. (b) shows the final $\mathcal{S}$ and we can see that although $\boldsymbol{x}_6$ and $\boldsymbol{x}_7$ could replace individual $\boldsymbol{s}_8$, they were not selected because they are similar to $\boldsymbol{s}_8$.

4.1 Generational Distance - Multi-Objective Evolutionary Algorithm (GD-MOEA)

In order to validate our selection mechanism, we designed a multi-objective evolutionary algorithm which 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 two selection mechanisms: The first is based on decomposition and the second one is based on the approximation of the hypervolume indicator. For this sake, we used the following MOEAs: MOEA/D [27] (using PBI to decompose the MOP) and SMS-EMOA-HYPE (a version of SMS-EMOA [2] that uses a fitness assignment based on the approximation of the hypervolume indicator, proposed in [1]). Since these MOEAs use the same operators as our proposed approach to create new individuals, the comparison is fair.

Our proposed MOEA is called **"Generational Distance - Multi-Objective Evolutionary Algorithm (GD-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

```
Input  : $\mathcal{P}$ (population), $s$ (number of individuals to choose $s < \|\mathcal{P}\|$).
Output: $\mathcal{S}$ (selected individuals).
Put in $\mathcal{S}$ the non-dominated individuals of $\mathcal{P}$;
if $s > \|\mathcal{S}\|$ then
    Put in $\mathcal{B}$ the dominated individuals of $\mathcal{P}$;
    Calculate the Euclidean distance $d_i$ from each individual $\boldsymbol{x}_i \in \mathcal{B}$ to its nearest neighbor
    in $\mathcal{S}$ and we also save its closest non-dominated 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 $\boldsymbol{s} \in \mathcal{S}'$ do
            if $\boldsymbol{s}.neighbor = \mathcal{B}.\boldsymbol{x}_i.neighbor$ then
                $contInd \leftarrow contInd + 1$;
            end
        end
        if $contInd \leq contIndAux$ then
            Put $\mathcal{B}.\boldsymbol{x}_i$ in $\mathcal{S}'$;
        end
        repeat
            $i \leftarrow i + 1$;
        until $\mathcal{B}.\boldsymbol{x}_i \notin \mathcal{S}'$;
        if $i = \|\mathcal{B}\|$ then
            $i \leftarrow 0$, $contIndAux \leftarrow contIndAux + 1$;
        end
    end
    $\mathcal{S} \leftarrow \mathcal{S} \cup \mathcal{S}'$;
else
    if $s < \|\mathcal{S}\|$ then
        Choose randomly $\|\mathcal{S}\| - s$ individuals of $\mathcal{S}$ and put them in a new set called $\mathcal{B}$;
        foreach $\boldsymbol{x}_i \in \mathcal{B}$ do
            if $\boldsymbol{x}_i$ is not similar to any individual in $\mathcal{S}$ then
                Obtain the nearest neighbor, $\boldsymbol{x}_{near}$, of $\boldsymbol{x}_i$ in $\mathcal{S}$;
                Choose a random individual, $\boldsymbol{x}_{rand}$, from $\mathcal{S}$ such that $\boldsymbol{x}_{near} \neq \boldsymbol{x}_{rand}$;
                $dxi \leftarrow$ Euclidean distance from $\boldsymbol{x}_i$ to $\boldsymbol{x}_{near}$;
                $dxr \leftarrow$ Euclidean distance from $\boldsymbol{x}_{rand}$ to its nearest neighbor in $\mathcal{S}$;
                if $dxi > dxr$ then
                    Replace $\boldsymbol{x}_{rand}$ with $\boldsymbol{x}_i$;
                else
                    $dxi \leftarrow$ Euclidean distance from $\boldsymbol{x}_i$ to its nearest neighbor in $\mathcal{S}$
                    without considering $\boldsymbol{x}_{near}$;
                    $dxn \leftarrow$ Euclidean distance from $\boldsymbol{x}_{near}$ to its nearest neighbor in $\mathcal{S}$;
                    if $dxi > dxn$ then
                        Replace $\boldsymbol{x}_{near}$ with $\boldsymbol{x}_i$;
                    end
                end
            end
        end
    end
end
return $\mathcal{S}$;
```

Algorithm 1: I_{GD}-Selection

of size $2P$. Then, we use the proposed selection mechanism 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.

5 Experimental Results

As mentioned before, we compare our proposed GD-MOEA with respect to MOEA/D and SMS-EMOA-HYPE. In the case of MOEA/D, we generated the convex weights using the technique proposed in [8] and after that, we applied clustering (k-means) to obtain a specific number of weights. In the case of SMS-EMOA-HYPE, we used the source code of HyPE available in the public domain [1] adopting 10^4 as our number of samples to assign fitness in the original SMS-EMOA.[3]

For our experiments, we used seven problems taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [9]. 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 [14], with $k\_factor = 2$ and $l\_factor = 10$. For each test problem, we performed 30 independent runs. For all three algorithms, we adopted the parameters suggested by the authors of NSGA-II: $p_c = 0.9$ (crossover probability), $p_m = 1/n$ (mutation probability), where n is the number of decision variables. We also used $\eta_c = 15$ and $\eta_m = 20$, 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).

5.1 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. To calculate the hypervolume indicator, we normalized the approximations of the Pareto optimal set, generated by the MOEAs, and $y_{ref} = [y_1, \cdots, y_k]$ such that $y_i = 1.1$ is used as our reference point. The normalization was performed considering all approximations generated by the different MOEAs (i.e., we put, in one set, all the non-dominated solutions found and from this set we calculate the maximum and minimum for each objective function).

5.2 Discussion of Results

Table 1 shows the results with respect to I_H as well as the results of the statistical analysis that we made to validate our experiments, for which we used Wilcoxon's rank sum. In Table 1, we can see that our proposed GD-MOEA outperformed MOEA/D in forty-three problems and in all cases we can reject the

---

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

null hypothesis (medians are equal). Only in thirteen problems our proposed approach was outperformed by MOEA/D. With respect to SMS-EMOA-HYPE, we can see that our GD-MOEA was outperformed in forty-nine problems. Only in six problems our GD-MOEA outperformed SMS-EMOA-HYPE and in one case they had a similar behavior (we cannot reject the null hypothesis). However, it is important to analyze the running time required by the three algorithms. In Table 2, we can see that MOEA/D is better than our GD-MOEA because, in the worst case, MOEA/D required 1.8199 seconds to find the approximation of the Pareto optimal set while our GD-MOEA required 2.6672 seconds, i.e., MOEA/D is 1.46 times faster than our GD-MOEA. In the case of SMS-EMOA-HYPE, we can see that it required 445.7333 seconds in the worst case, i.e., our GD-MOEA is 167.11 times faster than SMS-EMOA-HYPE. Therefore, we can say that our GD-MOEA is a good choice to solve MOPs having both low dimensionality and high dimensionality in objective function space, if we consider both quality in the approximation of the Pareto set and running time.

Finally, we will present a brief study on the effect of the population size on the performance of our approach. We know that if we increase the number of objective functions, we should increase the population size as well. However, algorithms such as SMS-EMOA cannot be used with large population sizes, because its running time rapidly increases (in the worst case, it needs to calculate P times the contribution to the hypervolume, where P is the population size, in order to decide which individual will be removed). In the case of MOEA/D and our GD-MOEA, it is indeed possible to increase the population size. In order to study the behavior of these two MOEAs, we adopted a population size of 300 individuals. Table 3 shows the results and we can see in (a) that in seven problems both algorithms have a similar behavior because we cannot reject the null hypothesis. In twelve cases, MOEA/D outperformed our proposed GD-MOEA and in thirty-seven cases our proposed GD-MOEA outperformed MOEA/D. With respect to the running time, we can see in (b) that in the worst case, MOEA/D required 4.7445 seconds and our proposed GD-MOEA required 11.4934 seconds, i.e., MOEA/D is 2.42 times faster than our proposed GD-MOEA. Therefore, we can say that our proposed GD-MOEA has a better performance than MOEA/D in most cases, while requiring a higher (but not significantly long) running time.

An interesting thing is that our proposed GD-MOEA had serious difficulties to solve DTLZ1, DTLZ6 and WFG1 with more than three objective functions when we used a population size of 100 individuals. However, when we increased the population size, our proposed GD-MOEA was able to obtain better results, and kept a good behavior, in general, for all the problems considered in our study.

6 Conclusions and Future Work

We have proposed a new selection mechanism based on the generational distance indicator (I_{GD}) and a technique based on Euclidean distances to improve the

f	moead I_H	gd-moea I_H	$P(H)$	sms-emoa-hype I_H	gd-moea I_H	$P(H)$
DTLZ1(3)	1.0710(0.003)	**1.0842(0.005)**	0.00(1)	**1.1011(0.006)**	1.0842(0.005)	0.00(1)
DTLZ2(3)	0.7102(0.000)	**0.7150(0.006)**	0.00(1)	**0.7435(0.002)**	0.7150(0.006)	0.00(1)
DTLZ3(3)	1.3130(0.001)	**1.3276(0.003)**	0.00(1)	**1.3299(0.000)**	1.3276(0.003)	0.00(1)
DTLZ4(3)	0.8191(0.000)	**0.8344(0.008)**	0.00(1)	**0.8639(0.002)**	0.8344(0.008)	0.00(1)
DTLZ5(3)	0.2467(0.001)	**0.2620(0.003)**	0.00(1)	**0.2654(0.000)**	0.2620(0.003)	0.00(1)
DTLZ6(3)	1.0002(0.011)	**1.1060(0.012)**	0.00(1)	1.1048(0.014)	**1.1060(0.012)**	0.92(0)
DTLZ7(3)	0.4472(0.026)	**0.5055(0.061)**	0.00(1)	**0.5389(0.034)**	0.5055(0.061)	0.00(1)
DTLZ1(4)	1.1858(0.005)	**1.2077(0.199)**	0.00(1)	**1.2586(0.057)**	1.2077(0.199)	0.43(0)
DTLZ2(4)	0.8603(0.001)	**0.9030(0.013)**	0.00(1)	**1.0086(0.003)**	0.9030(0.013)	0.00(1)
DTLZ3(4)	1.4556(0.001)	**1.4605(0.006)**	0.00(1)	**1.4636(0.000)**	1.4605(0.006)	0.00(1)
DTLZ4(4)	0.8589(0.001)	**0.9017(0.015)**	0.00(1)	**1.0143(0.004)**	0.9017(0.015)	0.00(1)
DTLZ5(4)	0.8553(0.022)	**0.9122(0.031)**	0.00(1)	**0.9842(0.002)**	0.9122(0.031)	0.00(1)
DTLZ6(4)	**1.0244(0.013)**	0.7536(0.087)	0.00(1)	**1.0561(0.018)**	0.7536(0.087)	0.00(1)
DTLZ7(4)	0.3389(0.008)	**0.5288(0.036)**	0.00(1)	0.5270(0.037)	**0.5288(0.036)**	0.95(1)
DTLZ1(5)	**1.2463(0.011)**	0.4102(0.490)	0.00(1)	**1.2371(0.348)**	0.4102(0.490)	0.00(1)
DTLZ2(5)	0.9483(0.003)	**1.0500(0.027)**	0.00(1)	**1.2614(0.005)**	1.0500(0.027)	0.00(1)
DTLZ3(5)	1.5818(0.008)	**1.5933(0.014)**	0.00(1)	**1.6089(0.000)**	1.5933(0.014)	0.00(1)
DTLZ4(5)	0.9299(0.003)	**1.0280(0.017)**	0.00(1)	**1.2537(0.005)**	1.0280(0.017)	0.00(1)
DTLZ5(5)	1.0353(0.027)	**1.1208(0.104)**	0.00(1)	**1.3005(0.003)**	1.1208(0.104)	0.00(1)
DTLZ6(5)	**1.2919(0.021)**	0.7747(0.104)	0.00(1)	**1.4414(0.009)**	0.7747(0.104)	0.00(1)
DTLZ7(5)	0.0967(0.067)	**0.4426(0.045)**	0.00(1)	**0.4964(0.049)**	0.4426(0.045)	0.00(1)
DTLZ1(6)	**1.3056(0.013)**	0.0149(0.048)	0.00(1)	**1.5000(0.235)**	0.0149(0.048)	0.00(1)
DTLZ2(6)	0.9712(0.011)	**1.1841(0.029)**	0.00(1)	**1.5413(0.005)**	1.1841(0.029)	0.00(1)
DTLZ3(6)	**1.7610(0.004)**	1.7133(0.052)	0.00(1)	**1.7711(0.000)**	1.7133(0.052)	0.00(1)
DTLZ4(6)	0.9559(0.005)	**1.1805(0.044)**	0.00(1)	**1.5446(0.005)**	1.1805(0.044)	0.00(1)
DTLZ5(6)	0.7277(0.014)	**0.8356(0.037)**	0.00(1)	**1.0777(0.010)**	0.8356(0.037)	0.00(1)
DTLZ6(6)	**1.3298(0.037)**	0.5249(0.069)	0.00(1)	**1.6348(0.010)**	0.5249(0.069)	0.00(1)
DTLZ7(6)	0.0194(0.004)	**0.5360(0.066)**	0.00(1)	0.4480(0.125)	**0.5360(0.066)**	0.00(1)
WFG1(3)	**0.9183(0.017)**	0.8572(0.046)	0.00(1)	**1.0174(0.068)**	0.8572(0.046)	0.00(1)
WFG2(3)	0.1539(0.202)	**0.5245(0.128)**	0.00(1)	**0.6506(0.055)**	0.5245(0.128)	0.00(1)
WFG3(3)	0.4989(0.026)	**0.5982(0.008)**	0.00(1)	**0.6061(0.007)**	0.5982(0.008)	0.00(1)
WFG4(3)	0.5943(0.013)	**0.6473(0.008)**	0.00(1)	**0.7023(0.005)**	0.6473(0.008)	0.00(1)
WFG5(3)	0.4710(0.010)	**0.5286(0.004)**	0.00(1)	**0.5370(0.003)**	0.5286(0.004)	0.00(1)
WFG6(3)	0.4548(0.007)	**0.5245(0.008)**	0.00(1)	**0.5475(0.004)**	0.5245(0.008)	0.00(1)
WFG7(3)	0.4933(0.056)	**0.6528(0.012)**	0.00(1)	0.5617(0.029)	**0.6528(0.012)**	0.00(1)
WFG1(4)	**1.1040(0.058)**	0.6312(0.110)	0.00(1)	**1.1404(0.026)**	0.6312(0.110)	0.00(1)
WFG2(4)	0.0030(0.016)	**0.1481(0.179)**	0.00(1)	**0.4888(0.227)**	0.1481(0.179)	0.00(1)
WFG3(4)	0.2872(0.034)	**0.4502(0.025)**	0.00(1)	**0.5343(0.016)**	0.4502(0.025)	0.00(1)
WFG4(4)	0.6492(0.026)	**0.7987(0.015)**	0.00(1)	**0.9308(0.008)**	0.7987(0.015)	0.00(1)
WFG5(4)	0.3672(0.015)	**0.5232(0.006)**	0.00(1)	**0.5586(0.005)**	0.5232(0.006)	0.00(1)
WFG6(4)	0.2887(0.016)	**0.3580(0.044)**	0.00(1)	**0.5654(0.010)**	0.3580(0.044)	0.00(1)
WFG7(4)	0.2887(0.036)	**0.6944(0.017)**	0.00(1)	0.4220(0.032)	**0.6944(0.017)**	0.00(1)
WFG1(5)	**1.2195(0.063)**	0.4916(0.037)	0.00(1)	**1.2517(0.027)**	0.4916(0.037)	0.00(1)
WFG2(5)	0.0105(0.033)	**0.1194(0.122)**	0.00(1)	**0.4422(0.241)**	0.1194(0.122)	0.00(1)
WFG3(5)	0.1508(0.038)	**0.2609(0.062)**	0.00(1)	**0.4852(0.027)**	0.2609(0.062)	0.00(1)
WFG4(5)	0.6399(0.024)	**0.8698(0.029)**	0.00(1)	**1.1159(0.019)**	0.8698(0.029)	0.00(1)
WFG5(5)	0.2401(0.014)	**0.4614(0.018)**	0.00(1)	**0.5798(0.010)**	0.4614(0.018)	0.00(1)
WFG6(5)	**0.2408(0.016)**	0.2149(0.054)	0.00(1)	**0.5222(0.022)**	0.2149(0.054)	0.00(1)
WFG7(5)	0.2149(0.014)	**0.6888(0.019)**	0.00(1)	0.3138(0.022)	**0.6888(0.019)**	0.00(1)
WFG1(6)	**1.1466(0.022)**	0.5648(0.041)	0.00(1)	**1.3539(0.029)**	0.5648(0.041)	0.00(1)
WFG2(6)	0.0094(0.034)	**0.1403(0.153)**	0.00(1)	**0.4834(0.239)**	0.1403(0.153)	0.00(1)
WFG3(6)	**0.0993(0.044)**	0.0962(0.044)	0.58(0)	**0.4375(0.034)**	0.0962(0.044)	0.00(1)
WFG4(6)	0.5947(0.029)	**0.9251(0.035)**	0.00(1)	**1.2866(0.025)**	0.9251(0.035)	0.00(1)
WFG5(6)	0.1613(0.017)	**0.3368(0.044)**	0.00(1)	**0.5823(0.015)**	0.3368(0.044)	0.00(1)
WFG6(6)	**0.2273(0.021)**	0.1651(0.049)	0.00(1)	**0.5142(0.033)**	0.1651(0.049)	0.00(1)
WFG7(6)	0.1842(0.014)	**0.6745(0.031)**	0.00(1)	0.2735(0.018)	**0.6745(0.031)**	0.00(1)

Table 1. Results obtained in the DTLZ and WFG test problems. We compare our proposed GD-MOEA with respect to 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 Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level.

diversity of the population. Our idea is to use I_{GD} as a convergence strategy and,

f	moead	gd-moea	sms-emoa-hype	gd-moea
	time	time	time	time
DTLZ1(3)	**0.4993(0.016)**	0.7695(0.010)	47.0000(2.620)	**0.7695(0.010)**
DTLZ2(3)	**0.5783(0.010)**	1.1593(0.014)	106.1333(4.105)	**1.1593(0.014)**
DTLZ3(3)	**0.5195(0.012)**	0.6366(0.026)	135.9667(21.629)	**0.6366(0.026)**
DTLZ4(3)	**0.6037(0.008)**	1.2361(0.082)	107.1667(3.822)	**1.2361(0.082)**
DTLZ5(3)	**0.5922(0.007)**	1.0845(0.015)	64.3333(5.430)	**1.0845(0.015)**
DTLZ6(3)	**0.5007(0.018)**	0.8883(0.035)	59.0667(9.747)	**0.8883(0.035)**
DTLZ7(3)	**0.5397(0.008)**	0.9039(0.038)	98.4333(9.106)	**0.9039(0.038)**
DTLZ1(4)	**0.5230(0.008)**	0.8578(0.033)	59.6667(3.280)	**0.8578(0.033)**
DTLZ2(4)	**0.6147(0.012)**	1.1623(0.014)	156.0333(6.555)	**1.1623(0.014)**
DTLZ3(4)	**0.5533(0.020)**	0.8227(0.047)	165.9333(18.995)	**0.8227(0.047)**
DTLZ4(4)	**0.6440(0.011)**	1.2127(0.021)	157.2667(9.602)	**1.2127(0.021)**
DTLZ5(4)	**0.6128(0.009)**	1.1188(0.023)	143.1667(4.796)	**1.1188(0.023)**
DTLZ6(4)	**0.5351(0.010)**	1.2979(0.019)	129.1000(7.648)	**1.2979(0.019)**
DTLZ7(4)	**0.5860(0.008)**	0.9282(0.016)	185.6667(16.067)	**0.9282(0.016)**
DTLZ1(5)	**0.5532(0.005)**	0.9387(0.033)	79.1333(5.632)	**0.9387(0.033)**
DTLZ2(5)	**0.6453(0.010)**	1.1378(0.030)	188.3333(8.231)	**1.1378(0.030)**
DTLZ3(5)	**0.5785(0.012)**	1.0330(0.072)	177.1000(24.347)	**1.0330(0.072)**
DTLZ4(5)	**0.6949(0.004)**	1.2085(0.022)	190.5000(6.845)	**1.2085(0.022)**
DTLZ5(5)	**0.6455(0.004)**	1.1452(0.030)	229.3333(14.328)	**1.1452(0.030)**
DTLZ6(5)	**0.5784(0.008)**	1.6253(0.033)	225.3667(11.056)	**1.6253(0.033)**
DTLZ7(5)	**0.6289(0.004)**	0.9783(0.068)	296.9333(23.678)	**0.9783(0.068)**
DTLZ1(6)	**0.5816(0.011)**	1.4877(0.288)	98.9333(6.904)	**1.4877(0.288)**
DTLZ2(6)	**0.6750(0.003)**	1.1846(0.122)	233.3667(11.182)	**1.1846(0.122)**
DTLZ3(6)	**0.6162(0.017)**	1.5827(0.185)	185.3000(22.371)	**1.5827(0.185)**
DTLZ4(6)	**0.7485(0.003)**	1.2092(0.024)	234.6333(10.581)	**1.2092(0.024)**
DTLZ5(6)	**0.6683(0.011)**	1.1497(0.024)	336.9000(18.293)	**1.1497(0.024)**
DTLZ6(6)	**0.6308(0.006)**	1.6625(0.024)	340.4333(16.669)	**1.6625(0.024)**
DTLZ7(6)	**0.6589(0.012)**	1.0392(0.026)	377.9000(42.232)	**1.0392(0.026)**
WFG1(3)	**1.1427(0.019)**	1.5656(0.027)	147.0000(3.670)	**1.5656(0.027)**
WFG2(3)	**0.9272(0.024)**	1.3674(0.018)	98.4333(6.786)	**1.3674(0.018)**
WFG3(3)	**0.9738(0.018)**	1.4281(0.017)	148.7333(3.941)	**1.4281(0.017)**
WFG4(3)	**0.9919(0.007)**	2.2732(0.061)	107.5000(4.233)	**2.2732(0.061)**
WFG5(3)	**0.9594(0.007)**	2.1035(0.031)	153.0667(8.246)	**2.1035(0.031)**
WFG6(3)	**0.9478(0.010)**	1.5356(0.007)	168.9333(8.330)	**1.5356(0.007)**
WFG7(3)	**1.1988(0.026)**	2.3854(0.020)	151.5667(6.530)	**2.3854(0.020)**
WFG1(4)	**1.1697(0.017)**	1.4167(0.017)	233.7333(8.434)	**1.4167(0.017)**
WFG2(4)	**0.9473(0.021)**	1.4471(0.016)	170.6333(11.232)	**1.4471(0.016)**
WFG3(4)	**1.0207(0.011)**	1.3199(0.034)	247.1000(7.939)	**1.3199(0.034)**
WFG4(4)	**1.0258(0.009)**	2.4373(0.017)	157.8333(6.455)	**2.4373(0.017)**
WFG5(4)	**0.9848(0.009)**	1.8332(0.080)	206.7667(19.689)	**1.8332(0.080)**
WFG6(4)	**0.9774(0.007)**	1.2582(0.010)	216.9667(17.647)	**1.2582(0.010)**
WFG7(4)	**1.2529(0.014)**	2.4961(0.021)	252.1333(8.429)	**2.4961(0.021)**
WFG1(5)	**1.2474(0.015)**	1.5476(0.027)	335.0667(7.607)	**1.5476(0.027)**
WFG2(5)	**1.0083(0.020)**	1.5561(0.015)	269.5667(20.717)	**1.5561(0.015)**
WFG3(5)	**1.0908(0.010)**	1.4337(0.018)	378.1667(6.362)	**1.4337(0.018)**
WFG4(5)	**1.1067(0.005)**	2.5754(0.015)	220.6667(13.553)	**2.5754(0.015)**
WFG5(5)	**1.0683(0.006)**	1.4752(0.033)	276.2000(31.841)	**1.4752(0.033)**
WFG6(5)	**1.0342(0.024)**	1.3588(0.013)	274.2667(47.308)	**1.3588(0.013)**
WFG7(5)	**1.4166(0.021)**	2.6519(0.051)	358.9667(10.005)	**2.6519(0.051)**
WFG1(6)	**1.3214(0.012)**	1.6343(0.022)	383.8000(42.576)	**1.6343(0.022)**
WFG2(6)	**1.0430(0.021)**	1.6106(0.019)	377.4333(29.319)	**1.6106(0.019)**
WFG3(6)	**1.1115(0.011)**	1.5232(0.037)	445.7333(46.018)	**1.5232(0.037)**
WFG4(6)	**1.1695(0.009)**	2.6444(0.049)	316.2000(12.098)	**2.6444(0.049)**
WFG5(6)	**1.1185(0.009)**	1.4109(0.027)	246.7000(6.435)	**1.4109(0.027)**
WFG6(6)	**1.0602(0.024)**	1.4250(0.013)	259.2333(5.024)	**1.4250(0.013)**
WFG7(6)	**1.8199(0.145)**	2.6672(0.072)	408.2667(40.609)	**2.6672(0.072)**

Table 2. Time required (in seconds) by MOEA/D, SMS-EMOA-HYPE and our proposed GD-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.

when having many non-dominated individuals, to switch to the use of a technique to maintain diversity. However, it is important to be careful in both cases. When using I_{GD} only as a convergence strategy, if we choose the individuals with low values of d_i without considering if we have already selected individuals close to a particular non-dominated individual, then, we can have difficulties, e.g., in MOPs with disconnected Pareto fronts such as DTLZ7 (in this case, we will only obtain some portions of the Pareto front). If we only use the distribution technique based on Euclidean distances without considering if the individual which will compete is similar to another individual in one objective function, then, we can obtain many weakly Pareto points and this could prevent us from converging to the true Pareto front.

Our preliminary results indicate that our proposed GD-MOEA is a good option to solve MOPs having both low and high dimensionality in objective function space, if we consider both quality in the solutions and running time required to obtain them. Our proposed approach is able to obtain better results than MOEA/D, in most cases, and MOEA/D is only 1.46 times faster than our GD-MOEA when we use a population size of 100 individuals and, it is only 2.4224 times faster when we use a population size of 300 individuals. Although, SMS-EMOA-HYPE is better, in most cases, than GD-MOEA in terms of the quality of the solutions generated, it requires up to 167.11 times more computational time than our proposed approach.

As part of our future work, we want to improve our proposed selection mechanism so that it can deal (in a better way) with problems in which many weakly Pareto optimal solutions are generated such as DTLZ1 and DTLZ3. Also, we want to use other indicators to conduct an in-depth study, e.g., we could use the two set coverage indicator to measure convergence and the spacing indicator to assess the quality of the distribution of solutions generated by our proposed approach. This is because SMS-EMOA-HYPE maximizes the hypervolume indicator, and therefore, it has advantages over other two MOEAs when we use the hypervolume indicator to assess our results (evidently, it is expected that SMS-EMOA-HYPE will have better hypervolume values than other MOEAs, since this is precisely the value that it aims to maximize).

References

1. Johannes Bader and Eckart Zitzler. HypE: An Algorithm for Fast Hypervolume-Based Many-Objective Optimization. *Evolutionary Computation*, 19(1):45–76, Spring, 2011.
2. Nicola Beume, Boris Naujoks, and Michael Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 16 September 2007.
3. Karl Bringmann and Tobias Friedrich. Approximating the least hypervolume contributor: NP-hard in general, but fast in practice. *Theoretical Computer Science*, 425:104–116, March 30 2012.
4. Dimo Brockhoff, Tobias Wagner, and Heike Trautmann. On the Properties of the $R2$ Indicator. In *2012 Genetic and Evolutionary Computation Conference*

(GECCO'2012), pages 465–472, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.

5. Carlos A. Coello Coello and Nareli Cruz Cortés. Solving Multiobjective Optimization Problems using an Artificial Immune System. *Genetic Programming and Evolvable Machines*, 6(2):163–190, June 2005.
6. Carlos A. Coello Coello, Gary B. Lamont, and David A. Van Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, New York, second edition, September 2007. ISBN 978-0-387-33254-3.
7. Carlos A. Coello Coello, David A. Van Veldhuizen, and Gary B. Lamont. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, New York, May 2002. ISBN 0-3064-6762-3.
8. Indraneel 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*, 8(3):631–657, March 1998.
9. Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In Ajith Abraham, Lakhmi Jain, and Robert Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.
10. Christian Domínguez-Medina, Güenter Rudolph, Oliver Schüetze, and Heike Trautmann. Evenly Spaced Pareto Fronts of Quad-objective Problems using PSA Partitioning Technique. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 3190–3197, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
11. M. Farina and P. Amato. On the Optimal Solution Definition for Many-criteria Optimization Problems. In *Proceedings of the NAFIPS-FLINT International Conference'2002*, pages 233–238, Piscataway, New Jersey, June 2002. IEEE Service Center.
12. 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)*, pages 975–980, Mérida, Yucatán, México, October 2011. IEEE Press.
13. Raquel Hernández Gómez and Carlos 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)*, pages 2488–2495, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
14. Simon Huband, Phil Hingston, Luigi Barone, and Lyndon While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.
15. Christian Igel, Nikolaus Hansen, and Stefan Roth. Covariance Matrix Adaptation for Multi-objective Optimization. *Evolutionary Computation*, 15(1):1–28, Spring 2007.
16. Joshua Knowles and David Corne. Properties of an Adaptive Archiving Algorithm for Storing Nondominated Vectors. *IEEE Transactions on Evolutionary Computation*, 7(2):100–116, April 2003.
17. Adriana Menchaca-Mendez and Carlos A. Coello Coello. A New Selection Mechanism Based on Hypervolume and its Locality Property. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 924–931, Cancún, México, 20-23 June 2013. IEEE Press.

18. Adriana Menchaca-Mendez and Carlos A. Coello Coello. MD-MOEA : A New MOEA based on the Maximin Fitness Function and Euclidean Distances between Solutions. In *2014 IEEE Congress on Evolutionary Computation (CEC'2014)*, pages 2148–2155, Beijing, China, 6-11 July 2014. IEEE Press. ISBN 978-1-4799-1488-3.
19. Cynthia A. Rodríguez Villalobos and Carlos A. Coello Coello. A New Multi-Objective Evolutionary Algorithm Based on a Performance Assessment Indicator. In *2012 Genetic and Evolutionary Computation Conference (GECCO'2012)*, pages 505–512, Philadelphia, USA, July 2012. ACM Press. ISBN: 978-1-4503-1177-9.
20. Oliver Schütze, Xavier Esquivel, Adriana Lara, and Carlos A. Coello Coello. Using the Averaged Hausdorff Distance as a Performance Measure in Evolutionary Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 16(4):504–522, August 2012.
21. Heike Trautmann, Tobias Wagner, and Dimo Brockhoff. R2-emoa: Focused multi-objective search using r2-indicator-based selection. In Giuseppe Nicosia and Panos Pardalos, editors, *Learning and Intelligent Optimization*, Lecture Notes in Computer Science, pages 70–74. Springer Berlin Heidelberg, 2013.
22. Dũng H. Phan and Junichi Suzuki. R2-IBEA: R2 Indicator Based Evolutionary Algorithm for Multiobjective Optimization. In *2013 IEEE Congress on Evolutionary Computation (CEC'2013)*, pages 1836–1845, Cancún, México, 20-23 June 2013. IEEE Press. ISBN 978-1-4799-0454-9.
23. David A. Van 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.
24. David A. Van Veldhuizen and Gary B. Lamont. Evolutionary Computation and Convergence to a Pareto Front. In John R. Koza, editor, *Late Breaking Papers at the Genetic Programming 1998 Conference*, pages 221–228, Stanford University, California, July 1998. Stanford University Bookstore.
25. David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Evolutionary Algorithm Test Suites. In Janice Carroll, Hisham Haddad, Dave Oppenheim, Barrett Bryant, and Gary B. Lamont, editors, *Proceedings of the 1999 ACM Symposium on Applied Computing*, pages 351–357, San Antonio, Texas, 1999. ACM.
26. Tobias Wagner, Heike Trautmann, and Dimo Brockhoff. Preference Articulation by Means of the $R2$ Indicator. In Robin C. Purshouse, Peter J. Fleming, Carlos M. Fonseca, Salvatore Greco, and Jane Shaw, editors, *Evolutionary Multi-Criterion Optimization, 7th International Conference, EMO 2013*, pages 81–95. Springer. Lecture Notes in Computer Science Vol. 7811, Sheffield, UK, March 19-22 2013.
27. Qingfu Zhang and Hui Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.
28. Eckart Zitzler and Simon Künzli. Indicator-based Selection in Multiobjective Search. In Xin Yao et al., editor, *Parallel Problem Solving from Nature - PPSN VIII*, pages 832–842, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.
29. Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M. Fonseca, and Viviane Grunert da Fonseca. Performance Assessment of Multiobjective Optimizers: An Analysis and Review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, April 2003.

f	moead I_H	gd-moea I_H	$P(H)$	moead time	gd-moea time
DTLZ1 (3)	**1.0395 (0.001)**	1.0265 (0.008)	0.00 (1)	**1.5296 (0.023)**	4.4234 (0.095)
DTLZ2 (3)	**0.8847 (0.000)**	0.8834 (0.004)	0.34 (0)	**1.7870 (0.011)**	5.5353 (0.143)
DTLZ3 (3)	1.3307 (0.000)	**1.3308 (0.001)**	0.00 (1)	**1.5982 (0.042)**	3.2699 (0.176)
DTLZ4 (3)	**0.7786 (0.000)**	0.7717 (0.007)	0.00 (1)	bf 1.7937 (0.015)	5.4645 (0.099)
DTLZ5 (3)	0.2612 (0.000)	**0.2666 (0.002)**	0.00 (1)	**1.7661 (0.039)**	5.4361 (0.058)
DTLZ6 (3)	**1.0933 (0.005)**	1.0813 (0.022)	0.00 (1)	**1.4588 (0.028)**	4.3155 (0.044)
DTLZ7 (3)	**0.6397 (0.001)**	0.5852 (0.080)	0.00 (1)	**1.5896 (0.021)**	4.8026 (0.146)
DTLZ1 (4)	1.3880 (0.003)	**1.4575 (0.002)**	0.00 (1)	**1.6032 (0.024)**	4.9574 (0.090)
DTLZ2 (4)	0.9845 (0.000)	**0.9906 (0.046)**	0.00 (1)	**1.8629 (0.008)**	5.0777 (0.096)
DTLZ3 (4)	1.4639 (0.000)	**1.4641 (0.000)**	0.00 (1)	**1.7703 (0.041)**	4.1087 (0.200)
DTLZ4 (4)	1.0157 (0.001)	**1.0201 (0.013)**	0.17 (0)	**1.9052 (0.018)**	5.1225 (0.033)
DTLZ5 (4)	1.0040 (0.002)	**1.0405 (0.009)**	0.00 (1)	**1.8249 (0.038)**	4.0589 (0.047)
DTLZ6 (4)	**1.3351 (0.003)**	1.2511 (0.025)	0.00 (1)	**1.5590 (0.029)**	5.9208 (0.049)
DTLZ7 (4)	0.4997 (0.002)	**0.6487 (0.053)**	0.00 (1)	**1.7103 (0.024)**	4.9728 (0.039)
DTLZ1 (5)	1.5813 (0.003)	**1.6104 (0.000)**	0.00 (1)	**1.6717 (0.019)**	5.4628 (0.091)
DTLZ2 (5)	1.1373 (0.002)	**1.1504 (0.022)**	0.00 (1)	**1.9844 (0.016)**	5.0243 (0.110)
DTLZ3 (5)	1.6102 (0.000)	**1.6104 (0.000)**	0.00 (1)	**1.8143 (0.042)**	5.1946 (0.265)
DTLZ4 (5)	1.1464 (0.002)	**1.1500 (0.041)**	0.25 (0)	**2.0551 (0.015)**	5.0612 (0.033)
DTLZ5 (5)	1.2309 (0.002)	**1.2474 (0.015)**	0.00 (1)	**2.0464 (0.115)**	4.0763 (0.043)
DTLZ6 (5)	**1.4478 (0.005)**	1.2493 (0.036)	0.00 (1)	**1.6522 (0.043)**	7.0208 (0.111)
DTLZ7 (5)	0.3545 (0.009)	**0.6688 (0.026)**	0.00 (1)	**1.8392 (0.014)**	5.1262 (0.057)
DTLZ1 (6)	1.7573 (0.001)	**1.7716 (0.000)**	0.00 (1)	**1.7966 (0.040)**	6.0618 (0.096)
DTLZ2 (6)	1.1549 (0.004)	**1.2616 (0.040)**	0.00 (1)	**2.0712 (0.019)**	5.1637 (0.063)
DTLZ3 (6)	**1.7697 (0.000)**	1.7694 (0.004)	0.00 (1)	**1.9080 (0.023)**	6.5666 (0.679)
DTLZ4 (6)	1.1329 (0.003)	**1.3010 (0.054)**	0.00 (1)	**2.2384 (0.016)**	5.4694 (0.084)
DTLZ5 (6)	1.2176 (0.007)	**1.3123 (0.019)**	0.00 (1)	**2.0917 (0.108)**	4.4795 (0.047)
DTLZ6 (6)	**1.6242 (0.005)**	1.3955 (0.049)	0.00 (1)	**1.7541 (0.032)**	7.9697 (0.137)
DTLZ7 (6)	0.0458 (0.009)	**0.5867 (0.029)**	0.00 (1)	**1.9459 (0.024)**	5.6369 (0.101)
WFG1 (3)	**0.7547 (0.002)**	0.6978 (0.016)	0.00 (1)	**3.3185 (0.042)**	5.4840 (0.082)
WFG2 (3)	0.7263 (0.079)	**0.9019 (0.034)**	0.00 (1)	**2.8354 (0.129)**	6.2730 (0.055)
WFG3 (3)	0.6044 (0.012)	**0.6241 (0.004)**	0.00 (1)	**2.8724 (0.052)**	5.5970 (0.044)
WFG4 (3)	0.7233 (0.006)	**0.7528 (0.004)**	0.00 (1)	**2.9580 (0.041)**	10.9987 (0.068)
WFG5 (3)	0.5484 (0.004)	**0.5673 (0.003)**	0.00 (1)	**2.8350 (0.028)**	8.8325 (0.235)
WFG6 (3)	0.5130 (0.004)	**0.5444 (0.005)**	0.00 (1)	**2.8116 (0.022)**	5.8362 (0.035)
WFG7 (3)	**0.7685 (0.011)**	0.7498 (0.009)	0.00 (1)	**3.4059 (0.025)**	10.2300 (0.120)
WFG1 (4)	0.2787 (0.018)	**0.4129 (0.033)**	0.00 (1)	**3.4881 (0.038)**	5.5105 (0.027)
WFG2 (4)	0.9008 (0.162)	**0.9788 (0.078)**	0.13 (0)	**2.8905 (0.109)**	6.5906 (0.056)
WFG3 (4)	0.4771 (0.017)	**0.5514 (0.013)**	0.00 (1)	**3.1342 (0.059)**	5.5723 (0.038)
WFG4 (4)	0.8773 (0.015)	**0.9835 (0.010)**	0.00 (1)	**3.0843 (0.018)**	11.4934 (0.153)
WFG5 (4)	0.5234 (0.008)	**0.5960 (0.006)**	0.00 (1)	**3.0682 (0.033)**	6.2124 (0.032)
WFG6 (4)	0.3364 (0.012)	**0.4274 (0.032)**	0.00 (1)	**2.9030 (0.013)**	5.6417 (0.035)
WFG7 (4)	0.7698 (0.036)	**0.8675 (0.015)**	0.00 (1)	**3.8373 (0.054)**	9.0491 (0.171)
WFG1 (5)	0.0533 (0.032)	**0.4287 (0.020)**	0.00 (1)	**3.7985 (0.056)**	6.2184 (0.037)
WFG2 (5)	0.9970 (0.134)	**1.0072 (0.162)**	0.55 (0)	**3.0944 (0.083)**	7.0006 (0.068)
WFG3 (5)	0.4882 (0.027)	**0.5041 (0.043)**	0.29 (0)	**3.2425 (0.060)**	6.1288 (0.060)
WFG4 (5)	0.9023 (0.035)	**1.0894 (0.017)**	0.00 (1)	**3.3310 (0.017)**	11.4864 (0.050)
WFG5 (5)	0.5085 (0.008)	**0.6029 (0.020)**	0.00 (1)	**3.1917 (0.021)**	5.7056 (0.046)
WFG6 (5)	0.2608 (0.012)	**0.2939 (0.042)**	0.00 (1)	**3.0811 (0.011)**	6.1855 (0.032)
WFG7 (5)	0.3798 (0.059)	**0.8928 (0.020)**	0.00 (1)	**4.4487 (0.087)**	8.7778 (0.227)
WFG1 (6)	0.0000 (0.000)	**0.4658 (0.020)**	0.00 (1)	**4.0216 (0.077)**	6.6320 (0.020)
WFG2 (6)	**1.2803 (0.199)**	1.1827 (0.224)	0.06 (0)	**3.1562 (0.066)**	7.1867 (0.063)
WFG3 (6)	**0.5612 (0.059)**	0.3782 (0.086)	0.00 (1)	**3.3847 (0.074)**	6.4182 (0.042)
WFG4 (6)	0.8909 (0.022)	**1.1262 (0.038)**	0.00 (1)	**3.4896 (0.063)**	11.1641 (0.074)
WFG5 (6)	0.5136 (0.011)	**0.6429 (0.031)**	0.00 (1)	**3.3490 (0.015)**	5.9626 (0.065)
WFG6 (6)	**0.2565 (0.019)**	0.2046 (0.034)	0.00 (1)	**3.1610 (0.019)**	6.4419 (0.036)
WFG7 (6)	0.2596 (0.009)	**0.7964 (0.032)**	0.00 (1)	**4.7445 (0.088)**	8.1669 (0.314)

(a) (b)

Table 3. In (a), we show the results obtained in the DTLZ and WFG test problems using a population size of 300 individuals. We compare our GD-MOEA with respect to MOEA/D, 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 Wilcoxon's rank sum. P is the probability of observing the given result (the null hypothesis is true). Small values of P cast doubt on the validity of the null hypothesis. $H = 0$ indicates that the null hypothesis ("medians are equal") cannot be rejected at the 5% level. $H = 1$ indicates that the null hypothesis can be rejected at the 5% level. In (b), we show the time required by MOEA/D and our proposed GD-MOEA for the test problems adopted in seconds. All algorithms were compiled using the GNU C compiler and they were executed on a computer with a 2.66GHz processor having 4GB in RAM.