Solving Multi-Objective Optimization Problems using Differential Evolution and a Maximin Selection Criterion

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 operator (based on a maximin scheme and a clustering technique), which is incorporated into a differential evolution algorithm to solve multi-objective optimization problems. The resulting algorithm is called Maximin-Clustering Differential Evolution (MCDE) and, is validated using standard test problems and performance measures taken from the specialized literature. Our preliminary results indicate that MCDE is able to outperform NSGA-II and that is competitive with a hypervolume-based approach (SMS-EMOA), but at a significantly lower computational cost.

I. INTRODUCTION

Many optimization problems arising in the real world involve multiple objective functions which must be satisfied simultaneously. They are generically called multiobjective optimization problems (MOPs) and usually their objectives are in conflict. In MOPs, the notion of optimality refers to the best possible trade-offs among the objectives. Consequently, no single solution exists, but several (the so-called Pareto optimal set whose image is called the Pareto front). When applying evolutionary algorithms to solve MOPs, we normally 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.

When studying multi-objective evolutionary algorithms (MOEAs), we find two main types of approaches: (i) those that incorporate the concept of Pareto optimality in their selection mechanism, and (ii) those that do not use Pareto dominance to select individuals.

Although the use of Pareto-based selection (mainly through the use of some Pareto ranking scheme [1]) has been the most popular choice within the specialized literature for the last 15 years, such type of approach has several limitations. From them, its poor scalability (when increasing the number of objectives) is, perhaps, the most remarkable. The quick increase in the number of nondominated solutions as we increase the number of objectives, rapidly dilutes the effect of the selection mechanism of a MOEA [2]. This has triggered an important amount of research on the so-called “many-objective optimization”, which refers to the study of problems having four or more objective functions.

In the current literature, we can identify three main approaches to cope with many-objective problems, namely: (i) to adopt or propose a preference relation that induces a finer grain order on the solutions than that induced by the Pareto dominance relation [3], [4], [5], [6], (ii) to reduce the number of objectives of the problem during the search process [7] or, a posteriori, during the decision making process [8], [9], and, (iii) to adopt a selection scheme that does not rely on Pareto optimality (e.g., using compromise functions [10], alternative ranking schemes [11] or a selection mechanism based on a performance measure (from which hypervolume has been a popular choice, in spite of its considerably high computational cost [13], [14]). Here, we study an approach from the third class, using differential evolution as our search engine. The main motivation of this work is to propose an alternative selection mechanism for MOEAs that can properly deal with many-objective optimization problems at a reasonably low computational cost.

The focus of our study is the maximin fitness function [15]. This technique assigns a fitness to each individual in the population without using the concept of Pareto optimality. This scheme encompasses a guidance mechanism based on very simple (and computationally efficient) operations. Our preliminary study of this approach has indicated its suitability as a selection operator in a MOEA whose search engine adopts differential evolution [16], even in the presence of a high number of objectives. However, its lack of an appropriate diversity maintenance mechanism makes it inappropriate with respect to state-of-the-art MOEAs, which led us to propose the incorporation of a clustering technique. The proposed approach, called Maximin-Clustering Differential Evolution (MCDE) is validated with several standard test problems and performance measures. As will be seen later on, our proposed MCDE is able to outperform NSGA-II [17] and is competitive with a state-of-the-art hypervolume-based MOEA (SMS-EMOA) [14], but requiring a much lower computational cost.

The remainder of this paper is organized as follows. Section II states the problem of our interest. The maximin fitness function is briefly described in Section III. Section IV describes in detail the selection operator that we propose and...
in Section V we present a full description of our approach. The experiments performed and the results obtained are shown in Section VI. Finally, we provide our conclusions and some possible paths for future work in Section VII.

II. PROBLEM STATEMENT

The problem of our interest is the general multi-objective optimization problem (MOP) which is defined as follows:

Find \( \vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T \) which optimizes

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

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

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

**Definition 2:** A point \( \vec{x}^* \in \Omega \) is Pareto optimal if and only if for all \( \vec{x} \in \Omega \) we have that \( \vec{x}^* \preceq_p \vec{x} \).

**Definition 3:** For a given MOP, \( \vec{f}(\vec{x}) \), the Pareto optimal set is defined as: \( P^* = \{ \vec{x} \in \Omega : \exists \vec{y} \in \Omega : \vec{f}(\vec{y}) \leq_p \vec{f}(\vec{x}) \} \).

**Definition 4:** Let \( \vec{f}(\vec{x}) \) be a given MOP and \( P^* \) the Pareto optimal set. Then the Pareto Front is defined as: \( \mathcal{PF}^* = \{ \vec{f}(\vec{x}) \mid \vec{x} \in P^* \} \).

III. MAXIMIN FITNESS FUNCTION

The maximin fitness function presented in [15] is derived from the **Definition 1** and, we can use it to solve MOPs. Let’s consider a MOP with \( K \) objectives and let’s assume that each objective is normalized. Let’s consider an evolutionary algorithm whose population size is \( P \). Let \( f_i^k \) be the normalized value of the \( k^{th} \) objective for the \( i^{th} \) individual in a particular generation. Assuming minimization, the \( j^{th} \) individual dominates the \( i^{th} \) individual if:

\[
f_j^k \leq f_k^k \forall k \in \{1, \ldots, K\} \quad \text{and} \quad \exists k \in \{1, \ldots, K\} \mid f_i^k < f_k^k
\]

Eq. (2) is equivalent to:

\[
\min_k (f_i^k - f_k^k) \geq 0
\]

The \( i^{th} \) individual in a particular generation will be dominated by another individual in the generation if:

\[
\max_{j \neq i} (\min_k (f_k^j - f_k^i)) > 0
\]

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

\[
\text{fitness}^i = \max_{j \neq i} (\min_k (f_k^j - f_k^i))
\]

In eq. (5), the \( \min \) is taken over all the objectives from 1 to \( K \), and the \( \max \) is taken over all the individuals in the population from 1 to \( P \), except for the same individual \( i \). From eq. (5), we know that any individual whose maximin fitness is greater than zero is a dominated solution and that, any individual whose maximin fitness is less than zero is a non-dominated solution. Finally, any individual whose maximin fitness is equal to zero is either a dominated solution or a duplicate non-dominated solution.

The algorithm to calculate the fitness of each individual of the population is shown in Algorithm 1. The complexity of this algorithm is \( O(KP^2) \), where \( K \) is the number of objective functions and \( P \) is the population size. In Algorithm 1, we multiply the fitness of each individual by \(-1\) in order to obtain a higher fitness for the individuals which are non-dominated and a lower fitness for the individuals which are dominated.

**Algorithm 1:** MaximinFitnessFunction

**Input:** \( X \) (Current population with their normalized objective values), \( P \) (population size) and, \( K \) (number of objectives).

**Output:** \( X \) (Current population with the fitness updated of each individual).

```
for \( i \leftarrow 1 \) to \( P \) do
    \text{maximal} \leftarrow -\infty;
    for \( j \leftarrow 1 \) to \( P \) do
        if \( j \neq i \) then
            \text{minimal} \leftarrow -\infty;
            for \( k \leftarrow 1 \) to \( K \) do
                if \( X[i].f[k] - X[j].f[k] < \text{minimal} \) then
                    \text{minimal} \leftarrow X[i].f[k] - X[j].f[k];
                end
            end
            if \( \text{minimal} > \text{maximal} \) then
                \text{maximal} \leftarrow \text{minimal};
            end
        end
    end
    \text{fitness}^i \leftarrow -1 * \text{maximal};
end
```

IV. SELECTION OPERATOR

When studying the maximin fitness function, we identified an important disadvantage of this approach when solving MOPs. If we take a look at the part where we get the minimum of the difference between normalized objective values of two given solutions, we can notice that if a particular objective is minimized more quickly than the others, then the remaining objectives are not considered. Let’s assume that we have a MOP with two objective functions \( f_1 \) and \( f_2 \) and that the objective \( f_1 \) is easier to optimize than objective \( f_2 \). In this case, when the maximin fitness function calculates the minimum, it will often obtain the component of objective \( f_1 \), without regarding \( f_2 \). Thus, if the maximin fitness function is incorporated into an evolutionary algorithm, it may occur that we only obtain solutions that minimize \( f_1 \), instead of finding the best possible trade-offs among the objectives, which is
our aim. Evidently, we cannot assume that the objectives can be optimized separately, since that assumption would only be reasonable when the objectives have no conflict among themselves, which makes employing a multi-objective approach unnecessary (a single-objective optimizer would be sufficient in that case).

In order to address this problem, we propose to check if the individual that we want to select is similar in at least one objective to another (selected) individual. The process to verify similarity between individuals is shown in Algorithm 2. As can be seen, we need the parameter 

\[ \text{min\_dif} \]

which indicates the minimum difference that must separate the two selected individuals in objective function space (this is similar to the niche radius adopted with fitness sharing [18]).

The complexity of Algorithm 2 is \( O(KP) \), where \( K \) is the number of objectives and \( P \) is the population size. The full selection process based on a maximin fitness function is shown in Algorithm 3 and its complexity is \( O(KP^2) \).

### Algorithm 2: IsSimilarToAny

**Input:** \( \text{min\_dif} \) (Minimum difference), \( x \) (individual), \( Y \) (population), \( P \) (population size), \( K \) (number of objectives).

**Output:** Returns 1, if the individual \( x \) is similar to any individual in the population \( Y \); otherwise, returns 0.

```plaintext
for i ← 1 to P do
    for k ← 1 to K do
        if \(|x.f[k] - Y[i].f[k]| < \text{min\_dif}\) then
            Returns 1;
        end
    end
end
Returns 0;
```

As indicated before, the search engine of a MOEA has two main goals: (i) to find solutions which 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.

The first of these goals is achieved by the mechanism described in Algorithm 3. However, we need one more mechanism, so that we can fulfill the second objective. Here, we propose to use the clustering technique described in Algorithm 4.

### V. MAXIMIN-CLUSTERING DIFFERENTIAL EVOLUTION

The approach that we propose here is called Maximin-Clustering Differential Evolution (MCDE) and is described next.

MCDE adopts the operators of differential evolution to create new individuals but the selection process is modified as follows: If the size of the population is \( P \), MCDE creates \( P \) new individuals. After that, it combines the population of parents and offspring to obtain a population of size \( 2P \). Then, MCDE uses Algorithms 3 and 4 to choose the \( P \) individuals that will take part of the following generation.

### Algorithm 3: MaximinSelection

**Input:** \( X_{\text{sorted}} \) (Sorted population from high to low, according to the fitness values), \( P \) (population size), \( K \) (number of objectives), \( S \) (the number of individuals to choose) and \( \text{min\_dif} \) (minimum difference between objectives).

**Output:** \( Y \) (individuals selected).

```
s ← 1, i ← 1;
/* Before selecting individuals, verify that there is not a similar one */
while s ≤ S AND i ≤ P do
    while IsSimilarToAny(min\_dif, X_{\text{sorted}}[i], Y, s, K) = 1 AND i ≤ P do
        i ← i + 1;
    end
    if i ≤ P then
        Y[s] ← X_{\text{sorted}}[i];
        s ← s + 1;
    end
end
/* Select only according to fitness */
i ← 1;
while s ≤ S do
    if X_{\text{sorted}}[i] has not been selected then
        Y[s] ← X_{\text{sorted}}[i];
        s ← s + 1;
    end
    i ← i + 1;
end
Returns Y;
```

In order to speed up convergence, we also propose to modify the process in which parents are selected to participate in the process of mutation and crossover as follows. Instead of randomly selecting three individuals for becoming parents (as normally done in the DE algorithm), we use a binary tournament selection for choosing the three individuals needed. At each tournament, two individuals are randomly selected and the one with the higher fitness value is chosen. Finally, Algorithm 5 shows the full algorithm of our proposed MCDE approach.

### VI. EXPERIMENTAL RESULTS

We compared our proposed MCDE with respect to two MOEAs representative of the state-of-the-art in the area:

- The Nondominated Sorting Genetic Algorithm II (NSGA-II) [17], which is a well-known MOEA whose selection mechanism is based on Pareto dominance. NSGA-II also incorporates a crowded comparison operator to produce well-distributed solutions along the Pareto front. This MOEA was chosen because is perhaps the most representative of the Pareto-based MOEAs.
- The S Metric Selection-Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [14], which bases its selection mechanism on the hypervolume performance measure [12] combined with the non-dominated
Algorithm 4: ClusteringSelection

```
Input : X_sorted (Sorted population from high to low, according to fitness), P (population size), K (number of objectives) and S (number of individuals to choose).

Output: Y (individuals selected).

/* Choose the first S individuals as centers of the clusters C */
C_j = {X_sorted[j]};
/* Do clustering */
for i ← S + 1 to P do
  if X_sorted[i] is closer to C_j then
    C_j ← C_j ∪ X_sorted[i];
  end
/* Obtain the new centers of the clusters */
for j ← 1 to S do
  μ_j ← 1 / |C_j| ∑ X[i];
end
/* Select individuals who are closest to the centers of the clusters */
for j ← 1 to S do
  if X[i] ∈ C_j is the nearest to the center μ_j then
    Y[j] ← X[i];
  end
end
Returns Y;
```

Algorithm 5: Maximin-Clustering Differential Evolution

```
Input : P (population size), g_{max} (maximum number of generations), N (number of decision variables), K (number of objective functions) Cr (crossover probability) and min_{dif} (minimum difference between objectives).

Output: Last population.

contGen ← 0;
Create a random initial population;
repeat
  for i ← 1 to P do
    Select three parents (X[i], X[i] and X[i]);
    Obtain the new individual (X_{new}[i]) from DE’s mutation and crossover operators;
  end
  Y ← X ∪ X_{new};
  MaximinFitnessFunction(Y, 2P, K);
  Y_sorted ← SortCurrentPopulation(Y);
  if The number of nondominated individuals is greater to P then
    X ← ClusteringSelection(Y_sorted, 2P, K, P);
  else
    X ← MaximinSelection(Y_sorted, 2P, K, min_{dif});
  end
  contGen ← contGen + 1;
until contGen < g_{max};
```

sorting procedure adopted in NSGA-II. This approach was chosen because it is a state-of-the-art hypervolume-based MOEA.

It is important to emphasize that we chose these algorithms because our aim was to validate the selection mechanism of our proposed approach. Although there are several MOEAs based on DE (see for example [19], [20], [21], [22]), most of them adopt a Pareto-based selection mechanism and, therefore, were not considered for our comparative study (we decided to adopt NSGA-II instead, because of its widespread use and availability (its source code is available in the public domain)). We believe that it would be more interesting to compare results with respect to a scalarization method such as MOEA/D (there is a version based on DE) [23], but this was not done here because of time constraints.

We performed 30 independent runs for each test problem. The parameters adopted for our proposed MCDE are shown in Table I (these values were empirically derived after numerous experiments). For both NSGA-II and SMS-EMOA, we adopted the parameters suggested by the authors of NSGA-II: crossover probability p_c = 0.9, mutation probability p_m = 1/n, where n is the number of decision variables, for crossover and mutation operators, n_c = 15 and n_m = 20, respectively. Finally, we used the same population size and the same value for the maximum number of generations with the three algorithms compared. All three algorithms performed the same number of objective function evaluations (for the ZDT test problems they performed 10,000 evaluations, and for the DTLZ test problems they performed 20,000 evaluations, except for ZDT4 and DTLZ3 in which they performed 36,000 and 25,000 evaluations, respectively).

<table>
<thead>
<tr>
<th></th>
<th>F</th>
<th>P_{cr}</th>
<th>min_{dif}</th>
<th>P</th>
<th>G</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>0.5</td>
<td>0.9</td>
<td>0.00001</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ZDT2</td>
<td>0.5</td>
<td>0.9</td>
<td>0.001</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ZDT3</td>
<td>0.5</td>
<td>0.5</td>
<td>0.00001</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>ZDT4</td>
<td>0.5</td>
<td>0.23</td>
<td>0.001</td>
<td>120</td>
<td>200</td>
</tr>
<tr>
<td>ZDT6</td>
<td>0.5</td>
<td>0.9</td>
<td>0.001</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>0.5</td>
<td>0.00001</td>
<td>0.0001</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>0.5</td>
<td>0.01</td>
<td>0.001</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>0.5</td>
<td>0.00001</td>
<td>0.001</td>
<td>100</td>
<td>250</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>0.5</td>
<td>0.00001</td>
<td>0.001</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.00001</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>0.5</td>
<td>0.3</td>
<td>0.00001</td>
<td>100</td>
<td>200</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>0.5</td>
<td>0.1</td>
<td>0.00001</td>
<td>100</td>
<td>200</td>
</tr>
</tbody>
</table>

**TABLE I** Parameters adopted by our proposed MCDE algorithm. F and P_{cr} are parameters used by the DE algorithm; min_{dif} is used by our proposed selection operator; P is the population size and G is the maximum number of generations.

A. Test problems

To validate our proposed MCDE, we adopted two sets of problems. The first consists of five bi-objective test problems taken from the Zitzler-Deb-Thiele suite [24]. The second
consists of seven problems having three or more objectives, taken from the Deb-Thiele-Laumanns-Zitzler (DTLZ) suite [25]. For the DTLZ test problems, we used \( k \) consists of seven problems having three or more objectives, four and five objective functions (i.e., \( M = 3, 4, \) and 5).

B. Quality indicators

To assess performance, we adopted the following indicators:

1) Generational Distance (GD). This indicator was proposed by Van Veldhuizen in [26] and, it represents how far a set \( A \) is from the Pareto front \( \mathcal{P} \). Formally, it is defined as:

\[
GD = \frac{1}{|A|} \left( \sum_{i=1}^{|A|} d_i^2 \right)^{\frac{1}{2}}
\]

(6)

where \( d_i \) is the Euclidean distance, in objective function space, between \( a_i \) and the nearest member of \( \mathcal{P} \). Lower GD values represent better approximations.

For the calculation of the GD indicator, we used the following:

- Pareto optimal sets generated by an enumerative approach for the ZDT test problems [1].
- A set of equations that describe the Pareto fronts for the DTLZ test problems 1 to 6.
- A value of the function \( g \) for the test problem DTLZ7.

2) Hypervolume indicator (\( \varphi \)). It was originally proposed by Zitzler and Thiele in [27], and it’s defined as the size of the space covered by the Pareto optimal solutions. \( \varphi \) rewards convergence towards the Pareto front as well as the maximum spread of the solutions obtained. If \( \Lambda \) denotes the Lebesgue measure, \( \varphi \) is defined as:

\[
\varphi(A, y_{\text{ref}}) = \Lambda \left( \bigcup_{y \in A} \{ y' \mid y < y' < y_{\text{ref}} \} \right)
\]

(7)

where \( y_{\text{ref}} \in \mathbb{R}^k \) denotes a reference point that should be dominated by all the Pareto optimal points. Fleischer proved in [28] that, given a finite search space and a reference point, maximizing the hypervolume indicator is equivalent to finding the Pareto optimal set. The disadvantage of this indicator is its high computational cost (the running time for calculating \( \varphi \) is exponential in the number of objective functions).

To compute \( \varphi \), we used the following reference points:

- For ZDT functions, we used \( y_{\text{ref}} = [1, 1] \).
- For function DTLZ1, we used \( y_{\text{ref}} = [y_1, \ldots, y_M] \mid y_i = 0.7 \forall i = 1, \ldots, M \).
- For functions DTLZ(2-6), we used \( y_{\text{ref}} = [y_1, \ldots, y_M] \mid y_i = 1.1 \forall i = 1, \ldots, M \).
- For function DTLZ7, we used \( y_{\text{ref}} = [y_1, \ldots, y_M] \) \( y_M = 6.1 \) and \( y_i = 1.1 \forall i = 1, \ldots, M - 1 \).

C. Results

In Table II, we can observe that our proposed MCDE outperforms both NSGA-II and SMS-EMOA, in most of the ZDT test problems (the best results are shown in boldface). As shown in Tables III and V, in the DTLZ test problems having three and five objective functions, SMS-EMOA is better than MCDE in four test problems and MCDE is better than SMS-EMOA in three test problems. As shown in Table IV, for the test problems having four objective functions, SMS-EMOA is better than MCDE in five test problems and MCDE is better than SMS-EMOA only in two. It is important to note that our proposed MCDE presents a consistent behavior when increasing the number of objectives, unlike NSGA-II whose performance quickly degrades, reaching a value of zero for the hypervolume indicator when solving three of the DTLZ test problems.

To validate the results in our experiments, we performed statistical analysis using Wilcoxon’s rank sum table. Table VII shows the results. With respect to generational distance, we can say that our algorithm, MCDE, is significantly better than SMS-EMOA in DTLZ6 and DTLZ7, with three objective functions, and also in DTLZ7, with four objective functions because the hypothesis that the medians are equal can be rejected. In most of the remaining problems, we can say that SMS-EMOA and MCDE have a similar behavior, except for ZDT6, DTLZ5, with three objective functions, DTLZ1, DTLZ3 and DTLZ4, with five objective functions because the probability that the hypothesis is true is less than 0.5. Regarding the hypervolume indicator only in DTLZ2 with five objectives, we can say that SMS-EMOA is significantly better than MCDE. Finally, we can say that SMS-EMOA and MCDE have a similar behavior in most of the ZDT and DTLZ problems with three and five objective functions.

Based on the results shown before, we claim that the performance of MCDE is competitive with respect to the performance of SMS-EMOA. However, it is important to note that the computational cost of the SMS-EMOA algorithm is considerably larger than that of MCDE. Table VI shows the CPU time, per run, required by each algorithm. In this table, we can note that MCDE needs only 1 or 2 seconds in any of the test problems adopted, even for instances having five objectives. In contrast, SMS-EMOA needs up to 11 hours, per run, for the test problems having five objectives.

This difference is due to the fact that computing the maximim fitness function is an inexpensive process (its complexity is linear with respect to the number of objectives), whereas the computation of the hypervolume is exponential with respect to the number of objectives. Thus, we argue that MCDE can be a good alternative for dealing with many-objective optimization problems, unless we can afford a very high computational cost.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a new selection operator to solve multiobjective optimization problems with a single-objective evolutionary algorithm (differential evolution in our case).
This operator takes into account the two main objectives of a MOEA: it uses a maximin technique to find solutions as close as possible to the true Pareto front and it uses a clustering technique to provide a good distribution of such solutions along the Pareto front. We chose these techniques in order to obtain a selection operator capable of solving problems of both low dimensionality (with two or three objective functions) and high dimensionality (more than three objective functions).

In our experimental study, we compared the performance of our proposed approach with respect to a state-of-the-art Pareto-based MOEA (NSGA-II) and with respect to a state-of-the-art hypervolume-based MOEA (SMS-EMOA) using standard test problems and performance measures taken from the specialized literature. Our results show that MCDE outperforms NSGA-II in all cases (low- and high-dimensionality test problems) and produces competitive results with respect to SMS-EMOA, but at a much lower computational cost.

As part of our future work, we plan to incorporate our selection operator into other evolutionary algorithms in order...
to assess the impact of the search engine in the performance of our algorithm. We also plan to perform a statistical analysis of the sensitivity of our proposed approach to its parameters. Finally, we plan to compare our algorithm with other algorithms that use techniques not based on Pareto optimality or the hypervolume indicator, such as MOEA/D [23] and maximinPSO [29]. It is important to mention, however, that maximinPSO was tested in [29] only with bi-objective optimization problems. Its authors also reported a poor performance of this approach in ZDT2.

**REFERENCES**


---

**TABLE V**

<table>
<thead>
<tr>
<th>Function</th>
<th>Objectives</th>
<th>SMS-EMOA</th>
<th>MCDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>2</td>
<td>≈ 4h</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>ZDT2</td>
<td>2</td>
<td>≈ 4h</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>ZDT3</td>
<td>2</td>
<td>≈ 4h</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>ZDT4</td>
<td>2</td>
<td>≈ 9h</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>ZDT6</td>
<td>2</td>
<td>≈ 2h</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>≈ 2m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>≈ 4m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>≈ 3m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>≈ 4m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>3</td>
<td>≈ 2m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>3</td>
<td>≈ 2m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>3</td>
<td>≈ 3m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>4</td>
<td>≈ 40m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>4</td>
<td>≈ 60m</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>4</td>
<td>≈ 40m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>4</td>
<td>≈ 40m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>4</td>
<td>≈ 20m</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>4</td>
<td>≈ 40m</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>4</td>
<td>≈ 20m</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>5</td>
<td>≈ 8h</td>
<td>≈ 1s</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>5</td>
<td>≈ 11h</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>5</td>
<td>≈ 8h</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>5</td>
<td>≈ 11h</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>5</td>
<td>≈ 6h</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>5</td>
<td>≈ 7h</td>
<td>≈ 2s</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>5</td>
<td>≈ 3h</td>
<td>≈ 1s</td>
</tr>
</tbody>
</table>

**TABLE VI**

<table>
<thead>
<tr>
<th>Function</th>
<th>Objectives</th>
<th>SMS-EMOA</th>
<th>MCDE</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>2</td>
<td>0.000000 (0.000000)</td>
<td>0.153918 (0.031912)</td>
</tr>
<tr>
<td>ZDT2</td>
<td>2</td>
<td>0.978240 (0.024770)</td>
<td>1.295810 (0.000108)</td>
</tr>
<tr>
<td>ZDT3</td>
<td>2</td>
<td>0.003487 (0.003088)</td>
<td>0.000000 (0.000002)</td>
</tr>
<tr>
<td>ZDT4</td>
<td>2</td>
<td>0.978240 (0.024770)</td>
<td>1.295810 (0.000108)</td>
</tr>
<tr>
<td>ZDT6</td>
<td>2</td>
<td>0.978240 (0.024770)</td>
<td>1.295810 (0.000108)</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>3</td>
<td>11.374256 (4.076644)</td>
<td>1.245409 (0.231560)</td>
</tr>
</tbody>
</table>

**RESULTS OBTAINED IN THE DTLZ TEST PROBLEMS WITH FIVE OBJECTIVE FUNCTIONS. WE SHOW AVERAGE VALUES OVER 30 INDEPENDENT RUNS. THE VALUES IN PARENTHESES CORRESPOND TO THE STANDARD DEVIATIONS.**

Both algorithms were implemented in the C programming language and they were executed on PCs with the same hardware and software characteristics.
TABLE VII
RESULTS OF STATISTICAL ANALYSIS APPLIED TO EXPERIMENTS THAT WE DID. K IS THE NUMBER OF OBJECTIVE FUNCTIONS. 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.

<table>
<thead>
<tr>
<th>Function</th>
<th>K</th>
<th>I</th>
<th>nsga-ii &amp; mcde</th>
<th>sms-emoa &amp; mcde</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZDT1</td>
<td>2</td>
<td>dg</td>
<td>0.888305 (0)</td>
<td>0.830253 (0)</td>
</tr>
<tr>
<td>ZDT2</td>
<td>2</td>
<td>dg</td>
<td>0.610008 (0)</td>
<td>0.748101 (0)</td>
</tr>
<tr>
<td>ZDT3</td>
<td>2</td>
<td>dg</td>
<td>0.95192 (0)</td>
<td>0.678982 (0)</td>
</tr>
<tr>
<td>ZDT4</td>
<td>2</td>
<td>dg</td>
<td>0.795846 (0)</td>
<td>0.935192 (0)</td>
</tr>
<tr>
<td>ZDT5</td>
<td>2</td>
<td>dg</td>
<td>0.876631 (0)</td>
<td>0.706164 (0)</td>
</tr>
<tr>
<td>ZDT6</td>
<td>2</td>
<td>dg</td>
<td>0.569220 (0)</td>
<td>0.853382 (0)</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>3</td>
<td>dg</td>
<td>0.958731 (0)</td>
<td>0.750587 (0)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>3</td>
<td>dg</td>
<td>0.795846 (0)</td>
<td>0.888300 (0)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>3</td>
<td>dg</td>
<td>0.395207 (0)</td>
<td>0.845779 (0)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>3</td>
<td>dg</td>
<td>0.251881 (0)</td>
<td>0.966263 (0)</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>3</td>
<td>dg</td>
<td>0.395267 (0)</td>
<td>0.923441 (0)</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>3</td>
<td>dg</td>
<td>0.970516 (0)</td>
<td>0.217908 (0)</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>3</td>
<td>dg</td>
<td>0.610008 (0)</td>
<td>0.251881 (0)</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>4</td>
<td>dg</td>
<td>0.735362 (0)</td>
<td>0.695211 (0)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>4</td>
<td>dg</td>
<td>0.510598 (0)</td>
<td>0.122353 (0)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>4</td>
<td>dg</td>
<td>0.066284 (0)</td>
<td>0.125876 (0)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>4</td>
<td>dg</td>
<td>0.539510 (0)</td>
<td>0.596984 (0)</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>4</td>
<td>dg</td>
<td>0.869994 (0)</td>
<td>0.371077 (0)</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>4</td>
<td>dg</td>
<td>0.917090 (0)</td>
<td>0.711262 (0)</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>4</td>
<td>dg</td>
<td>0.482517 (0)</td>
<td>0.347828 (0)</td>
</tr>
<tr>
<td>DTLZ1</td>
<td>5</td>
<td>dg</td>
<td>0.917090 (0)</td>
<td>0.711262 (0)</td>
</tr>
<tr>
<td>DTLZ2</td>
<td>5</td>
<td>dg</td>
<td>0.510598 (0)</td>
<td>0.122353 (0)</td>
</tr>
<tr>
<td>DTLZ3</td>
<td>5</td>
<td>dg</td>
<td>0.066284 (0)</td>
<td>0.125876 (0)</td>
</tr>
<tr>
<td>DTLZ4</td>
<td>5</td>
<td>dg</td>
<td>0.539510 (0)</td>
<td>0.596984 (0)</td>
</tr>
<tr>
<td>DTLZ5</td>
<td>5</td>
<td>dg</td>
<td>0.869994 (0)</td>
<td>0.371077 (0)</td>
</tr>
<tr>
<td>DTLZ6</td>
<td>5</td>
<td>dg</td>
<td>0.917090 (0)</td>
<td>0.711262 (0)</td>
</tr>
<tr>
<td>DTLZ7</td>
<td>5</td>
<td>dg</td>
<td>0.482517 (0)</td>
<td>0.347828 (0)</td>
</tr>
</tbody>
</table>

[21] F. Xue, A. C. Sanderson, and R. J. Graves, “Pareto-based Multi-
[22] T. Robic and B. Filipic, “DEMO: Differential Evolution for Multiobjec-
[23] Q. Zhang and H. Li, “MOEA/D: A Multiobjective Evolutionary Algo-
[26] D. A. Veldhuizen, Multiobjective Evolutionary Algorithms: Classi-
[27] E. Zitzler and L. Thiele, “Multiobjective Optimization Using Evo-
[28] M. Fleischer, “The Measure of Pareto Optima. Applications to Multi-