An Algorithm Based on Differential Evolution for Multi-Objective Problems

Luis Vicente Santana-Quintero and Carlos A. Coello Coello
CINVESTAV-IPN (Evolutionary Computation Group)
Electrical Engineering Department, Computer Science Section
Av. IPN No. 2508, Col. San Pedro Zacatenco
México D.F. 07300, MÉXICO
lvspenny@hotmail.com
ccoello@cs.cinvestav.mx

Abstract- This paper presents a new multi-objective evolutionary algorithm based on differential evolution. The proposed approach adopts a secondary population in order to retain the nondominated solutions found during the evolutionary process. Additionally, the approach also incorporates the concept of ϵ-dominance to get a good distribution of the solutions retained. The main goal of this work was to keep the fast convergence exhibited by Differential Evolution in global optimization when extending this heuristic to multi-objective optimization. We adopted standard test functions and performance measures reported in the specialized literature to validate our proposal. Our results are compared with respect to another multi-objective evolutionary algorithm based on differential evolution (Pareto Differential Evolution) and with respect to two approaches that are representative of the state-of-the-art in the area: the NSGA-II and ϵ-MOEA.

1 Introduction

Most real world problems involve the simultaneous optimization of two or more (often conflicting) objectives. The solution of such problems (called "multi-objective") is different from that of a single-objective optimization problem. The main difference is that multi-objective optimization problems normally have not one but a set of solutions which are all equally good.

In the past, a wide variety of evolutionary algorithms (EA's) have been used to solve multi-objective optimization problems [5]. However, from the several types of EAs available, few researchers have attempted to extend Differential Evolution (DE) [25] to solve multi-objective optimization problems. DE has been very successful in the solution of a variety of continuous (single-objective) optimization problems in which it has shown a great robustness and a very fast convergence. These are precisely the characteristics of DE that make it attractive to extend it to solve multi-objective optimization problems.

In this paper, we propose a new multi-objective evolutionary algorithm based on differential evolution. Our approach uses an external archive in which the relaxed form of Pareto dominance known as ϵ-dominance is adopted. We also adopt a hybrid selection scheme in which a random and an elitist scheme are interleaved. Additionally, a constraint-handling technique is also provided for our approach. The proposed mechanisms are able to overcome some of the limitations of differential evolution when dealing with multi-objective optimization problems (such limitations are mainly related to the loss of diversity produced by the high selection pressure of the differential evolution algorithm).

The remainder of this paper is organized as follows. Section 2 provides some basic definitions related to multi-objective optimization. Section 3 provides a brief introduction to differential evolution. In Section 4, we provide a quick review of the most relevant previous related work, including a more detailed description of Pareto Differential Evolution (PDE), against which we compared results. Section 5 describes our proposed approach. Our comparison of results is provided in Section 6.

2 Definitions

Some basic definitions related to this work are introduced next.

2.1 Multi-objective optimization problem (MOP)

The multi-objective optimization problem can be formally defined as the problem of finding:

$\vec{x}^* = [x_1^*, x_2^*, \ldots, x_n^*]^T$ which satisfies the *m* inequality constraints:

$$g_i(\vec{x}) \leq 0; i = 1, \ldots, m$$

the p equality constraints:

$$h_i(\vec{x}) = 0; i = 1, \ldots, p$$

and optimizes the vector function:

$$f_1(\vec{x}), f_2(\vec{x}), \ldots, f_k(\vec{x})$$

In other words, we aim to determine from among the set $\mathcal{F}$ of all numbers which satisfy the constraints those that yield the optimum values for all the k-objective functions. The constraints define the feasible region $\mathcal{F}$ and any point $\vec{x}$ in the feasible region is called a feasible solution.

2.2 Pareto Dominance

Pareto dominance is formally defined as follows:

A vector $\vec{u} = (u_1, \ldots, u_k)$ *is said to dominate* $\vec{v} = (v_1, \ldots, v_k)$ *if and only if* $\vec{u}$ *is partially less than* $\vec{v}$*, i.e.,* $\forall i \in (1, \ldots, k), u_i \leq v_i \wedge \exists i \in (1, \ldots, k) : u_i < u_i$

In words, this definition says that a solution dominates another one, only if it's strictly better in at least one objective, and not worse in any of them. So, when we are comparing two different solutions A & B, there are 3 possibilities:

- A dominates B
- A is dominated by B
- A and B are nondominated

2.3 Pareto Optimality

The formal definition of *Pareto optimality* is provided next:

A solution $\overrightarrow{x_u} \in \mathcal{F}$ *(where* $\mathcal{F}$ *is the feasible region) is said to be Pareto optimal if and only if there is no* $\overrightarrow{x_v} \in \mathcal{F}$ *for which* $v = f(x_v) = (v_1, \ldots, v_k)$ *dominates* $u = f(x_u) = (u_1, \ldots, u_k)$, where k is the number of objectives.

In words, this definition says that $\overrightarrow{x}_u$ is Pareto optimal if there exists no feasible vector $\overrightarrow{x_v}$ which would decrease some criterion without causing a simultaneous increase in at least one other criterion (assuming minimization).

Unfortunately, this does not provide us a single solution (in decision variable space), but a set of solutions called *Pareto Optimal Set*. The vectors that correspond to the solutions included in the Pareto optimal set are called *nondominated*.

2.4 Pareto Front

When all nondominated solutions are plotted in objective space, the nondominated vectors are collectively known as the *Pareto Front*. Formally:

For a given MOP $\overrightarrow{f}(x)$ and Pareto optimal set P^*, the Pareto front (FP^*) is defined as:

$$FP^* := \{\overrightarrow{f} = [f_1(x), \ldots, f_k(x)] | x \in P^*\}$$

3 Differential Evolution

Differential Evolution [25, 26] is a relatively recent heuristic designed to optimize problems over continuous domains. In DE, each decision variable is represented in the chromosome by a real number. As in any other evolutionary algorithm, the initial population of DE is randomly generated, and then evaluated. After that, the selection process takes place. During the selection stage, three parents are chosen and they generate a single offspring which competes with a parent to determine who passes to the following generation. DE generates a single offspring (instead of two as the genetic algorithm) by adding the weighted difference vector between two parents to a third parent. In the context of single-objective optimization, if the resulting vector yields a lower objective function value than a predetermined population member, the newly generated vector replaces the vector with respect to which it was compared. In addition, the best parameter vector $X_{best,G}$ is evaluated for every generation G in order to keep track of the progress that is made during the minimization process. Formally, the process is described as follows:

For each vector $\overrightarrow{x_{i,G}}; i = 0, 1, 2, \ldots, N-1.$, a trial vector $\overrightarrow{v}$ generated according to:

$$\overrightarrow{v} = \overrightarrow{x_{r1,G}} + F \cdot (\overrightarrow{x_{r2,G}} - \overrightarrow{x_{r3,G}})$$

with $r_1, r_2, r_3 \in [0, N-1]$, integer an mutually different, and $F > 0$.

The integers r_1, r_2 and r_3 are chosen randomly from the interval [0, $N-1$] and are different from i. F is a real and constant factor which controls the amplification of the differential variation $(\overrightarrow{x_{r2,G}} - \overrightarrow{x_{r3,G}})$. Figure 1 shows a two dimensional example that illustrates the different vectors which play a role in DE.

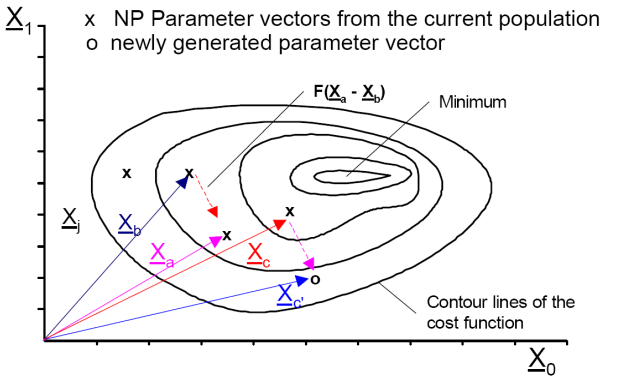

Figure 1: Two dimensional example of an objective function showing its contour lines and the process for generating offspring.

In order to increase the diversity of the parameter vectors, the following vector is adopted:

$$\overrightarrow{u} = (u_1, u_2, \ldots, u_n)^T$$

with:

$$u_j = \begin{cases} v_j, & \text{if } (x \in [0,1]) \leq CR \text{ or } j = i; \\ x_{ij,G}, & \text{otherwise.} \end{cases}$$

where i,j = 1, 2, ..., n; and CR is a user-defined *crossover* rate in the range [0, 1]. In other words, a certain sequence of the vector elements of $\overrightarrow{u}$ are identical to the elements of $\overrightarrow{v}$, and at least one of the elements of $\overrightarrow{u}$ acquires the original values of $\overrightarrow{x_{i,G}}$.

In order to decide whether the new vector $\overrightarrow{u}$ shall become a population member at generation $G+1$, it is compared to $\overrightarrow{x_{i,G}}$. If $\overrightarrow{u}$ yields a lower objective function value than $\overrightarrow{x_{i,G}}$, then it replaces it (i.e., $\overrightarrow{x_{i,G}} = \overrightarrow{u}$). Otherwise, the old value $\overrightarrow{x_{i,G}}$ is retained.

4 Previous Related Work

Currently, there are relatively few papers that propose ways of extending DE to handle multiple objectives. The most representative of them are briefly discussed next:

4.1 Pareto-based Differential Evolution Approach

The Pareto-Based Differential Evolution was proposed in [18]. In this algorithm, Differential Evolution is extended to multi-objective optimization by incorporating a nondominated sorting and ranking selection procedure proposed by Deb et al. [8, 10]. Once the new candidate is obtained using DE operators, the new population is combined with the existing parents population and then the best members of

the combined population (parents plus offspring) are chosen. This algorithm is not compared with respect to any other approach and is tested on 10 different unconstrained problems performing 250,000 evaluations. The authors indicate that the approach has difficulties to converge to the true Pareto front in two problems (Kursawe's test function [16] and ZDT4 [32]).

4.2 Multi-Objective Differential Evolution (MODE)

MODE was proposed in [31]. This algorithm uses a variant of the original DE, in which the best individual is adopted to create the offspring. A Pareto-based approach is introduced to implement the selection of the best individual. If a solution is dominated, a set of nondominated individuals can be identified and the "best" turns out to be any individual (randomly picked) from this set. Also, the authors adopt $(\mu + \lambda)$ selection, Pareto ranking and crowding distance in order to produce and maintain well-distributed solutions. MODE is used to solve five high dimensionality unconstrained problems with 250,000 evaluations and the results are compared only to those obtained by SPEA [33].

4.3 Differential Evolution for Multi-Objective Optimization

It was proposed in [3]. This algorithm uses the single-objective Differential Evolution strategy with an aggregating function to solve bi-objective problems. A single optimal solution is obtained after N iterations using both a *Penalty Function Method* (to handle the constraints) and the *Weighing Factor Method* (to provide the importance of each objective from the user's perspective) [7] to optimize a single value. The authors present results for two bi-objective problems and compare them with respect to a simple GA. The authors indicate that the DE algorithm provides the exact optimum with a lower number of evaluations than the GA.

4.4 Vector Evaluated Differential Evolution for Multi-Objective Optimization (VEDE)

VEDE was proposed in [20]. It is a parallel, multi-population Differential Evolution algorithm, which is inspired by the Vector Evaluated Genetic Algorithm (VEGA) [22] approach. A number M of subpopulations are considered in a ring topology. Each population is evaluated using one of the objective functions of the problem, and there is an exchange of information among the populations through the migration of the best individuals. VEDE is validated using four bi-objective unconstrained problems and was compared to VEGA. The authors indicate that the proposed approach outperformed VEGA in all cases.

4.5 Nondominated Sorting Differential Evolution (NSDE)

This approach was proposed in [12]. It is a simple modification of the NSGA-II [10]. The only difference between this approach and the NSGA-II is in the method for generating new individuals. The NSGA-II uses a real-coded crossover and mutation operator, but in the NSDE, these operators were replaced with the operators of Differential Evolution. New candidates are generated using the DE/current-to-rand/1 strategy. NSDE is used to solve rotated problems with a certain degree of rotation on each plane. The results of the NSDE outperformed those produced by the NSGA-II.

4.6 Generalized Differential Evolution (GDE)

This approach was proposed in [15]. GDE extends the selection operation of the basic DE algorithm for constrained multi-objective optimization. The basic idea in this selection rule is that the trial vector is required to dominate the old population member used as a reference either in constraint violation space or in objective function space. If both vectors are feasible and nondominated with respect to each other, the one residing in a less crowded region is chosen to become part of the population of the next generation. GDE its validated using five bi-objective unconstrained problems. Results are compared with respect to the NSGA-II and SPEA [33]. The authors report that the performance of GDE is similar to the NSGA-II, but they claim that their approach requires a lower CPU time. GDE is able to outperform SPEA in all the test functions adopted.

4.7 Differential Evolution for Multi-Objective Optimization (DEMO)

DEMO was proposed in [21]. This algorithm combines the advantages of DE with the mechanisms of Pareto-based ranking and crowding distance sorting. DEMO only maintains one population and it is extended when newly created candidates take part immediately in the creation of the subsequent candidates. This enables a fast convergence towards the true Pareto front, while the use of nondominated sorting and crowding distance (derived from the NSGA-II [10]) of the extended population promotes the uniform spread of solutions. DEMO is compared in five high-dimensionality unconstrained problems outperforming in some problems to the NSGA-II, PDEA [1], PAES [14], SPEA [33] and MODE [31].

4.8 Pareto Differential Evolution (PDE)

PDE was proposed in [2]. Also, it is important to mention that there is another version of this approach (called the Self-Adaptive Pareto Differential Evolution) in which self-adaptive crossover and mutation operators are adopted [1]. The PDE algorithm was chosen for comparison, so it is described next in more detail:

1. The initial population is initialized according to a Gaussian Distribution $N(0.5, 0.15)$.
2. The step length parameter F, is generated from a Gaussian Distribution $N(0, 1)$.
3. Reproduction is undertaken only among nondominated solutions at each generation.

4. The boundary constraints are preserved either by reversing the sign if the variable is less than 0 or by successively subtracting 1 if it is greater than 1 until the variable is within its boundaries.

5. Offspring are placed into the population if they dominate the main parent.

The algorithm works as follows. An initial population is generated at random from a Gaussian distribution. All dominated solutions are removed from the population. The remaining nondominated solutions are retained for reproduction. If the number of nondominated solutions exceeds some threshold, a distance metric relation ($D(x)$, as defined in equation 1) is used to remove those parents which are very close to each other. Three parents are selected at random. A child is generated from the three parents and is placed into the population if it dominates the first selected parent; otherwise a new selection process takes place.

A maximum number of nondominated solutions in each generation was set to 50. If this maximum is exceeded, the following nearest neighbor distance function is adopted:

$$D(x) = \frac{(min\|x - x^i\| + min\|x - x^j\|)}{2} \qquad (1)$$

where $x \neq x^i \neq x^j$.

That is, the nearest neighbor distance is the average Euclidean distance between the two closest points. The nondominated solution with the smallest neighbor distance is removed from the population until the total number of nondominated solutions is reduced to 50.

After analyzing the diverse algorithms available in the specialized literature, we identified Pareto ranking and crowding distance as two of the most common and effective mechanisms adopted. We also found that most of the previous approaches were tested in unconstrained test problems with only two objectives and that little attention was paid to the importance of the selection criteria (which are a key issue to regulate the high selection pressure of differential evolution) and that none of these previous proposals adopted ϵ-dominance [17] to generate a set of well-spread solutions. These were the main motivations for our algorithmic design, which is presented in the next section.

5 Our Proposed Algorithm

The pseudocode of our proposed approach (called ϵ-MyDE) is shown in Algorithm 1. Our approach keeps two populations: the main population (which is used to select the parents) and a secondary (external) population, in which we adopt the concept of ϵ-dominance to retain the nondominated solutions found and to distribute them in an uniform way. The concept of ϵ-dominance does not allow two solutions with a difference less than ϵ_i in the i-th objective to be nondominated with respect to each other, thereby allowing a good spread of solutions.

ϵ-MyDE uses real numbers representation, where each chromosome is a vector of real numbers (each number corresponds to a decision variable of the problem). We also incorporate a constraint-handling mechanism that allows infeasible solutions to intervene during recombination. This helps to solve in a more efficient way highly constrained multi-objective optimization problems.

Algorithm 1 Proposed Algorithm: ϵ - MyDE

```
1: Initialize vectors of the population P
2: Evaluate the cost of each vector
3: for i = 0 to G do
4:    repeat
5:       Select three distinct vectors randomly
6:       Perform crossover using DE scheme
7:       Perform mutation
8:       Evaluate objective values
9:       if offspring is better than main parent then
10:         replace main parent in the population
11:      end if
12:   until population is completed
13:   Identify nondominated solutions in the population
14:   Add nondominated solutions into secondary population
15: end for
```

At the beginning of the evolutionary process, our approach randomly initializes all the individuals of the population. Each decision variable is normalized within its allowable bounds. The expression that we adopt is the following:

$$x_i = \text{LI}_i + U(0,1) \cdot (\text{LS}_i - \text{LI}_i)$$

where: $j = 0, 1, 2, \ldots, v-1$. (v = total number of decision variables), LI_j and LS_j are the upper and lower bounds of the variable j, respectively. $U(0,1)$ generates a random number between 0 and 1 with a uniform distribution.

Our approach has two selection mechanisms that are activated based on the total number of generations and a parameter called $sel_2 \in$ (0.2 - 1), which regulates the selection pressure. For example, if sel_2 = 0.6 and the total number of generations is $G_{max} = 200$, this means that during the first 120 generations (60 % of G_{max}), a random selection will be adopted, and during the last 80 generations an elitist selection will be adopted.

$$\text{Type of Selection} = \begin{cases} \text{Random}, & gen < (sel_2 * G_{max}) \\ \text{Elitist}, & \text{otherwise} \end{cases}$$

where:
gen = generation number
G_{max} = total number of generations

In both selections (random and elitist), a single parent is selected as a reference. This parent is used to compare the offspring generated by three different parents. This mechanism guarantees that all the parents of the main population will be reference parents for only one time during the generating process. Both types of selection are described next:

1. **Random Selection.-** 3 different parents are randomly selected from the primary population.

2. **Elitist Selection.-** 3 different parents are selected from the secondary population such that they maintain a close distance f_{near} among them. In Figure 2, we illustrate the f_{near} parameter. If no parent exist which fulfills this condition, we randomly select another parent from the secondary population.

$$f_{near} = \frac{\sqrt{\sum_{i=0}^{FUN} (X_{i,max} - X_{i,min})^2}}{2^{FUN}}$$

where:
FUN = number of objective functions
$X_{i,max}$ = upper bound of $i-th$ objective function of the secondary population
$X_{i,min}$ = lower bound of $i-th$ objective function of the secondary population

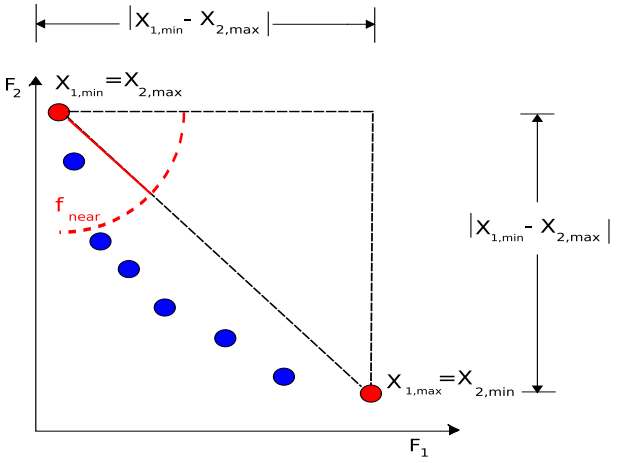

Figure 2: Parameter f_{near} for a bi-objective optimization problem

In Figure 3, we illustrate (for a bi-objective problem) the main difference between the random and the elitist selection once the main parent has been selected, and it is required to select two more parents. The candidate solutions are those inside the dotted circle.

Recombination in our approach is performed using the following procedure. For each parent vector $\overrightarrow{p_i}$; $i = 0, 1, 2, \ldots, P-1$(P = population), the offspring vector $\overrightarrow{h}$ is generated as:

$$h_j = \begin{cases} p_{r1,j} + F \cdot (p_{r2,j} - p_{r3,j}), & \text{if } x < p_{crossover}; \\ p_{ref,j}, & \text{otherwise.} \end{cases}$$

where: $j = 0, 1, 2, \ldots, var-1$. ($var$ = number of variables for each solution vector), $x \in U(0,1)$, $p_{r1}, p_{r2}, p_{r3} \in [0, P-1]$, are integers and mutually different. $F > 0$. The integers r_1, r_2 and r_3 are the indexes of the selected parents randomly chosen from the interval [0, N-1] and ref is the index of the reference parent. F is a constant factor (a real number) which controls the amplification of the differential variation $p_{r2,j} - p_{r3,j}$. The optimal value of F for most of the functions lies within the range from 0.4 to 1.0 [26].

Differential Evolution doesn't use an specific mutation operator, since such operator is embedded within its recombination operator. However, in multi-objective optimization problems, we found it necessary to provide an additional mutation operator to allow a better exploration of the search space (mainly in constrained problems). We adopted uniform mutation for that sake:

$$h_j = \begin{cases} \text{LI}_j + U(0,1) \cdot (\text{LS}_j - \text{LI}_j)], & \text{if } x < p_{mutation}; \\ p_j, & \text{otherwise.} \end{cases}$$

where: $j = 0, 1, 2, \ldots, v-1$. (v = number of variables). LI_j and LS_j are the lower an upper bounds for the variable j, respectively.

Once a child has been generated, it is compared with respect to the reference parent, against which it competes to determine who passes to the following generation.

It is important to mention that in our approach, we normalize the constraints so that their value ranges between 0 and 1. This normalization is transparent for user (the algorithm does this without requiring any input from the user). This normalization mechanism is described next:

For each constraint C_i, two different variables CS and CI store its upper and lower value. Therefore, whenever a constraint is required, the following expression is adopted:

$$NC_i = \frac{\text{C}_i - \text{CI}_i}{\text{CS}_i - \text{CI}_i}$$

The rules of comparison between a child and its parent are the following:

Version for unconstrained problems

* *if parent dominates child*, the parent is chosen
* *if child dominates parent*, the child is chosen
* *if both are nondominated with respect to each other*, perform a $flip$ (0.5) to determine who passes to the following generation.

Version for constrained problems

* *if parent is infeasible and child is infeasible*, the solution that is closest to the feasible region is selected.
* *if parent is feasible and child is infeasible*, the child is chosen if and only if the child is at least at a distance of 0.1 of the feasible region and a $flip$ (0.5) returns true. Otherwise, the father is chosen.
* *if parent is infeasible and child is feasible*, the parent is chosen if and only if the parent is at least at a distance of 0.1 of the feasible region and a $flip$ (0.5) returns true. Otherwise, the child is chosen.
* *if parent is feasible and child is feasible*, Pareto dominance is verified between them.

Note that the scheme previously described allows some infeasible solutions to intervene during the recombination

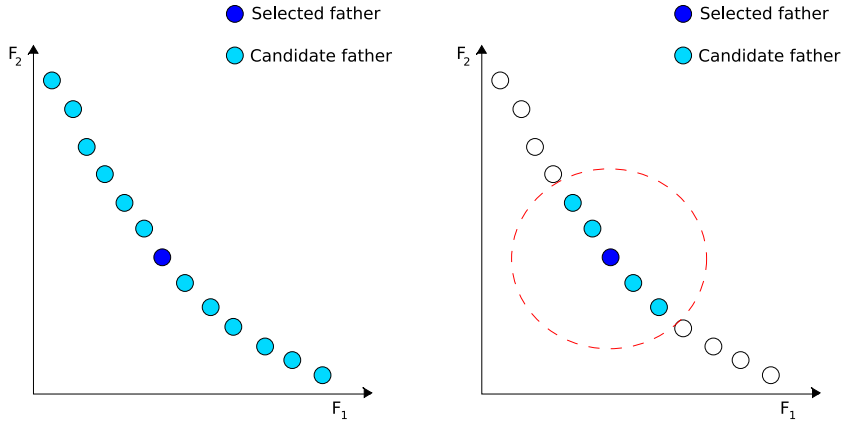

Figure 3: Graphical illustration of (1) random selection (right) and (2) elitist selection (left)

process. We found that this sort of scheme is particularly useful when dealing with highly constrained problems. It has been previously shown that maintaining infeasible solutions that lie in the frontier between the feasible and infeasible regions, helps to obtain better solutions in highly constrained problems (particularly when dealing with equality constraints) [19, 23].

For producing a well-distributed set of nondominated solutions, we adopted a relaxed form of dominance called ϵ-*dominance*, which is defined as follows [17]:

Let $f, g \in \Re^m$, then $\vec{f}$ is said to ϵ-dominate $\vec{g}$ for some $\epsilon > 0$, denoted as $f \prec_\epsilon g$, if and only if for all $i \in \{1, \ldots,$ m $\}$.

$$(1 - \epsilon) \cdot f_i \leq g_i$$

As indicated before, our proposed approach uses an external archive (also called secondary population). In order to include a solution into this archive, it is compared with respect to each member already contained in the archive using ϵ-dominance. This procedure has been used in another multi-objective evolutionary algorithm called ϵ-MOEA proposed in [9]. The procedure is described next.

Every solution in the archive is assigned an identification array ($\mathbf{B} = (B_1, B_2, \ldots, B_M)^T$, where M is the total number of objectives) as follows:

$$B_j(f) = \begin{cases} (\lfloor (f_j - f_j^{min})/\epsilon_j \rfloor), & \text{for minimizing } f_j; \\ (\lceil (f_j - f_j^{min})/\epsilon_j \rceil), & \text{for maximizing } f_j. \end{cases}$$

where: f_j^{min} is the minimum possible value of the j-th objective and ϵ_j is the allowable tolerance in the j-th objective [17]. The identification array divides the whole objective space into hyper-boxes, each having ϵ_j size in the j-th objective. With the identification arrays calculated for the offspring c_1 and each archive member a, we use the procedure illustrated in Figure 4 and described next:

1. If the identification array B_a of any archive member a dominates that of the offspring c_i, then it means that the offspring is ϵ-dominated by this archive member and so the offspring *is not accepted*. This is case (a) in Figure 4.

2. If B_{c_i} of the offspring dominates the B_a of any archive member a, the archive member is deleted and the offspring *is accepted*. This is case (b) in Figure 4.

   If neither of the above two cases occur, then it means that the offspring is ϵ-nondominated with respect to the archive contents. There are two further possibilities in this case:

   (a) If the offspring shares the same **B** vector with an archive member (meaning that they belong to the same hyper-box), then they are first checked for the usual nondomination. If the offspring dominates the archive member or the offspring is nondominated with respect to the archive member but is closer to the **B** vector (in terms of the Euclidian distance) than the archive member, then the offspring *is retained*. This is case (c) in Figure 4.

   (b) In the event of an offspring not sharing the same **B** vector with any archive member, the offspring *is accepted*. This is case (d) in Figure 4.

Using the above procedure, we can guarantee the generation of a well-distributed set of nondominated solutions. Also, the value of ϵ adopted regulates the size of the external archive. Thus, there is no need to pre-fix an upper limit on the size of the archive as done in most traditional multi-objective evolutionary algorithms.

One of the difficulties of this approach is the choice of the ϵ-vector, as it dictates the cardinality of the obtained nondominated set. If a large value of ϵ_i is chosen, fewer solutions will be obtained, and vice versa. However, this difficulty can also become an advantage if considered from a decision-maker's point of view. Particularly, for high dimensionality problems (with many objectives), there usually exist a large number of Pareto-optimal solutions. By choosing an appropriate ϵ-vector, the decision-maker can specify the minimum difference he would like to achieve in each objective. This would allow him to obtain a handful of Pareto-optimal solutions with certain (minimum desirable) trade-offs among them. Finally, Section 7 contains our conclusions and some possible paths for future research.

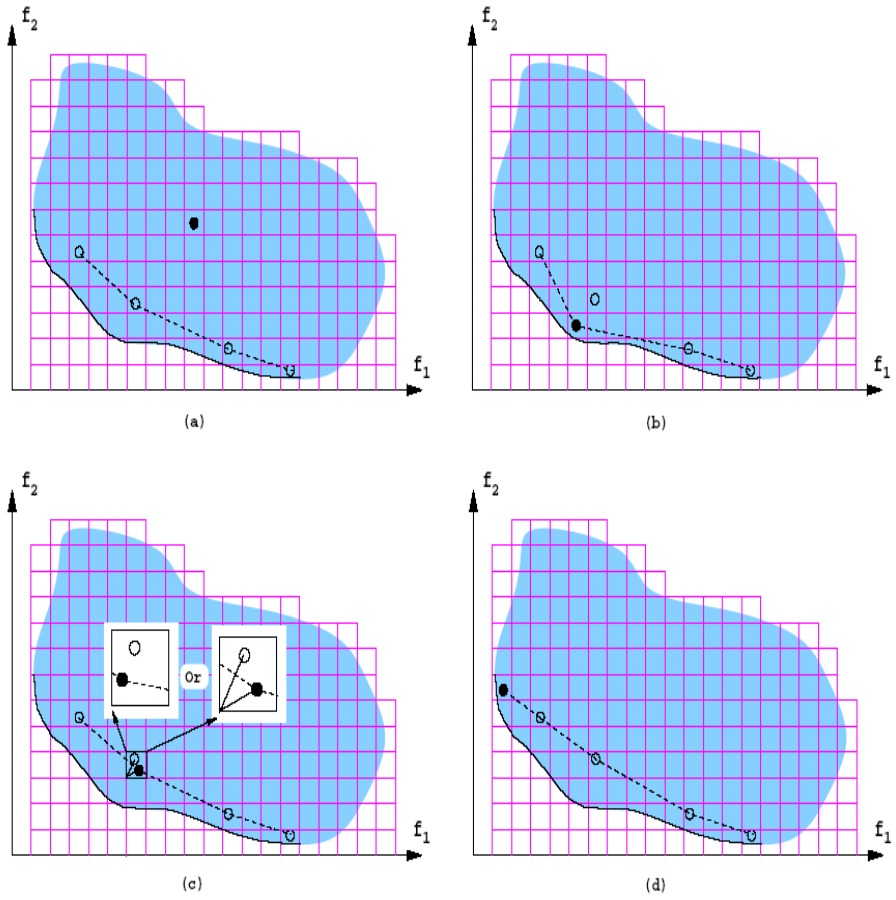

Figure 4: Four cases of inserting a child into the external archive

6 Comparison of Results

In order to validate our proposed approach, we adopted standard test functions and metrics reported in the specialized literature. The specific metrics are described next.

1.- Error Ratio (ER) [28] The error ratio metric reflects the number of members of PF_{known} (the approximation of the Pareto front obtained by the approach to be validated) that are in PF_{true} (the true Pareto front of the problem, normally obtained by enumeration). This metric is mathematically represented by:

$$ER \triangleq = \frac{\sum_{i=1}^{n} e_i}{n} \tag{2}$$

where (n) is the number of vectors in PF_{known} and

$$e_i = \begin{cases} 0 & \text{if vector } i, \forall\, i = (1, \ldots, n) \in PF_{true}, \\ 1 & \text{otherwise} \end{cases} \tag{3}$$

For example, ER = 0 indicates that every vector reported by the algorithm in PF_{known} is actually in PF_{true}; ER = 1 indicates that none of the elements of PF_{known} are in PF_{true}.

2.- Generational Distance (GD) [29, 30] This metric is a value representing in the average how "far" PF_{known} is from PF_{true} and is defined as:

$$GD = \frac{(\sum_{i=1}^{n} d_i^p)^{\frac{1}{p}}}{n}$$

where n is the number of vectors in PF_{known}, p = 2, and d_i is the Euclidean distance (in objective function space) between each vector and the nearest member of PF_{true}. A result of 0 indicates $PF_{true} = PF_{known}$; any other value indicates PF_{known} deviates from PF_{true}.

3.- Spacing (S) [24] Here, one desires to measure the spread (distribution) of vectors throughout PF_{known}. This metric is mathematically represented by:

$$S = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(\bar{d} - d_i\right)^2}$$

where n is the number of vectors in PF_{known}, and $d_i = min_j(|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$, $i, j = 1, \ldots, n$ is the mean of all d_i. A value of 0 indicates that all members of PF_{known} are equidistantly spaced.

4.- Two Set coverage (CS) [32] This metric can be termed relative coverage comparison of two sets. CS is defined as the mapping of the order pair (X', X'') to the interval [0,1] as follows:

$$CS(X', X'') = \frac{|\{a'' \in X''; \exists a' \in X' : a \succeq a''\}|}{|X''|}$$

Consider $X', X'' \subseteq X'$as two sets of decision vectors. If all points in X' dominate or are equal to all points in X", then by definition CS = 1. CS = 0 implies the opposite. Observe that it is not a distance measure of how close these sets are as that is a different metric.

Our results are compared with respect to those generated by another multi-objective evolutionary algorithm based on Differential Evolution: PDE [2]. Additionally, we also compared results with respect to two approaches representative of the state-of-the-art in the area: the NSGA-II [10] and ϵ-MOEA (this was the first multi-objective evolutionary algorithm to incorporate the concept of ϵ-dominance). These approaches are briefly described next.

NSGA-II Nondominated Sorting Genetic Algorithm - II (NSGA - II) was proposed in [8, 10]. The NSGA-II uses a $(\mu + \lambda)$ selection (μ parents are combined with its λ offspring and ranked based on Pareto dominance. Also, it adopts a crowding comparison operator which keeps diversity without requiring any additional parameters from the user. The NSGA-II is one of the most competitive multi-objective evolutionary algorithms available today and it's often used as a reference to determine the performance of new approaches.

ϵ-MOEA This approach was proposed in [9]. It uses the concept of ϵ-dominance introduced in [17] and it has been found to be a very competitive multi-objective evolutionary algorithm.

PDE Pareto Differential Evolution (PDE) was proposed in [2], and is one of the most representative multi-objective extensions of differential evolution. This approach was described in Section 4.8.

The parameter settings adopted for all the algorithms compared are summarized in Table 1. In Table 1, P refers to the population at each generation, G_{max} is the total number of generations (or iterations) to be performed. Note that all the algorithms perform the same number of objective function evaluations: 5,000 for unconstrained problems and 25,000 for constrained problems. NP is the size of the secondary population. Note however that neither of ϵ-MyDE or ϵ-MOEA adopt this parameter, since the size of their secondary populations is controlled by the value of ϵ. F is a parameter applicable only to differential evolution. P_c and P_m are the crossover and mutation rates, respectively.

Next, we show the plot of all the nondominated solutions generated by the different algorithms in each of the test functions. In all cases, we generated the PF_{true} of the problems using exhaustive enumeration so that we could make a graphical comparison of the quality of the solutions produced by our approach.

For each example, two tables are shown:

Parameter	ϵ-MyDE	NSGA-II	PDE	ϵ-MOEA
P	100	100	100	100
NP	100 (aprox)	100	100	100 (aprox)
G_{max}	50 ó 250	50 ó 250	50 ó 250	50 ó 250
P_c	0.95	0.8	nr	1.0
P_m	1/N	1/N	nr	1/N
F	0.5	nr	N(0, 1)	nr

donde: N = variable number
d = objective number
nr = not required
G_{max} = 50 for unconstrained problems
G_{max} = 250 for constrained problems

Table 1: Parameters used by the algorithms compared.

1. The first one corresponds to the statistical results from applying the performance metrics over the twenty runs performed by each of the algorithms.

2. The second one is the Set Coverage (SC) metric, applied to the union of 20 runs of each algorithm to obtain 4 different files (one by each algorithm). These four files were compared using CS. We first computed CS $(alg1, alg2) = Table_{_{alg1,alg2}}$ and then CS $(alg2, alg1) = Table_{_{alg2,alg1}}$. In the *dominated row*, a value close to 0 means that the algorithm outperformed the others; analogously, in the *dominates column* a value close to 1 indicates a superior performance.

MOP1. Proposed by Kursawe [16].

Minimize:

$$\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}))$$

where:

$$\begin{aligned} f_1(\vec{x}) &= \sum_{i=1}^{n-1} \left(-10e^{(-0.2)\sqrt{x_i^2+x_{i+1}^2}}\right) \\ f_2(\vec{x}) &= \sum_{i=1}^{n} (|x_i|^a + 5\sin(x_i^b)) \end{aligned}$$

with:

$$\begin{aligned} a &= 0.8 \\ b &= 3 \\ n &= 3 \end{aligned}$$

and:

$$-5 \leq x_1, x_2, x_3 \leq 5$$

Our first problem has several disconnected and asymmetrical areas in solution space. Its PF_{true} consists of three disconnected Pareto curves [4].

The Pareto front generated by each of the algorithms are shown in Figure 5. The statistical results obtained from applying the performance metrics are shown in Table 2.

Graphically, we can see that ϵ-MyDE and ϵ-MOEA both cover all of PF_{true}. The NSGA-II does not reach the edges of the front. Regarding the Spacing metric, we can see that all the algorithms were able to obtain well-distributed solutions.

Numerically, the performance metrics adopted indicate that ϵ-MyDE is the algorithm that obtained the best results regarding Error Ratio and Generational Distance. Regarding Spacing, the NSGA-II slightly outperformed the other approaches.

When analyzing Set Coverage (SC), we can see that ϵ-MyDE had the best results, because is the one that dominated more solutions of the other algorithms while being less dominated by the others. Thus, we can conclude that our ϵ-MyDE is the best overall performer for this test function.

MOP2. Proposed by Deb [6].

Minimize:

$$\vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}))$$

where:

$$\begin{aligned} f_1(\vec{x}) &= x_1 \\ f_2(\vec{x}) &= \frac{g(x_2)}{x_1} \end{aligned}$$

with:

$$g(x_2) = 2 - e^{-(\frac{x_2-0.2}{0.004})^2} - 0.8e^{-(\frac{x_2-0.6}{0.4})^2}$$

and:

$$0.1 \leq x_1, x_2 \leq 1$$

Our second problem has a convex PF_{true}, and P_{true} is disconnected. Note that this problem has a local front to which an algorithm can be easily attracted.

The Pareto front generated by all the algorithms compared are shown in Figure 6 and statistical results of the performance metrics adopted are shown in Table 3.

Graphically, we can see that all the algorithms get close to PF_{true} and produce a well-distributed set of solutions. Note however, that PDE was unable to reach PF_{true} in several of its runs.

Regarding the performance metrics adopted, ϵ-MyDE was the approach the produced the best results with respect to Error Ratio, Generational Distance and Spacing.

With respect to Set Coverage (SC), ϵ-MyDE has the best results, because is the one that dominates more solutions of the other algorithms and is less dominated by others. So, we can conclude that our ϵ-MyDE was the best overall performer in this test function.

MOP3. Proposed by Kita [13]

Maximize:

$$\vec{f}(x) = (f_1(x, y), f_2(x, y))$$

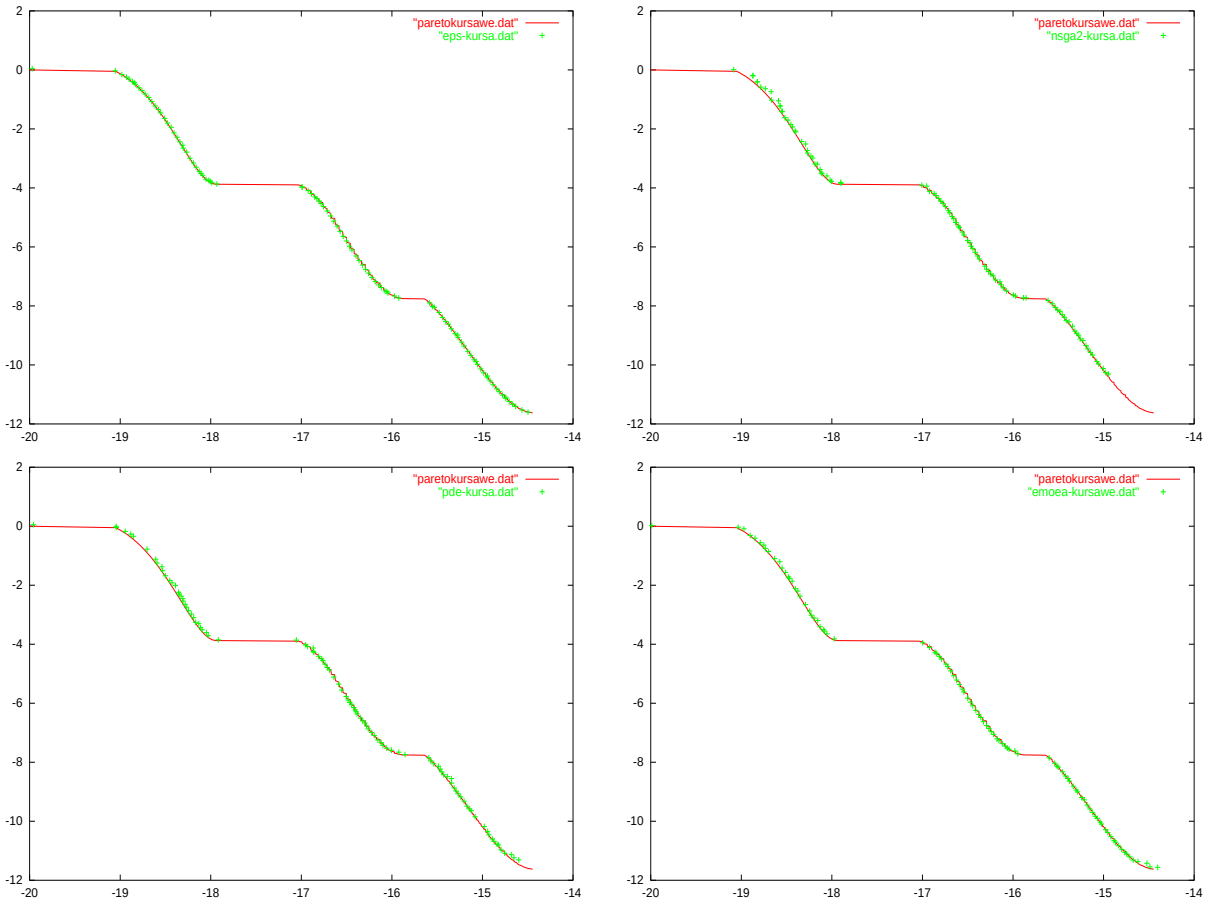

Figure 5: Pareto front generated by ϵ-MyDE (top - left), NSGA-II (top - right), PDE (bottom-left) and ϵ-MOEA (bottom - right) for MOP1.

Metric	Algorithm	average	best	worst	St. Dev.
ER	e-MyDE	**0.15792013**	**0.0654206**	**0.23301**	**0.04288708**
	NSGA-II	0.4575	0.31	0.64	0.0907208
	PDE	0.5085	0.25	0.64	0.0958219
	ϵ-MOEA	0.2820835	0.16326	1	0.17783033
GD	e-MyDE	**0.00327069**	**0.00288871**	**0.00374484**	**0.00025872**
	NSGA-II	0.00416653	0.00308047	0.0061631	0.00086198
	PDE	0.00467552	0.00366814	0.00667008	0.000966
	ϵ-MOEA	0.09699642	0.00304755	1.87613	0.41876465
S	e-MyDE	0.09448982	0.0548549	**0.109133**	**0.01243547**
	NSGA-II	**0.06180859**	**0.047978**	0.11722	0.01663756
	PDE	0.10522477	0.0687954	0.145438	0.0224935
	ϵ-MOEA	0.10113374	0.0564897	0.126934	0.01358711

SC	ϵ-MyDE	NSGA-II	PDE	ϵ-MOEA	Dominates
ϵ-MyDE		0.824752	0.825743	0.615924	**0.755473**
NSGA-II	0.234375		0.633663	0.375063	0.414367
PDE	0.238511	0.676733		0.408112	0.441118
ϵ-MOEA	0.323529	0.712871	0.729703		0.588701
They are dominated	**0.26547167**	0.73811867	0.729703	0.46636633	

Table 2: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC) for MOP1. The best performance measure values are shown in **boldface**.

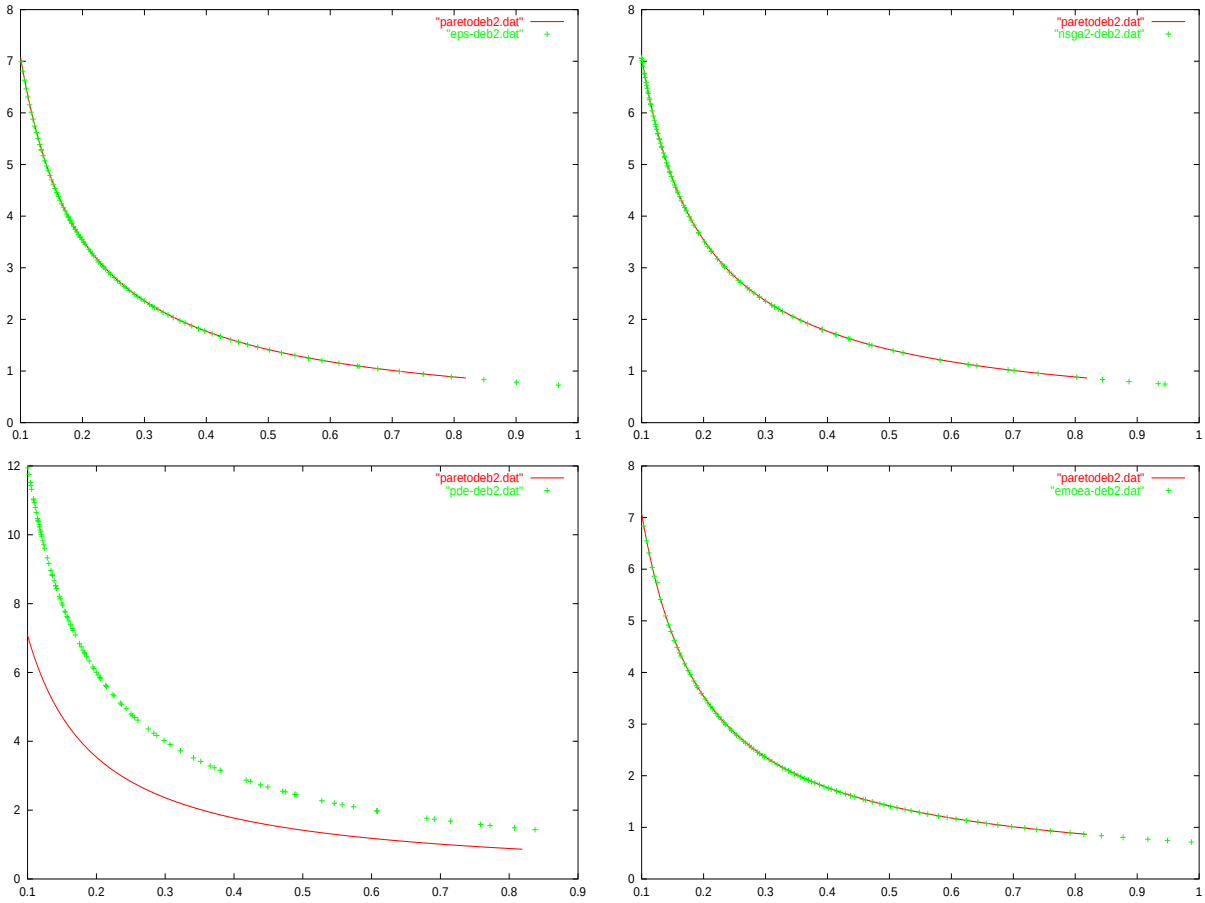

Figure 6: Pareto front generated by ϵ-MyDE (top - left), NSGA-II (top - right), PDE (bottom-left) and ϵ-MOEA (bottom - right) for MOP2.

Metric - Algorithm		average	best	worst	desv. Estándar
ER	e-MyDE	**0.005804**	**0**	**0.02**	**0.00788824**
	NSGA-II	0.42	**0**	1	0.49375042
	PDE	0.85	**0**	1	0.36634755
	ϵ-MOEA	0.454029	**0**	1	0.50551394
GD	e-MyDE	**0.00728447**	0.00184096	**0.0506528**	**0.01420113**
	NSGA-II	0.07853142	0.00288569	0.205945	0.09408827
	PDE	0.16741538	0.00204314	0.20732	0.05819178
	ϵ-MOEA	0.03319034	**0.00116984**	0.0783318	0.03670135
S	e-MyDE	0.08512233	0.0319032	0.5076	0.1420395
	NSGA-II	**0.03992726**	0.0296285	**0.0562794**	**0.00955353**
	PDE	0.05213913	**0.0284482**	0.0851937	0.01131328
	ϵ-MOEA	0.07010082	0.0471254	0.0947282	0.02001717

SC	ϵ-MyDE	NSGA-II	PDE	ϵ-MOEA	Dominates
e-MyDE		0.556931	0.952475	0.648549	**0.7193183**
NSGA-II	0.0413223		0.94604	0.592814	0.5267254
PDE	0.00729217	0.411881		0.516352	0.3118417
ϵ-MOEA	0.0291687	0.439109	0.90247		0.4569159
They are dominated	**0.02592772**	0.469307	0.93366167	0.585905	

Table 3: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC) for MOP2.

where:

$$\begin{aligned} f_1(x,y) &= -x^2 + y, \\ f_2(x,y) &= \frac{1}{2}x + y + 1 \end{aligned}$$

subject to:

$$\begin{aligned} 0 &\geq \frac{1}{6}x + y - \frac{13}{2}, \\ 0 &\geq \frac{1}{2}x + y - \frac{15}{2}, \\ 0 &\geq 5x + y - 30 \end{aligned}$$

and:

$$0 \leq x, y$$

Our third problem has a disconnected, concave PF_{true} and P_{true} is also disconnected. Note however, that P_{true} is in the boundary between the feasible and the infeasible region.

The Pareto front generated by all the algorithms compared are shown in Figure 7 and the statistical results obtained from applying the performance metrics are shown in Table 4.

Graphically, we can see that all algorithms get close the PF_{true}. However, only ϵ-MyDE covers all the Pareto front while obtaining a well-distributed set of solutions.

Regarding the performance metrics, we can see that our ϵ-MyDE obtained the best results with respect to Error Ratio and Generational Distance, and ϵ-MOEA was the best with respect to Spacing.

With respect to Set Coverage (SC), ϵ-MyDE had the best results. Thus, we can conclude that our approach was the best overall performer in this test function.

MOP4. Proposed by Tamaki [27]

Maximize:

$$\vec{f}(x) = (f_1(x,y,z), f_2(x,y,z), f_3(x,y,z))$$

donde:

$$\begin{aligned} f_1(x,y,z) &= x, \\ f_2(x,y,z) &= y, \\ f_3(x,y,z) &= z \end{aligned}$$

subject to:

$$0 < x^2 + y^2 + z^2$$

and:

$$0 \leq x, y, z \leq 1$$

Our fourth problem has a connected PF_{true} with a curved surface. Its P_{true} is also connected and forms a curved surface. It is a 3-objective problem that has been used before by several researchers.

The Pareto front generated by all the algorithms compared are shown in Figure 8 and the statistical results obtained from applying the performance metrics are shown in Table 5.

Graphically, we can see that almost all the algorithms are able to reach the true Pareto front of this problem. However, the NSGA-II only covers a portion of the Pareto surface, whereas the others cover most of it.

Regarding the performance metrics adopted, the NSGA-II obtained the best results with respect to Error Ratio, Generational Distance and Spacing.

With respect to set coverage (SC), ϵ-MOEA is the winner, closely followed by our ϵ-MyDE. Although based on the performance metrics is difficult to assess who is the winner in this case, graphically, it is clear that both ϵ-MOEA and ϵ-MyDE had a better performance than the two other approaches.

MOP5. Proposed by Golinski [11].

The objective of Golinski's Speed Reducer problem (see Figure 9) is to find the minimum of a gear box volume f (and, hence, its minimum weight), subject to several constraints. The problem is illustrated in Figure 9 and there are seven design variables, x_1 to x_7, which represent:

x_1 width of the gear face, cm

x_2 teeth module, cm

x_3 number of pinion teeth

x_4 shaft 1 length between bearings, cm

x_5 shaft 2 length between bearings, cm

x_6 diameter of shaft 1, cm

x_7 diameter of shaft 2, cm

Mathematically, the problem is specified as follows:

Minimize:

$$\vec{f}(x) = (f_1(\vec{x}), f_2(\vec{x}))$$

where:

$$\begin{aligned} f_1(\vec{x}) &= 0.7854 x_1 x_2^2 (3.3333 x_3^2 + 14.9334 x_3 - 43.0934) \\ &\quad -1.5079(x_6^2 + x_7^2)x_1 + 7.477(x_6^3 + x_7^3) \\ &\quad +0.7854(x_4 x_6^2 + x_5 x_7^2) \\ f_2(\vec{x}) &= \frac{\sqrt{(\frac{745 \cdot x_4}{x_2 x_3})^2 + 1.69e^7}}{0.1 x_6^3} \end{aligned}$$

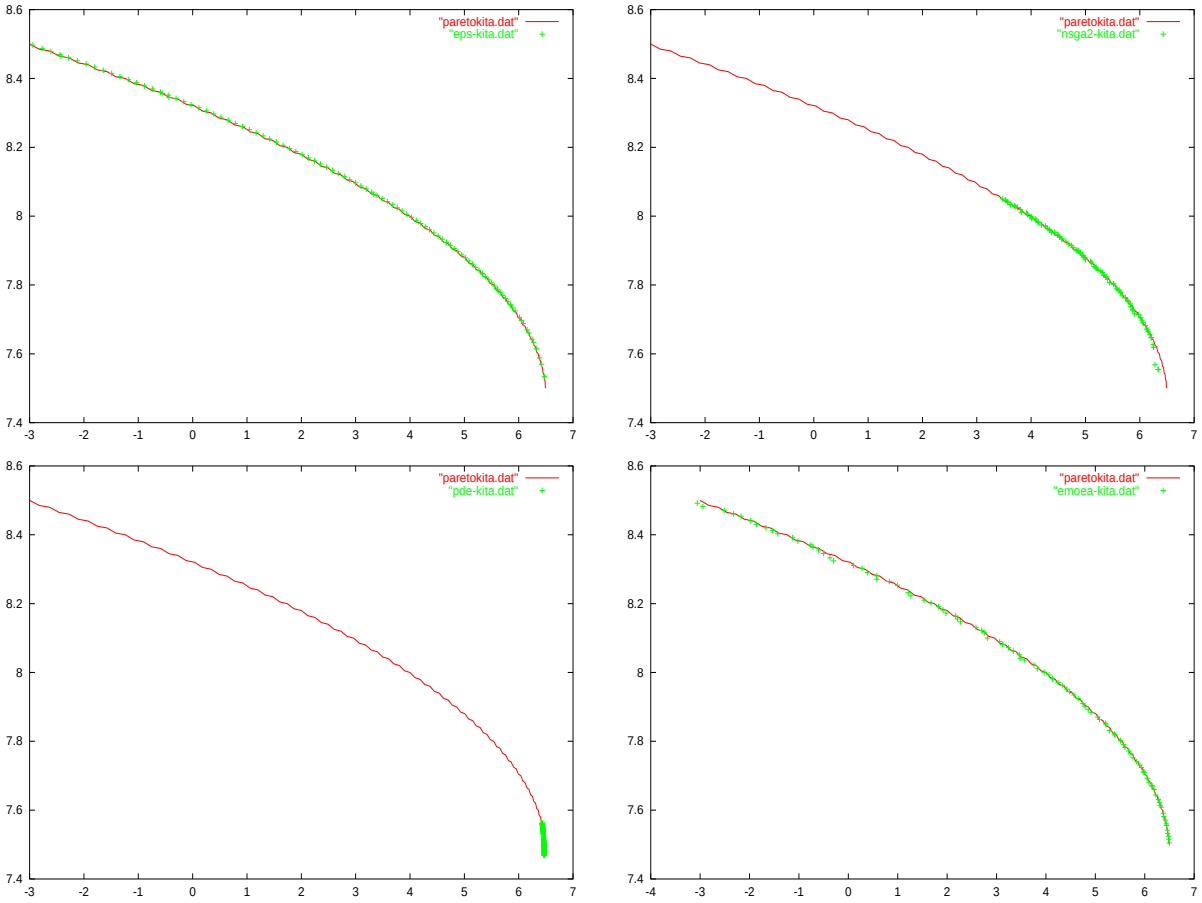

Figure 7: Pareto front generated by ϵ-MyDE (top - left), NSGA-II (top - right), PDE (bottom-left) and ϵ-MOEA (bottom - right) for MOP3.

Metric - Algorithm		average	best	worst	St. Dev.
ER	e-MyDE	**0.01640604**	**0**	**0.0550459**	**0.01635306**
	NSGA-II	0.316	0.19	0.53	0.09394287
	PDE	0.8395	0.35	1	0.21211901
	ϵ-MOEA	0.2500851	0.15625	0.333333	0.04454198
GD	e-MyDE	**0.00491203**	0.0014577	0.0464165	0.0097998
	NSGA-II	0.01470336	0.00136745	0.0902511	0.02729836
	PDE	0.00576845	**0.0003873**	**0.032941**	**0.00769517**
	ϵ-MOEA	0.00631189	0.00220538	0.0476246	0.0104118
S	e-MyDE	0.07363313	0.0394301	0.492726	0.09905315
	NSGA-II	0.0925237	0.00705687	0.56773	0.1688492
	PDE	**0.01728939**	**3.6065E-05**	**0.2736**	**0.06115304**
	ϵ-MOEA	0.07712804	0.0458646	0.492715	0.09907925

SC	ϵ-MyDE	NSGA-II	PDE	ϵ-MOEA	Dominates
e-MyDE		0.806436	0.928498	0.879053	**0.871329**
NSGA-II	0.0427757		0.666667	0.62069	0.443377
PDE	0.00522814	0.0965347		0.0761709	0.059311
ϵ-MOEA	0.039924	0.55099	0.951132		0.514015
They are dominated	**0.02930928**	0.48465357	0.84876567	0.52530463	

Table 4: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC) for MOP3.

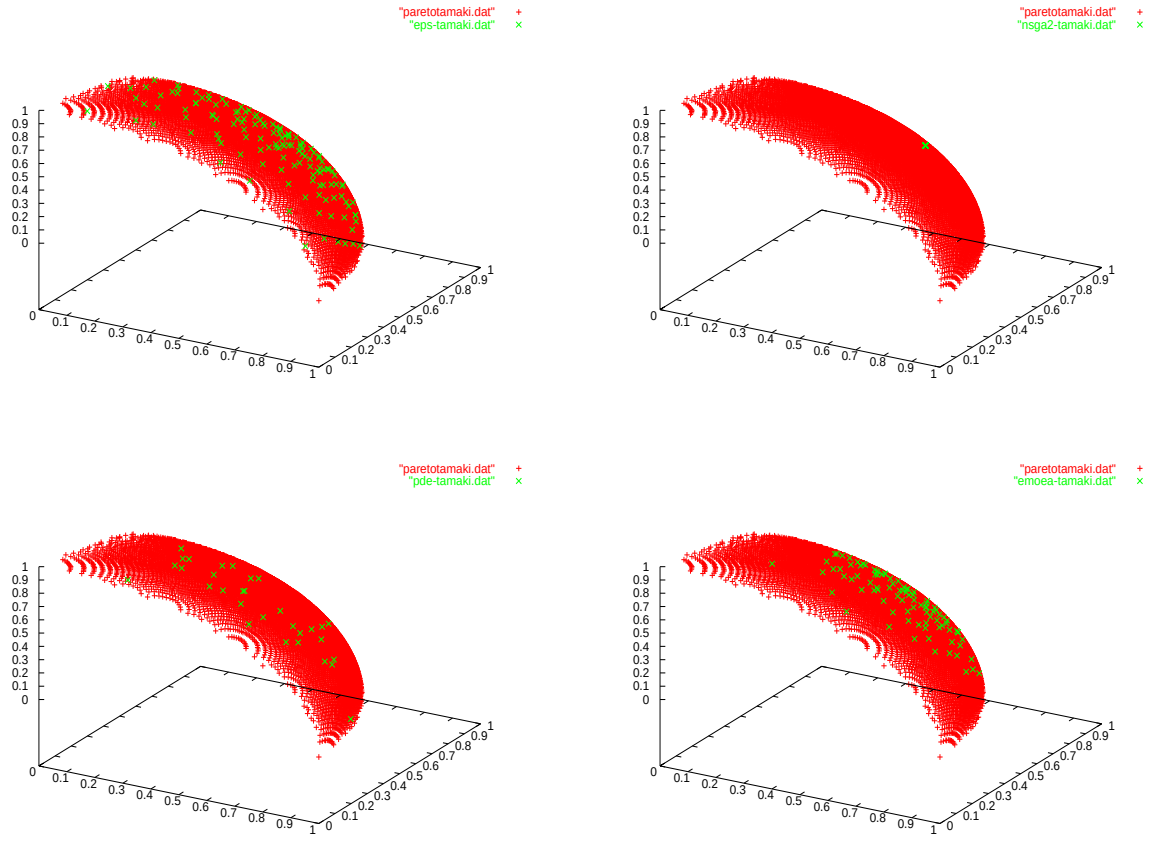

Figure 8: Pareto front generated by ϵ-MyDE (top - left), NSGA-II (top - right), PDE (bottom-left) and ϵ-MOEA (bottom - right) for MOP4.

Metric - Algorithm		average	best	worst	St. Dev.
ER	e-MyDE	0.53778185	0.478261	0.592593	0.03600738
	NSGA-II	**0**	**0**	**0**	**0**
	PDE	0.93140125	0.814815	1	0.0536189
	ϵ-MOEA	0.249779	0.142857	0.345794	0.05573555
GD	e-MyDE	0.00087087	0.00077159	0.00103772	7.1952E-05
	NSGA-II	**0.00035469**	**0.00023439**	**0.00050671**	**7.5268E-05**
	PDE	0.00804467	0.00586278	0.011823	0.00163697
	ϵ-MOEA	0.00074238	0.00056286	0.00105992	0.0001399
S	e-MyDE	0.04267574	0.0381671	0.0493826	0.0031751
	NSGA-II	**4.8229E-05**	**1.4082E-06**	**0.00021099**	**5.7565E-05**
	PDE	0.09345934	0.0654136	0.14195	0.02128756
	ϵ-MOEA	0.04332862	0.0371789	0.0519563	0.00372295

SC	ϵ-MyDE	NSGA-II	PDE	ϵ-MOEA	Dominates
e-MyDE		0	0.735154	0.0704441	0.268532
NSGA-II	0.00579822		0.118765	0.0010209	0.041861
PDE	0.0204871	0		0.0117407	0.010742
ϵ-MOEA	0.175106	0	0.732779		**0.302628**
They are dominated	0.06713044	**0**	0.52889933	0.02773523	

Table 5: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC) for MOP4.

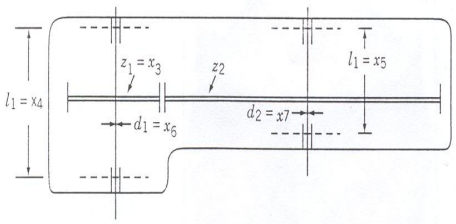

Figure 9: Problem MOP5

and:

$$
\begin{aligned}
0 &\geq \frac{1}{x_1 \cdot x_2^2 \cdot x_3} - \frac{1}{27}, \\
0 &\geq \frac{1}{x_1 \cdot x_2^2 \cdot x_3^2} - \frac{1}{397.5}, \\
0 &\geq \frac{x_3^3}{x_2 \cdot x_3 \cdot x_6^4} - \frac{1}{1.93}, \\
0 &\geq \frac{x_5^3}{x_2 \cdot x_3 \cdot x_7^4} - \frac{1}{1.93}, \\
0 &\geq x_2 \cdot x_3 - 40, \\
0 &\geq \frac{x_1}{x_2} - 12, \\
0 &\geq 5 - \frac{x_1}{x_2}, \\
0 &\geq 1.9 - x_4 + 1.5x_6, \\
0 &\geq 1.9 - x_5 + 1.1x_7, \\
1300 &\geq f_2, \\
1100 &\geq \frac{\sqrt{(\frac{745 \cdot x_5}{x_1 \cdot x_2})^2 + 1.575e^8}}{0.1 \cdot x_6^3}
\end{aligned}
$$

and:

$$
\begin{aligned}
2.6 &\leq x_1 \leq 3.6, \\
0.7 &\leq x_2 \leq 0.8, \\
17 &\leq x_3 \leq 28 \\
7.3 &\leq x_4, x_5 \leq 8.3, \\
2.9 &\leq x_6 \leq 3.9, \\
5.0 &\leq x_7 \leq 5.5
\end{aligned}
$$

The Pareto front generated by all the algorithms compared are shown in Figure 10 and the statistical results obtained from applying the performance metrics are shown in Table 6.

Graphically, we can see that all the algorithms were able to converge either on the true Pareto front or very close to it, but they had difficulties with the nearly horizontal part of the front. Our ϵ-MyDE was the only approach able to cover this nearly horizontal portion of the front.

Regarding the performance metrics, we can see that the NSGA-II obtained the best results with respect to Error Ratio and Spacing, and ϵ-MyDE was the best with respect to generational distance.

Regarding Set Coverage (SC), ϵ-MOEA had the best results, followed by our ϵ-MyDE. Again, in this case is difficult to determine who was the best overall performer, but graphically, ϵ-MOEA and ϵ-MyDE seem to have produced the best results.

7 Conclusions and Future Work

We have introduced an approach that uses Differential Evolution to solve both unconstrained and constrained multiobjective optimization problems. The high convergence rate that characterizes the Differential Evolution algorithm was controlled using two elitist selection schemes. Our proposed approach was able to produce results that are very competitive with respect to other approaches that are representative of the state-of-the-art in the area (the NSGA-II, PDE and ϵ-MOEA).

The constraint-handling scheme adopted in our algorithm allowed a successful exploration of the search space even in the presence of problems whose optimum lies in the boundary between the feasible and infeasible regions. Finally, the use of ϵ-dominance introduced the capability of controlling the convergence of our approach while achieving a good spread of solutions.

As part of our future work, we plan to experiment with different variations of the ϵ-dominance mechanism, aiming to facilitate the way of computing the ϵ values. It is also of our interest to experiment with different mechanisms (e.g., different mutation operators) to control the high selection pressure of the Differential Evolution algorithm.

Acknowledgments

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Section of the Electrical Engineering Department at CINVESTAV-IPN. The second author acknowledges support from CONACyT project number 45683.

Bibliography

[1] Hussein A. Abbass. The Self-Adaptive Pareto Differential Evolution Algorithm. In *Congress on Evolutionary Computation (CEC'2002)*, volume 1, pages 831–836, Piscataway, New Jersey, May 2002. IEEE Service Center.

[2] Hussein A. Abbass and Ruhul Sarker. The Pareto Differential Evolution Algorithm. *International Journal on Artificial Intelligence Tools*, 11(4):531–552, 2002.

[3] B.V. Babu and M. Mathew Leenus Jehan. Differential Evolution for Multi-Objective Optimization. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 4, pages 2696–2703, Canberra, Australia, December 2003. IEEE Press.

[4] Carlos A. Coello Coello and Arturo Hernández Aguirre. Design of Combinational Logic Circuits through an Evolutionary Multiobjective Optimization

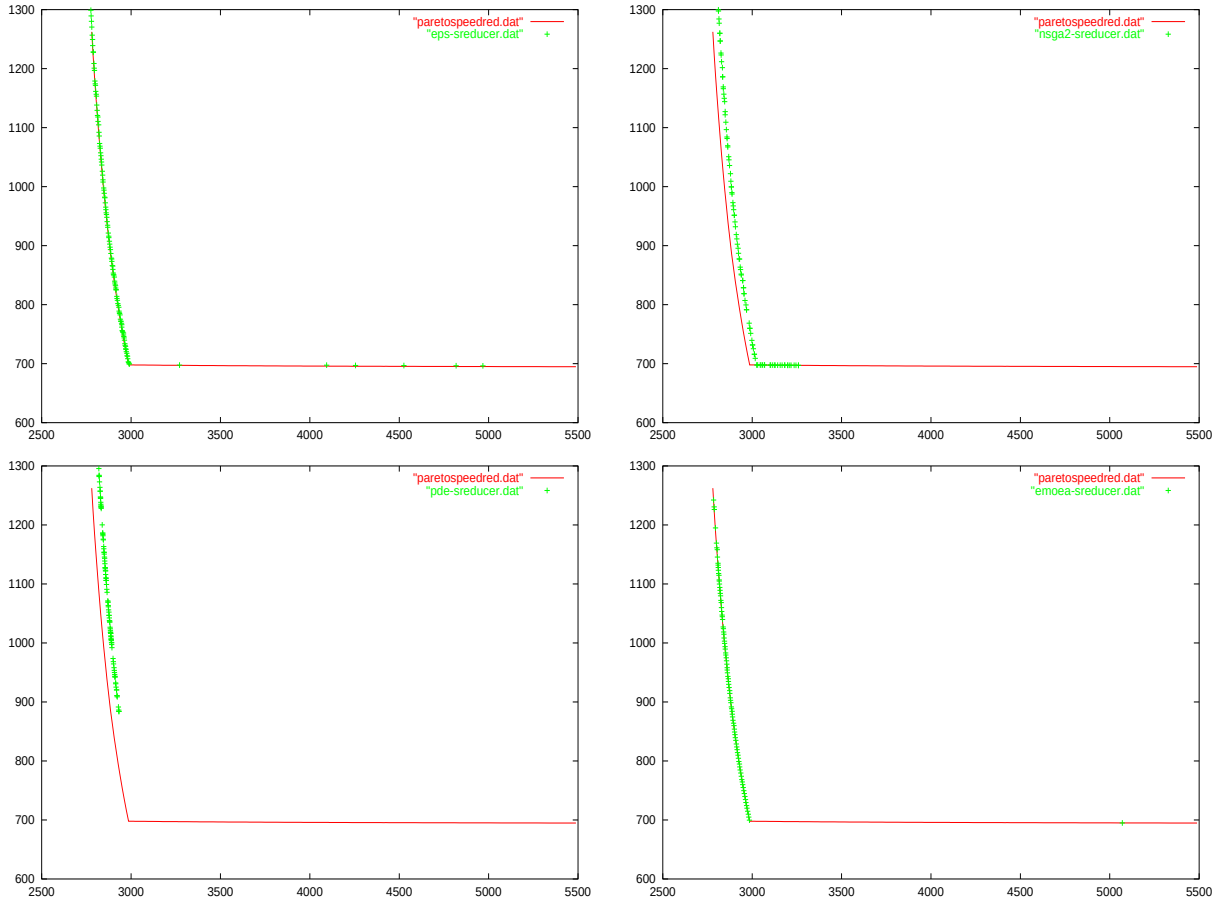

Figure 10: Pareto front generated by ϵ-MyDE (top - left), NSGA-II (top - right), PDE (bottom-left) and ϵ-MOEA (bottom - right) for MOP5.

Metric - Algorithm		average	best	worst	St. Dev.
ER	e-MyDE	1	1	**1**	**0**
	NSGA-II	**0.7825**	**0.06**	**1**	0.27466487
	PDE	1	1	**1**	**0**
	ϵ-MOEA	1	1	**1**	**0**
GD	e-MyDE	**0.8778645**	**0.794507**	**0.971234**	0.04478237
	NSGA-II	10.3952045	1.04607	32.8294	11.7194163
	PDE	9.71233435	0.952927	34.7247	9.4525265
	ϵ-MOEA	0.8952482	0.825196	0.984609	**0.04333157**
S	e-MyDE	54.6245215	2.46762	110.915	29.5975387
	NSGA-II	**6.631766**	3.20602	14.1835	**3.79596756**
	PDE	6.8780055	1.92143	**13.9599**	3.8825971
	ϵ-MOEA	52.8384655	**1.75852**	228.307	90.2445227

SC	ϵ-MyDE	NSGA-II	PDE	ϵ-MOEA	Dominates
e-MyDE		0.599505	0.821287	0.135014	0.518602
NSGA-II	0.0226777		0.79703	0.00045	0.2733859
PDE	0.56389	0.578218		0.313231	0.485113
ϵ-MOEA	0.953336	0.608911	0.935149		**0.83246533**
They are dominated	0.51330123	0.59554467	0.85115533	**0.149565**	

Table 6: Error Ratio(ER), Generational Distance (GD), Spacing (S) and Set Coverage (SC) for MOP5.

Approach. *Artificial Intelligence for Engineering, Design, Analysis and Manufacture*, 16(1):39–53, January 2002.

[5] 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.

[6] Kalyanmoy Deb. Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. *Evolutionary Computation*, 7(3):205–230, Fall 1999.

[7] Kalyanmoy Deb. *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, Chichester, UK, 2001. ISBN 0-471-87339-X.

[8] Kalyanmoy Deb, Samir Agrawal, Amrit Pratab, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In Marc Schoenauer, Kalyanmoy Deb, Günter Rudolph, Xin Yao, Evelyne Lutton, Juan Julian Merelo, and Hans-Paul Schwefel, editors, *Proceedings of the Parallel Problem Solving from Nature VI Conference*, pages 849–858, Paris, France, 2000. Springer. Lecture Notes in Computer Science No. 1917.

[9] Kalyanmoy Deb, Manikanth Mohan, and Shikhar Mishra. Towards a Quick Computation of Well-Spread Pareto-Optimal Solutions. In Carlos M. Fonseca, Peter J. Fleming, Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele, editors, *Evolutionary Multi-Criterion Optimization. Second International Conference, EMO 2003*, pages 222–236, Faro, Portugal, April 2003. Springer. Lecture Notes in Computer Science. Volume 2632.

[10] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[11] J. Golinski. Optimal synthesis problems solved by means oj nonlinear programming and random methods. *Journal of Mechanisms*, 5:287 – 309, 1970.

[12] Antony W. Iorio and Xiaodong Li. Solving rotated multi-objective optimization problems using differential evolution. In *AI 2004: Advances in Artificial Intelligence, Proceedings*, pages 861–872. Springer-Verlag, Lecture Notes in Artificial Intelligence Vol. 3339, 2004.

[13] Hajime Kita, Yasuyuki Yabumoto, Naoki Mori, and Yoshikazu Nishikawa. Multi-Objective Optimization by Means of the Thermodynamical Genetic Algorithm. In Hans-Michael Voigt, Werner Ebeling, Ingo Rechenberg, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature—PPSN IV*, Lecture Notes in Computer Science, pages 504–512, Berlin, Germany, September 1996. Springer-Verlag.

[14] Joshua D. Knowles and David W. Corne. Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. *Evolutionary Computation*, 8(2):149–172, 2000.

[15] Saku Kukkonen and Jouni Lampinen. An Extension of Generalized Differential Evolution for Multi-objective Optimization with Constraints. In *Parallel Problem Solving from Nature - PPSN VIII*, pages 752–761, Birmingham, UK, September 2004. Springer-Verlag. Lecture Notes in Computer Science Vol. 3242.

[16] Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In H. P. Schwefel and R. Männer, editors, *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, volume 496 of *Lecture Notes in Computer Science Vol. 496*, pages 193–197, Berlin, Germany, October 1991. Springer-Verlag.

[17] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining Convergence and Diversity in Evolutionary Multi-objective Optimization. *Evolutionary Computation*, 10(3):263–282, Fall 2002.

[18] Nateri K. Madavan. Multiobjective Optimization Using a Pareto Differential Evolution Approach. In *Congress on Evolutionary Computation (CEC'2002)*, volume 2, pages 1145–1150, Piscataway, New Jersey, May 2002. IEEE Service Center.

[19] Efrén Mezura-Montes and Carlos A. Coello Coello. A Simple Evolution Strategy to Solve Constrained Optimization Problems. In Erick Cantú-Paz, James A. Foster, Kalyanmoy Deb, Lawrence David Davis, Rajkumar Roy, Una-May O' Reilly, Hans-Georg Beyer, Russell Standish, Graham Kendall, Stewart Wilson, Mark Harman, Joachim Wegener, Dipankar Dasgupta, Mitch A. Potter, Alan C. Schultz, Kathryn A. Dowsland, Natasha Jonoska, and Julian Miller, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2003)*, pages 640–641, Heidelberg, Germany, July 2003. Chicago, Illinois, Springer Verlag. Lecture Notes in Computer Science Vol. 2723.

[20] K.E. Parsopoulos, D.K. Taoulis, N.G. Pavlidis, V.P. Plagianakos, and M.N. Vrahatis. Vector Evaluated Differential Evolution for Multiobjective Optimization. In *2004 Congress on Evolutionary Computation (CEC'2004)*, volume 1, pages 204–211, Portland, Oregon, USA, June 2004. IEEE Service Center.

[21] Tea Robič and Bodgan Filipič. DEMO: Differential Evolution for Multiobjective Optimization. In Carlos A. Coello Coello, Arturo Hernández Aguirre, and Eckart Zitzler, editors, *Evolutionary Multi-Criterion Optimization. Third International Conference, EMO 2005*, pages 520–533, Guanajuato, México, March

2005. Springer. Lecture Notes in Computer Science Vol. 3410.

[22] J. David Schaffer. Multiple Objective Optimization with Vector Evaluated Genetic Algorithms. In *Genetic Algorithms and their Applications: Proceedings of the First International Conference on Genetic Algorithms*, pages 93–100. Lawrence Erlbaum, 1985.

[23] Marc Schoenauer and Zbigniew Michalewicz. Evolutionary Computation at the Edge of Feasibility. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Proceedings of the Fourth Conference on Parallel Problem Solving from Nature (PPSN IV)*, pages 245–254, Heidelberg, Germany, September 1996. Berlin, Germany, Springer-Verlag.

[24] Jason R. Schott. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[25] Rainer Storn and Kenneth Price. Differential evolution - a simple and efficient adaptative scheme for global optimization over continuous spaces. Technical Report TR-95- 12, International Computer Science, Berkeley, California, March 1995.

[26] Rainer Storn and Kenneth Price. Differential Evolution - A Fast and Efficient Heuristic for Global Optimization over Continuous Spaces. *Journal of Global Optimization*, 11:341–359, 1997.

[27] Hisashi Tamaki, Hajime Kita, and Shigenobu Kobayashi. Multi-Objective Optimization by Genetic Algorithms : A Review. In Toshio Fukuda and Takeshi Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation (ICEC'96)*, pages 517–522, Nagoya, Japan, 1996. IEEE.

[28] 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.

[29] David A. Van Veldhuizen and Gary B. Lamont. Multiobjective Evolutionary Algorithm Research: A History and Analysis. Technical Report TR-98-03, Department of Electrical and Computer Engineering, Graduate School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio, 1998.

[30] David A. Van Veldhuizen and Gary B. Lamont. On Measuring Multiobjective Evolutionary Algorithm Performance. In *2000 Congress on Evolutionary Computation*, volume 1, pages 204–211, Piscataway, New Jersey, July 2000. IEEE Service Center.

[31] Feng Xue, Arthur C. Sanderson, and Robert J. Graves. Pareto-based Multi-Objective Differential Evolution. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 862–869, Canberra, Australia, December 2003. IEEE Press.

[32] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[33] Eckart Zitzler and Lothar Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257–271, November 1999.

Authors' Biographies

Luis Vicente Santana-Quintero received the B.Sc. in computer systems engineering from the Escuela Superior de Cómputo of the Instituto Politécnico Nacional, and an M.Sc. in computer science from CINVESTAV-IPN (both located in Mexico City), in 2002 and 2004, respectively.

He is currently working towards the Ph.D. degree in the Department of Electrical Engineering at the Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional (CINVESTAV-IPN), in México City.

His current research interests include: evolutionary multiobjective optimization, differential evolution and evolutionary algorithms in general.

Carlos A. Coello Coello received the B.Sc. degree in civil engineering from the Universidad Autónoma de Chiapas, México, and the M.Sc. and the PhD degrees in computer science from Tulane University, USA, in 1991, 1993, and 1996, respectively.

He is currently a professor (CINVESTAV-3D Researcher) at the electrical engineering department of CINVESTAV-IPN, in Mexico City, México. Dr. Coello has authored and co-authored over 120 technical papers and several book chapters. He has also co-authored the book *Evolutionary Algorithms for Solving Multi-Objective Problems* (Kluwer Academic Publishers, 2002). Additionally, Dr. Coello has served in the program committees of over 40 international conferences and has been technical reviewer

for over 35 international journals. He is currently associate editor of the *IEEE Transactions on Evolutionary Computation* and member of the editorial boards of the journals *Evolutionary Computation*, *Soft Computing*, *Engineering Optimization* and the *International Journal of Computational Intelligence Research*. He also chairs the *Task Force on Multi-Objective Evolutionary Algorithms* of the IEEE Computational Intelligence Society. He is a senior member of the IEEE, and is a member of the ACM, Sigma Xi, and the Mexican Academy of Sciences.

His major research interests are: evolutionary multi-objective optimization, constraint-handling techniques for evolutionary algorithms, and evolvable hardware.