A Multi-objective Evolutionary Algorithm based on Decomposition for Constrained Multi-objective Optimization

Saúl Zapotecas Martínez
CINVESTAV-IPN,
Departamento de Computación
Av. IPN 2508. San Pedro Zacatenco
México D.F. 07300, MÉXICO,
email: saul.zapotecas@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 spite of the popularity of the Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D), its use in Constrained Multi-objective Optimization Problems (CMOPs) has not been fully explored. In the last few years, there have been a few proposals to extend MOEA/D to the solution of CMOPs. However, most of these proposals have adopted selection mechanisms based on penalty functions. In this paper, we present a novel selection mechanism based on the well-known ε-constraint method. The proposed approach uses information related to the neighborhood adopted in MOEA/D in order to obtain solutions which minimize the objective functions within the allowed feasible region. Our preliminary results indicate that our approach is highly competitive with respect to a state-of-the-art MOEA which solves in an efficient way the constrained test problems adopted in our comparative study.**

I. Introduction

Multi-objective Evolutionary Algorithms (MOEAs) have been successfully applied to the solution of a wide variety of problems in the fields of science and engineering [1], [2]. Although MOEAs were originally designed for solving unconstrained and box-constrained multi-objective optimization problems, in real-world applications, there are several problems which require satisfying equality and/or inequality constraints. Generally, Constrained Multi-objective Optimization Problems (CMOPs) are difficult to solve, because finding feasible solutions may require substantial additional computational resources (i.e., an important number of extra objective function evaluations). One of the major issues for constrained multi-objective optimization is how to deal with the infeasible solutions produced during the search process. A straightforward way to handle infeasible solutions is to completely disregard them and continue the search process only with the feasible individuals generated by our search engine. However, this sort of mechanism is not appropriate in highly constrained search spaces. Additionally, this sort of scheme does not exploit any information contained in the infeasible solutions that are generated during the search, which significantly increases the computational cost of the optimization process.

In recent years, the Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [3] has shown to be an efficient algorithm to deal with complex Multi-objective Optimization Problems (MOPs) [3], [4], [5]. However, the use of MOEA/D for solving CMOPs has been scarcely reported in the specialized literature. Pal et al. [6] employed a constraint handling mechanism based on the superiority of the feasible solutions for the optimal synthesis of linear antennas in a multi-objective context. This approach was initially introduced by Powell and Skolnick [7] for single-objective optimization. Jan and Zhang [8] proposed a version of MOEA/D based on a penalty function, which uses a threshold to control the amount of penalization to be applied to the infeasible solutions. The authors of this approach reported very promising results in several CMOPs with complicated Pareto sets. Konstantinidis and Yang [9] proposed a constraint-handling approach that adopted MOEA/D and an M-tournament selection as its search engine. This approach was used to solve a multi-objective k-connected deployment and power assignment problem. Recently, Asafuddoula et al. [10] proposed an adaptive constraint-handling scheme which was coupled to MOEA/D. In this approach, a gradient local search engine was adopted in order to repair the infeasible solutions produced during the search process. This approach was validated using benchmark CMOPs and a real-world problem.

In this paper, we present a novel selection mechanism based on the well-known ε-constraint method [11], which was initially proposed to deal with constrained single-objective optimization problems. The ε-constraint method has shown to be highly competitive when incorporated to different Evolutionary Algorithms (EAs) [11], [12], [13]. Our proposed approach adopts as its baseline algorithm the well-known MOEA/D with Differential Evolution (DE) [14] (MOEA/D-DE) [5] which is a popular variant of the original MOEA/D [3]. In our proposed approach, the information related to the neighboring solutions is employed in order to obtain solutions which minimize the objective functions within the feasible region. Furthermore, we present a novel ε level comparison procedure in order

to balance between the generation of feasible solutions and speeding up convergence towards the Pareto optimal front. Our preliminary results indicate that our proposed approach is highly competitive when it is compared with respect to a state-of-the-art MOEA which solves in an efficient way the test problems adopted in our comparative study.

The remainder of this paper is organized as follows. In Section II, we provide the basic concepts required for understanding the rest of the paper. Section III describes our proposed approach, including a detailed explanation of the proposed ε level comparison and its use in order to maintain a suitable balance between feasible solutions and convergence towards the Pareto optimal front. Section IV presents the experimental study used for assessing the performance of our proposed algorithm. In Section V, we provide a discussion of our results. Finally, in Section VI, we provide our conclusions and some possible paths for future research.

II. Basic Concepts

A. Constrained Multi-objective Optimization Problems

A nonlinear Constrained Multi-objective Optimization Problem (CMOP) can be stated as (assuming minimization of all the objective functions):

$$\begin{array}{lll} \min & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_k(\mathbf{x}))^T & \\ \text{s.t.} & g_i(\mathbf{x}) \leq 0, & i = 1, \ldots, p \\ & h_j(\mathbf{x}) = 0, & j = 1, \ldots, q \\ & L_l \leq x_l \leq U_l, & l = 1, \ldots, n \end{array} \quad (1)$$

where $\mathbf{x} = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$ is an n dimensional vector of decision variables, $g_i(\mathbf{x}) \leq 0$ and $h_j(\mathbf{x}) = 0$ represent the p inequality constraints and the q equality constraints, respectively. L_l and U_l are the lower and upper bounds of each of the decision variables x_l. The vector $\mathbf{F}$ consists of k objective functions f_j's ($j = 1, \ldots, k$) to be minimized. The set of solutions that satisfy the constraints of problem (1) defines the feasible region $\Omega \subseteq \mathbb{R}^n$.

In order to describe the concept of optimality in which we are interested on, the following definitions are introduced [15]:

Definition 1. Let $\mathbf{x}, \mathbf{y} \in \Omega$, we say that $\mathbf{x}$ *dominates* $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) if and only if, $f_j(\mathbf{x}) \leq f_j(\mathbf{y})$ and $f_j(\mathbf{x}) < f_j(\mathbf{y})$ in at least one f_j for all $j = 1, \ldots, k$.

Definition 2. Let $\mathbf{x}^\star \in \Omega$, we say that $\mathbf{x}^\star$ is a *Pareto optimal* solution, if there is no other solution $\mathbf{y} \in \Omega$ such that $\mathbf{y} \prec \mathbf{x}^\star$.

Definition 3. The *Pareto optimal set* PS is defined by: $PS = \{\mathbf{x} \in \Omega | \mathbf{x} \text{ is Pareto optimal solution}\}$, and the *Pareto optimal front* PF is defined as: $PF = \{\mathbf{F}(\mathbf{x}) | \mathbf{x} \in \mathcal{PS}\}$.

We thus wish to find the best possible *trade-offs* among the objectives, such that no objective can be improved without worsening another one. However, in this case, our solutions must also satisfy the constraints of problem (1). We are also interested in generating a set of solutions that are well-distributed along the Pareto front.

B. Decomposition of a Multi-objective Optimization Problem

It is well-known that a Pareto optimal solution to the problem (1), under certain conditions, could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the objective functions f_i's. Many scalar approaches have been proposed to aggregate the objectives of an MOP. Among them, the Tchebycheff approach is one of most widely used methods reported in the specialized literature. In the following, we describe the Tchebycheff problem which is adopted in this study. Note however, that other scalarization approaches could also be easily coupled to this work, see for example those presented in [16], [15], [17].

Tchebycheff approach: This approach transforms the vector of function values $\mathbf{F}$ into a scalar optimization problem which is of the form:

$$\begin{array}{ll} \min & g^{te}(\mathbf{x}|\mathbf{w}, \mathbf{z}) = \max_{1 \leq j \leq k} \{w_j | f_j(\mathbf{x}) - z_j|\} \\ s.t. & \mathbf{x} \in \Omega \end{array} \quad (2)$$

where Ω is the feasible region, $\mathbf{z} = (z_1, \ldots, z_k)^T$, such that: $z_j = \min\{f_j(\mathbf{x}) | \mathbf{x} \in \Omega\}$ and $\mathbf{w} = (w_1, \ldots, w_k)^T$ is a weight vector, i.e., $\sum_{j=1}^{k} w_j = 1$ and $w_j \geq 0$ for each $j \in \{1, \ldots, k\}$.

For each Pareto optimal point $\mathbf{x}^\star$ there exists a weight vector $\mathbf{w}$ such that $\mathbf{x}^\star$ is the optimum solution of equation (2) and each optimal solution of equation (2) is a Pareto optimal solution of equation (1). An appropriate representation of the Pareto front could be reached by solving different scalarization problems. Such problems can be defined by a set of well-distributed weight vectors, which establish the search direction in the optimization process.

C. Constraint Violation Degree

The overall constraint violation $\phi(\mathbf{x})$ of a solution $\mathbf{x}$ can be given by the maximum of all constraints or the sum of all constraints:

$$\phi(\mathbf{x}) = \max\{\max_i\{0, g_i(\mathbf{x})\}, \max_j\{|h_j(\mathbf{x})|\}\} \quad (3)$$

$$\phi(\mathbf{x}) = \sum_i ||\max\{0, g_i(\mathbf{x})\}||^\alpha + \sum_j ||h_j(\mathbf{x})||^\alpha \quad (4)$$

where α is a positive number. Equality constraints can be transformed into inequality constraints by using [18]:

$$|h_j(\mathbf{x}) - \varepsilon| \leq 0, j = 1, \ldots, q \quad (5)$$

where ε is a small real-value threshold.

For methods that do not require gradient information, it does not matter if equation (5) is non-differentiable. Therefore, a CMOP with equality constraints can be stated as a CMOP having only inequality constraints. Assuming that all constraints of problem (1) are inequality constraints, the constraint violation in equations (3) and (4) can be computed as:

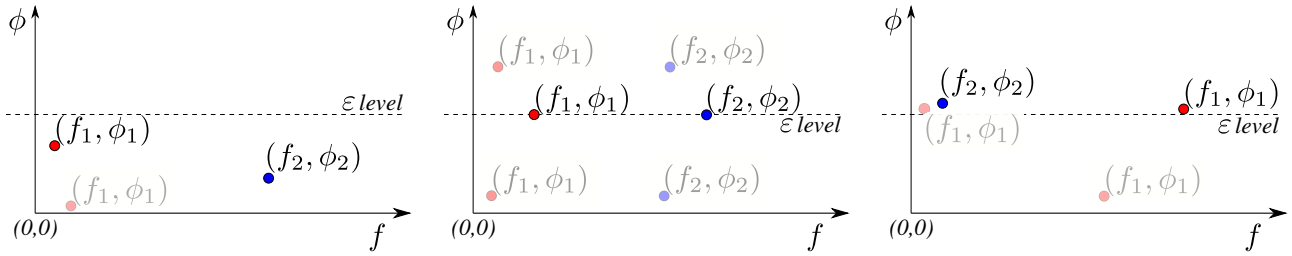

Fig. 1. Illustration of the ε level comparison between (f_1, ϕ_1) and (f_2, ϕ_2). In Figures A, B and C, solutions in red are preferred based on the ε level comparison $(f_1, \phi_1) <_\varepsilon (f_2, \phi_2)$. Figure A shows that when $\phi_1, \phi_2 \leq \varepsilon$, solutions such that $f_1 < f_2$, are preferred even when $\phi_2 < \phi_1$. Figure B shows that when $\phi_1 = \phi_2$, solutions such that $f_1 < f_2$, are preferred even when $\phi_1, \phi_2 > \varepsilon$. Figure C shows that when $\phi_1 < \phi_2$, solutions with a lower ϕ value are preferred even when $f_1 > f_2$ and the difference between ϕ_1 and ϕ_2 is meaningless, being, in some cases, close to the ε level.

$$\phi(\mathbf{x}) = \max_{1 \leq i \leq p} \{0, g_i(\mathbf{x})\} \quad (6)$$

$$\phi(\mathbf{x}) = \sum_{i=1}^{p} ||\max\{0, g_i(\mathbf{x})\}||^\alpha \quad (7)$$

Thus, we can say that, without loss of generality, in this paper we consider only constraint functions of the form $g(\mathbf{x}) \leq 0$. Any equality constraint $h(\mathbf{x}) = 0$ can be transformed into an inequality constraint using equation (5).

D. The ε-Constraint Method

The ε-constraint method for single-objective optimization was proposed by Takahama and Sakai [12]. This constraint-handling technique adopts a lexicographic ordering with relaxation of the constraints. The basic idea is to define an ε level comparison in order to state an order relation on pairs consisting of the objective function value and the constraint violation value of a solution $(f(\mathbf{x}), \phi(\mathbf{x}))$. If the violation of a constraint is greater than 0, then the solution is not feasible and its fitness value will be low. Since the feasibility of a solution $\mathbf{x}$ is more important than the minimization of the objective function $f(\mathbf{x})$, the ε level comparisons are defined as a lexicographic order in which $\phi(\mathbf{x})$ precedes $f(\mathbf{x})$.

Let $f_1(f_2)$ and $\phi_1(\phi_2)$ be the function values and the constraint violation at a solution $\mathbf{x}^1(\mathbf{x}^2)$, respectively. Then, for any ε satisfying $\varepsilon \geq 0$, the ε level comparison $<_\varepsilon$ and $\leq_\varepsilon$ between (f_1, ϕ_1) and (f_2, ϕ_2) is defined as follows:

$$(f_1, \phi_1) <_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 < f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 < f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \quad (8)$$

$$(f_1, \phi_1) \leq_\varepsilon (f_2, \phi_2) \Leftrightarrow \begin{cases} f_1 \leq f_2, & \text{if } \phi_1, \phi_2 \leq \varepsilon \\ f_1 \leq f_2, & \text{if } \phi_1 = \phi_2 \\ \phi_1 < \phi_2, & \text{otherwise} \end{cases} \quad (9)$$

In case of $\varepsilon = \infty$, the ε level comparison $<_\infty$ and $\leq_\infty$ are equivalent to the ordinal comparison $<$ and $\leq$ between function values. Furthermore, the cases $\varepsilon = 0$, $<_0$ and $\leq_0$ are equivalent to the lexicographic order in which the constraint violation $\phi(\mathbf{x})$ precedes the function value $f(\mathbf{x})$. Fig. 1 illustrates the behavior of the ε level comparison in different scenarios.

E. The Framework of MOEA/D-DE

The Multi-Objective Evolutionary Algorithm Based on Decomposition with Differential Evolution (DE) [14] operators (MOEA/D-DE) [5], transforms a MOP into several scalarization problems. Therefore, an approximation of the Pareto front is obtained by solving the N scalarization subproblems in which a MOP is decomposed.

Considering $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ as the well-distributed set of weight vectors, MOEA/D-DE finds the best solution to each subproblem defined by each weight vector using the Tchebycheff approach. The objective function of the i^{th} subproblem is then defined by $g(\mathbf{x}|\mathbf{w}^i, \mathbf{z})$, where $\mathbf{w}^i \in W$ and $\mathbf{z} = (z_1, \ldots, z_k)^T$ is the artificial utopian vector whose component z_j is the minimum value found so far for the objective f_j, for each $j \in \{1, \ldots, k\}$. In MOEA/D-DE, a neighborhood of the weight vector $\mathbf{w}^i$ is defined as a set of its closest weight vectors in W (in terms of the Euclidean distance). The indexes set of its neighboring weight vectors from $\mathbf{w}^i$ is denoted by $B(\mathbf{w}^i)$.

At each generation, MOEA/D-DE finds the best solution to each subproblem throughout the evolutionary process and maintains:

1) a population of N points $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, where $\mathbf{x}^i$ is the current solution to the i^{th} subproblem;
2) $F^1, \ldots, F^N$, where F^i is the F-value of $\mathbf{x}^i$, i.e., $F^i = \mathbf{F}(\mathbf{x}^i)$ for each $i = 1, \ldots, N$;

Algorithm 1 presents the general framework of MOEA/D-DE. However, for a more detailed description the interested reader is referred to [5].

In line 10 of Algorithm 1, each component y_j in $\mathbf{y} = (y_1, \ldots, y_n)^T$ is generated by using the DE operator, that is:

$$y_j = \begin{cases} x_j^{r1} + F \times (x_j^{r2} - x_j^{r3}), & \text{if } rand() < CR \\ x_j^{r1}, & \text{otherwise} \end{cases} \quad (10)$$

for each $j = 1, \ldots, n$.

After the DE operator is applied, Polynomial-based Mutation (PBM) [19] is performed on $\mathbf{y}$. $rand()$ in line 8 of

Algorithm 1 and equation (10) denotes a random number having uniform distribution in the range $(0, 1]$.

Algorithm 1: General Framework of MOEA/D-DE

Input:
a stopping criterion;
N: the number of the subproblems considered in MOEA/D-DE;
W: a set of weight vectors $\{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$;
T: the number of weight vectors in the neighborhood of each weight vector;
δ: the probability that parent solutions are selected from the neighborhood $B(\mathbf{w}^i)$;
n_r: a maximum number of replacements in the neighborhood;
Output:
P: the final population found by MOEA/D-DE.

1 Initialize a random population $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$;
2 $F^i = \mathbf{F}(\mathbf{x}^i)$;
3 $B(\mathbf{w}^i) = \{i_1, \ldots, i_T\}$ such that: $\mathbf{w}^{i_1}, \ldots, \mathbf{w}^{i_T}$ are the T closest weight vectors to $\mathbf{w}^i$, for each $i = 1, \ldots, N$;
4 $\mathbf{z} = (+\infty, \ldots, +\infty)^T$;
5 **while** *stopping criterion is not satisfied* **do**
6 **foreach** $i \in \{1, \ldots, N\}$ **do**
7 $B_s^i = \begin{cases} B(\mathbf{w}^i), & \text{if } rand() < \delta \\ \{1, \ldots, N\}, & otherwise \end{cases}$;
8 Set $r1 = i$ and randomly select two indexes $r2, r3$ from B_s such that: $r1 \neq r2 \neq r3$;
9 Generate a trial solution $\mathbf{y}$ from $\mathbf{x}^{r1}, \mathbf{x}^{r2}$ and $\mathbf{x}^{r3}$ by using Differential Evolution (DE) operator, and perform mutation operator on $\mathbf{y}$;
10 Calculate $\mathbf{F}(\mathbf{y})$;
11 $z_j = \begin{cases} f_j(\mathbf{y}), & \text{if } f_j(y) < z_j \\ z_j, & \text{otherwise} \end{cases}$, $j \in \{1, \ldots, k\}$;
12 $c = 0$;
13 **foreach** $l \in B_s^i$ **do**
14 **if** $g^{te}(\mathbf{y}|\mathbf{w}^l, \mathbf{z}) < g^{te}(\mathbf{x}^l|\mathbf{w}^l, \mathbf{z})$ *and* $c < n_r$ **then**
15 $\mathbf{x}^l = \mathbf{y}$;
16 $F^l = \mathbf{F}(\mathbf{y})$;
17 $c = c + 1$;
18 **end**
19 **end**
20 **end**
21 **end**

III. Our Proposed Approach

As indicated before, the proposed ε Multi-objective Evolutionary Algorithm based on Decomposition (eMOEA/D) for constrained multi-objective optimization is based on the ε-constraint method [12]. In order to deal with complicated Pareto sets, we adopt as baseline algorithm, the well-known MOEA/D-DE proposed by Li and Zhang [5]. However, the selection mechanism that we propose here, can also be used in other decomposition-based MOEAs, such as those proposed in [20], [4], [21]. Our proposed eMOEA/D-DE employs information related to the neighborhood of different subproblems used by MOEA/D-DE.

In the following description we assume that $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ is a set of well-distributed weight vectors. Along the search, eMOEA/D-DE finds the best solution to each subproblem defined by the Tchebycheff approach using each weight vector in W. In this way, the objective function of the i^{th} subproblem is then defined by $g(\mathbf{x}|\mathbf{w}^i, \mathbf{z})$, where $\mathbf{w}^i \in W$ and $\mathbf{z} = (z_1, \ldots, z_k)^T$ is the artificial utopian vector whose component z_j is the minimum value found so far for the objective f_j, for each $j \in \{1, \ldots, k\}$. eMOEA/D-DE uses a neighborhood of the weight vector $\mathbf{w}^i$, which is defined as a set of its closest weight vectors in W (in terms of the Euclidean distance). The indexes set of its neighboring weight vectors from $\mathbf{w}^i$ is denoted by $B(\mathbf{w}^i)$.

At each generation, eMOEA/D-DE finds the best solution to each subproblem according to the decomposition approach and the proposed ε level comparison. Throughout the search, eMOEA/D-DE maintains:

1) a population of N points $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$, where $\mathbf{x}^i$ is the current solution to the i^{th} subproblem;
2) $F^1, \ldots, F^N$, where F^i is the F-value of $\mathbf{x}^i$, i.e., $F^i = \mathbf{F}(\mathbf{x}^i)$ for each $i = 1, \ldots, N$;

In order to understand the way in which our proposed approach works, Algorithm 2 presents the general framework of eMOEA/D-DE, and its internal procedures are explained in more detail in the next subsections.

In Algorithm 2, the trial solution is generated by using the DE operator as in equation (10); we also adopted Polynomial-based Mutation (PBM) [19] in our proposed approach.

A. Our proposed constraint-handling scheme

1) Normalized Constraint Violation Degree: In order to measure an adequate constraint violation between two solutions, we use the normalized constraint violation degree. In the following description, we assume that the constraint violation degree of a solution $\mathbf{x}$ is computed using equation (7).

Let B_s^i be the indexes set for the i^{th} subproblem to be minimized by eMOEA/D-DE (see line 7 of Algorithm 2). Assuming p inequality constraints of the form $g_j \leq 0$ $(j = 1, \ldots, p)$, the normalized constraint violation of any solution $\mathbf{x}$ can be computed as:

$$\phi_N(\mathbf{x}) \quad = \quad \sum_{j=1}^{p} \left|\left| \max\left\{0, \frac{g_j(\mathbf{x})}{g^j_{\max}}\right\} \right|\right|^{\alpha} \qquad (11)$$

where α is a positive number (in this work, $\alpha = 1$) and g^j_{max} is the maximum violation value for the j^{th} constraint in the set of solutions associated with the indexes in B_s^i. To be more precise, g^j_{max} can be calculated by:

$$g^j_{\max} = \max_{l \in B_s^i}\{0, g_j(\mathbf{x}^l)\}, \quad j = 1, \ldots, p \qquad (12)$$

Note however that if $g^j_{max} = 0$, all the solutions $\mathbf{x}^l$'s (for all $l \in B_s^i$) are feasible for the j^{th} constraint and equation (11) cannot be computed. For this case, we set $g^j_{max} = 1$, thus penalizing the normalized constraint value $\phi_N(\mathbf{x})$ if solution $\mathbf{x}$ is not feasible for the j^{th} constraint.

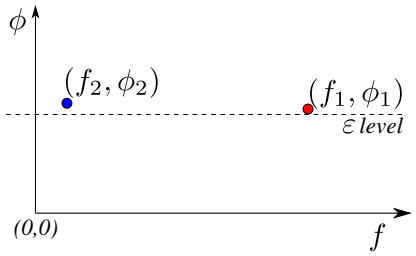

Fig. 2. Illustration of the "*ε level comparison drift*". The values ϕ_1 and ϕ_2 do not satisfy the first two rules of the ε level comparison (see equations (8) and (9)). Therefore, according to the third rule, a solution having a lower value for ϕ is preferred. In this case, solution $\mathbf{x}^1$ is preferred sacrificing convergence to a minimum value of the objective function f, even when solution $\mathbf{x}^2$ is close to obtaining a ϕ_2 value lower than both ϕ_1 and ε.

2) Defining the ε value: The ε value is defined by:

$$\varepsilon = \phi_{N,min} + \tau \times (\phi_{N,max} - \phi_{N,min}) \tag{13}$$

where $\phi_{N,min}$ and $\phi_{N,max}$ are the minimum and the maximum value given by the normalized constraint violation of $\mathbf{x}^l$, for each $l \in B_s^i$. To be more precise:

$$\begin{array}{rcl} \phi_{N,max} & = & \max_{l \in B_s^i}\{\phi_N(\mathbf{x}^l)\} \\ \phi_{N,min} & = & \min_{l \in B_s^i}\{\phi_N(\mathbf{x}^l)\} \end{array} \tag{14}$$

τ represents a parameter given by the user, which tries to balance the generation of feasible solutions with the minimization of each subproblem defined by eMOEA/D-DE.

The value of τ should be given in the range $[0, 1]$. If $\tau = 0$, the search is driven towards the generation of solutions with a normalized constraint value lower than $\phi_{N,min}$. If $\tau = 1$, the search is driven towards the generation of solutions with a lower cost for each subproblem, instead of preferring to satisfy the constraints of the problem.

3) Our proposed ε-constraint method for MOEA/D: Let $\mathbf{x}^1$ and $\mathbf{x}^2$ be two pairs of solutions with their corresponding $f_1(f_2)$ and $\phi_1(\phi_2)$ values. According to the original ε-constraint method [12] (see Section II-D), if $\phi_1, \phi_2 \leq \varepsilon$ the solution that minimizes function f is preferred regardless of its corresponding constraint violation value ϕ. If $\phi_1 = \phi_2$, then the solution that minimizes function f is also preferred. For the third rule (see equations (8) and (9)), solutions with lower constraint violation value ϕ are preferred regardless of their corresponding function values f, i.e., if $\phi_1 < \phi_2$, solution $\mathbf{x}^1$ is preferred even when $f_2 < f_1$, see Fig. 1 C). In this case, it can be that the two solutions have similar ϕ_1 and ϕ_2 values, and they could satisfy $\phi_2 \leq \phi_1$ in a further stage of the search. However, according to the third rule, solution $\mathbf{x}^2$ should be ignored, sacrificing convergence to the solution corresponding to the minimum for objective function f. We call this problem "*the ε level comparison drift*" and we illustrate it in Fig. 2. In order to deal with this drawback of the ε-constraint method, we propose here the introduction of a new rule to be coupled to the ε-constraint method.

Let $\mathbf{y}$ and $\mathbf{x}^l$ be the new solution generated by MOEA/D-DE and the candidate solution to be replaced in eMOEA/D-DE (see Algorithm 2), respectively.

Let $g_1^{te}(g_2^{te})$ and $\phi_N^1(\phi_N^2)$ be the function values given by the Tchebycheff aggregation function (equation (2)) and the normalized constraint violation at the solution $\mathbf{y}(\mathbf{x})$, respectively. Then, for any ε satisfying $\varepsilon \geq 0$, the modified ε level comparison $\leq_\varepsilon$ between (g_1^{te}, ϕ_N^1) and (g_2^{te}, ϕ_N^2) is defined as follows:

$$(g_1^{te}, \phi_N^1) \leq_\varepsilon (g_2^{te}, \phi_N^2) \Leftrightarrow \begin{cases} g_1^{te} \leq g_2^{te}, & \text{if } \phi_N^1, \phi_N^2 \leq \varepsilon \\ g_1^{te} \leq g_2^{te}, & \text{if } \phi_N^1 = \phi_N^2 \\ g_1^{te} \leq g_2^{te}, & \text{if } \delta_e < R_e \\ \phi_N^1 < \phi_N^2, & \text{otherwise} \end{cases} \tag{15}$$

where R_e represents the ratio of solutions $\mathbf{x}^l$'s ($l \in B_s^i$) for which, its corresponding $\phi_N(\mathbf{x}^l)$ is less than the ε value.

With this, we provide a mechanism for dealing with the ε constraint comparison drift. In other words, since the ratio of solutions $\mathbf{x}^l$'s ($l \in B_s^i$) for which, its corresponding $\phi_N(\mathbf{x}^l)$ is less than the ε, and it is greater than δ_e (i.e. $\delta_e < R_e$), we prefer to sacrifice feasibility instead of convergence. Therefore, the δ_e value needs to be properly adjusted.

This set of rules gives rise to the selection mechanism used in the proposed eMOEA/D-DE, i.e., a solution $\mathbf{x}^l$ ($l \in B_s^i$) is replaced by the new solution $\mathbf{y}$, if and only if,

$$(g^{te}(\mathbf{y}|\mathbf{w}^l, \mathbf{z}), \phi_N(\mathbf{y})) \leq_\varepsilon (g^{te}(\mathbf{x}^l|\mathbf{w}^l, \mathbf{z}), \phi_N(\mathbf{x}^l)) \tag{16}$$

which is computed by equation (15).

IV. Experimental Results

A. Test Problems

In order to assess the performance of our proposed eMOEA/D-DE, we compare its results with respect to those obtained by the proposal presented by Jan and Zhang in [8], which we call cMOEA/D-DE. In our comparative study, we adopt the ten CMOPs with complicated Pareto optimal sets proposed by Zhang et al. [22], which constitute the well-known CEC'2009 constrained test suite. In this work, we will denote these problems as CF1 to CF10, respectively. Due to space limitations, the description of these problems is omitted. However, the interested reader is referred to [22] for details of such problems. We used 10 decision variables for all test problems as suggested in [22]. In order to assess the performance of our proposed eMOEA/D-DE on the test problems adopted, we compared it with respect to cMOEA/D-DE using the following performance measures.

Hypervolume: The Hypervolume (I_H) performance measure was proposed in [23]. This performance measure is Pareto compliant [24], and quantifies both convergence and spread of nondominated solutions along the Pareto optimal front. The hypervolume corresponds to the non-overlapped volume of all the hypercubes formed by a reference point $\mathbf{r}$ (given by the user) and each solution $\mathbf{p}$ in the Pareto set approximation (P). It is mathematically stated as:

$$I_H(P) = \Lambda\left(\bigcup_{\mathbf{p} \in P} \{\mathbf{x} | \mathbf{p} \prec \mathbf{x} \prec \mathbf{r}\}\right) \tag{17}$$

Algorithm 2: General Framework of eMOEA/D-DE

Input:
a stopping criterion;
N: the number of the subproblems considered in eMOEA/D-DE;
W: a set of weight vectors $\{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$;
T: the number of weight vectors in the neighborhood of each weight vector;
δ: the probability that parent solutions are selected from the neighborhood $B(\mathbf{w}^i)$;
δ_e: the threshold used for the proposed ε level comparison;
n_r: a maximum number of replacements in the neighborhood;
Output:
P: the final population found by MOEA/D-DE.

Initialize a random population $P = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$;
$F^i = \mathbf{F}(\mathbf{x}^i)$;
$B(\mathbf{w}^i) = \{i_1, \ldots, i_T\}$ such that: $\mathbf{w}^{i_1}, \ldots, \mathbf{w}^{i_T}$ are the T closest weight vectors to $\mathbf{w}^i$, for each $i = 1, \ldots, N$;
$\mathbf{z} = (+\infty, \ldots, +\infty)^T$;
while *stopping criterion is not satisfied* **do**
  foreach $i \in \{1, \ldots, N\}$ **do**
    $B_s^i = \begin{cases} B(\mathbf{w}^i), & \text{if } rand() < \delta \\ \{1, \ldots, N\}, & otherwise \end{cases}$;
    Set $r1 = i$ and randomly select two indexes $r2, r3$ from B_s such that: $r1 \neq r2 \neq r3$;
    Generate a trial solution $\mathbf{y}$ from $\mathbf{x}^{r1}, \mathbf{x}^{r2}$ and $\mathbf{x}^{r3}$ by using the DE operator, and perform mutation operator on $\mathbf{y}$;
    Calculate $\mathbf{F}(\mathbf{y})$;
    $z_j = \begin{cases} f_j(\mathbf{y}), & \text{if } f_j(y) < z_j \\ z_j, & \text{otherwise} \end{cases}$, $j \in \{1, \ldots, k\}$;
    $c = 0$;
    `// for the following description see section III-A`
    $\phi_N^l = \phi_N(\mathbf{x}^l)$, for each $l \in B_s^i$;
    $\phi_{N,max} = \max_{l \in B_s^i}\{\phi_N(\mathbf{x}^l)\}$;
    $\phi_{N,min} = \min_{l \in B_s^i}\{\phi_N(\mathbf{x}^l)\}$;
    $\varepsilon = \phi_{N,min} + \tau \times (\phi_{N,max} - \phi_{N,min})$;
    foreach $l \in B_s^i$ **do**
      if $(g^{te}(\mathbf{y}|\mathbf{w}^l, \mathbf{z}), \phi_N(\mathbf{y})) \leq_\varepsilon (g^{te}(\mathbf{x}^l|\mathbf{w}^l), \mathbf{z}), \phi_N(\mathbf{x}^l))$ *and* $c < n_r$ **then**
        $\mathbf{x}^l = \mathbf{y}$;
        $F^l = \mathbf{F}(\mathbf{y})$;
        $c = c + 1$;
      end
    end
  end
end

where Λ denotes the Lebesgue measure and $\mathbf{r} \in \mathbb{R}^k$ denotes a reference vector being dominated by all valid candidate solutions in P. A high I_H value, indicates that the approximation P is close to PF and has a good spread towards the extreme portions of the Pareto front.

Feasibility Ratio: The feasibility ratio (I_F) indicator refers to the ratio of the number of feasible solutions found in the final approximation P to the Pareto front. It is mathematically stated as:

$$I_F(P) = \frac{P_f}{|P|} \tag{18}$$

where P_f denotes the number of feasible solutions in P and $|P|$ represents the cardinality of the population P.

B. Parameters Settings

As we said before, we compared the results obtained by our proposed eMOEA/D-DE with respect to those obtained by cMOEA/D. In order to allow a fair comparison, the set of weight vectors was the same for both algorithms, and they were generated in the same way, as described in [3], i.e., the settings of N and $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ were controlled by a parameter H. More precisely, $\mathbf{w}^1, \ldots, \mathbf{w}^N$ are all the weight vectors in which each individual weight w_j^i ($i = 1, \ldots, N$ and $j = 1, \ldots, k$) takes a value from:

$$\left\{\frac{0}{H}, \frac{1}{H}, \ldots, \frac{H}{H}\right\}$$

Therefore, the number of such vectors in W is given by $N = C_{H+k-1}^{k-1}$, where k is the number of objective functions. Here, we use $H = 99$ (for two-objective problems) and $H = 23$ (for three-objective problems), i.e., we generated 100 and 300 weight vectors for CMOPs having two and three objectives, respectively,

For each MOP, 30 independent runs were performed with each algorithm. The parameters for both algorithms are summarized in Table I, where N represents the number of initial solutions (100 for bi-objective problems and 300 for three-objective problems). N_{it} represents the maximum number of iterations, which was set to 500 for all test problems. Therefore, both algorithms performed 50,000 (for the bi-objective problems) and 150,000 (for the three-objective problems) fitness function evaluations for each problem. The parameters T_n, δ, F and CR represent the neighborhood size, the probability that parent solutions are selected from the neighborhood $B(\mathbf{w}^i)$, the differential factor and the crossover ratio for the DE operator, respectively. η_m, P_m and n_r are the mutation index (for Polynomial-based Mutation (PBM)), mutation rate and the number of solutions to be replaced in both algorithms, respectively. These values are set as in [8] to allow a fair comparison between the two algorithms. For eMOEA/D-DE, τ and δ_e represent the control parameters for equations (13) and (15), respectively. s_1 and s_2 are the control parameters used by cMOEA/D-DE, and they were set as proposed by Jan and Zhang in [8] (i.e., we adopted the best parameters settings found by Jan and Zhang for cMOEA/D-DE for the test problems of our comparative study).

For each CMOP, the algorithms were evaluated using the Hypervolume (I_H) indicator and the Feasible Ratio indicator (I_F). The results obtained are summarized in Table II. These tables display both the $average$ and the standard deviation (σ) for the two performance measures adopted (i.e., I_H and I_F) for each CMOP considered. The reference vectors used for computing the I_H performance measure are shown in Table II. These vectors are established close to the individual minima for each MOP, i.e., close to the extremes of the Pareto optimal front. With that, a good measure of approximation and spread is reported when the algorithms converge along the Pareto

TABLE I
PARAMETERS FOR EMOEA/D-DE AND CMOEA/D-DE

Parameter	eMOEA/D-DE	MOEA/D-DE
N	100/300	100/300
N_{it}	500	500
T_n	$\lceil 0.1 \times N \rceil$	$\lceil 0.1 \times N \rceil$
δ	0.9	0.9
F	0.5	0.5
CR	1.0	1.0
η_m	20.0	20.0
P_m	$1/n$	$1/n$
n_r	$\lceil 0.01 \times N \rceil$	$\lceil 0.01 \times N \rceil$
τ	0.3	–
δ_e	0.7	–
s_1	–	0.01
s_2	–	20.0

optimal front. For an easier interpretation, the best results are presented in **boldface** for each test problem adopted.

V. DISCUSSION OF RESULTS

As indicated before, the results obtained by our proposed approach (i.e., eMOEA/D-DE) were compared against those produced by cMOEA/D [8]. According to the results presented in Table II, eMOEA/D-DE had a better performance than cMOEA/D-DE in most of the CMOPs adopted. This table provides a quantitative assessment of the performance of eMOEA/D-DE in terms of the I_H and I_F indicators. That means that the solutions obtained by eMOEA/D-DE achieved a better approximation of the Pareto optimal front than the solutions obtained by cMOEA/D-DE while maintaining an acceptable ratio of feasible solutions.

Note however, that for CF3, CF5 and CF8, the I_H indicator showed that our proposed eMOEA/D-DE did not improve the performance of cMOEA/D-DE. Although for CF3, cMOEA/D-DE the I_H was not significantly better, for CF5 and CF8 the performance of eMOEA/D-DE was indeed poor. It is worth noting that the problems adopted in our comparative study, have peculiar shapes in their PS. In the particular cases of CF5 and CF8, their Pareto optimal sets are more scattered and look stranger in comparison with the other CMOPs adopted (see [22]). This leads us to conjecture, that for this type of problems, the computation of the ε constraint value (which is computed by using information from the neighborhood) is not done in a proper way. In fact, the use of a misguided ε value could mislead the search of optimal solutions. However, in order to deal with this drawback, we could either adjust in a dynamic way the parameters ε and δ_e (used in our ε-constraint method) or increase the number of weight vectors. Nonetheless, this task is, indeed, a promising path for future research.

In terms of the feasibility of solutions, we can see from Table II that our proposed eMOEA/D-DE obtained 100% of feasible solutions in most of the test problems adopted. The exceptions were CF1, CF8 and CF10. In fact, even when for CF10, our proposed approach did not obtain 100% of feasible solutions, the I_H and I_F measures indicate that our proposed eMOEA/D-DE had a better performance than cMOEA/D-DE for this specific problem (CF10). On the other hand, the convergence for problems CF1 and CF8, as assessed by the I_H indicator, was not that poor. Although for CF1, cMOEA/D-DE achieved a better ratio of feasible solutions, eMOEA/D-DE had a better convergence to the Pareto optimal front according to the I_H indicator. However, we argue that our proposed approach is highly competitive with respect to cMOEA/D-DE being, in most cases, a better choice.

VI. CONCLUSIONS AND FUTURE WORK

We have proposed a new approach based on MOEA/D-DE for constrained multi-objective optimization. Our proposed eMOEA/D-DE introduces a new selection mechanism based on the ε-constraint method in order to deal with CMOPs. Although the use of the new constraint-handling technique was adopted here for MOEA/D-DE, this approach can be easily adopted by any other decomposition-based MOEA, such as those reported in [20], [4], [21]. As we could see, our proposed eMOEA/D-DE was found to be highly competitive and, in some cases better, than cMOEA/D-DE. In this study, we analyzed the main drawback of the ε-constraint method, which we called "*ε level comparison drift*". In order to deal with this problem, we proposed a new rule which was initially designed for MOEA/D, but it can be easily extended to any other MOEA.

As part of our future work, we intend to focus on the design of a strategy that allows us to adjust, in a dynamic way, the parameters ε and δ_e employed by our proposed eMOEA/D-DE. We also plan to explore different strategies in order to deal with the ε level comparison drift. Furthermore, in order to deal in an efficient way with more complex CMOPs, the introduction of local search mechanisms to eMOEA/D-DE seems to be a promising path for future research. We believe that the use of an appropriate local search mechanism coupled with a MOEA (such as those presented in [25], [26], [27]) could give rise to a powerful search engine capable of dealing with more complex CMOPs in a more effective way. Finally, we also aim to extend our proposed approach to deal with CMOPs having many objectives (three or more), which is an area that has remained practically unexplored so far, to the authors' best knowledge.

REFERENCES

[1] C. A. Coello Coello and G. B. Lamont, Eds., *Applications of Multi-Objective Evolutionary Algorithms*. Singapore: World Scientific, 2004, iSBN 981-256-106-4.

[2] C. A. Coello Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-Objective Problems*, 2nd ed. New York: Springer, September 2007, iSBN 978-0-387-33254-3.

[3] Q. Zhang and H. Li, "MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, December 2007.

[4] Q. Zhang, W. Liu, and H. Li, "The Performance of a New Version of MOEA/D on CEC09 Unconstrained MOP Test Instances," in *2009 IEEE Congress on Evolutionary Computation (CEC'2009)*. Trondheim, Norway: IEEE Press, May 2009, pp. 203–208.

[5] H. Li and Q. Zhang, "Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 2, pp. 284–302, April 2009.

TABLE II
RESULTS OF I_H FOR EMOEA/D-DE AND CMOEA/D-DE

MOP	eMOEA/D-DE		cMOEA/D-DE		reference vector $\mathbf{r}$
	I_H	I_F	I_H	I_F	
	$average$ (σ)	$average$ (σ)	$average$ (σ)	$average$ (σ)	
CF1	**0.688684** (0.001521)	63.20 (0.112841)	0.667054 (0.002620)	**99.600000** (0.011431)	$(1.1, 1.1)^T$
CF2	**0.808304** (0.007086)	**100.00** (0.000000)	0.808195 (0.008220)	**100.000000** (0.000000)	$(1.1, 1.1)^T$
CF3	0.195502 (0.052938)	**100.00** (0.000000)	**0.201052** (0.049597)	**100.000000** (0.000000)	$(1.1, 1.1)^T$
CF4	**0.542160** (0.026495)	**100.00** (0.000000)	0.533707 (0.035476)	98.500000 (0.032838)	$(1.1, 1.1)^T$
CF5	0.323096 (0.080843)	**100.00** (0.000000)	**0.348432** (0.073378)	99.970000 (0.001795)	$(1.1, 1.1)^T$
CF6	**0.790963** (0.014816)	**100.00** (0.000000)	0.773796 (0.016918)	99.900000 (0.003000)	$(1.1, 1.1)^T$
CF7	**0.588919** (0.101440)	**100.00** (0.000000)	0.524981 (0.071369)	99.700000 (0.011299)	$(1.1, 1.1)^T$
CF8	0.494555 (0.042286)	62.56 (0.089307)	**0.583036** (0.015104)	99.960000 (0.001133)	$(1.1, 1.1, 1.1)^T$
CF9	**0.686537** (0.022241)	**100.00** (0.000000)	0.664365 (0.025965)	99.990000 (0.000598)	$(1.1, 1.1, 1.1)^T$
CF10	**0.343309** (0.089898)	**99.03** (0.025391)	0.259685 (0.170106)	73.330000 (0.442217)	$(1.1, 1.1, 1.1)^T$

[6] S. Pal, B. Qu, S. Das, and P. N. Suganthan, "Linear Antenna Array Synthesis with Constrained Multi-objective Differential Evolution," *Progress In Electromagnetics Research B*, vol. 21, pp. 87–111, 2010.

[7] D. Powell and M. M. Skolnick, "Using genetic algorithms in engineering design optimization with non-linear constraints," in *Proceedings of the 5th International Conference on Genetic Algorithms*, San Mateo, CA, USA, 1993.

[8] M. A. Jan and Q. Zhang, "Moea/d for constrained multiobjective optimization: Some preliminary experimental results," in *2010 UK Workshop on Computational Intelligence (UKCI)*, 2010, pp. 1–6.

[9] A. Konstantinidis and K. Yang, "Multi-objective K-connected Deployment and Power Assignment in WSNs using a problem-specific constrained evolutionary algorithm based on decomposition," *Computer Communications*, vol. 34, pp. 83–98, 2011.

[10] M. Asafuddoula, T. Ray, R. Sarker, and K. Alam, "An Adaptive Constraint Handling Approach Embedded MOEA/D," in *2012 IEEE Congress on Evolutionary Computation (CEC'2012)*, 2012, pp. 1–8.

[11] T. Takahama and S. Sakai, "Constrained Optimization by ε Constrained Particle Swarm Optimizer with ε-level Control," in *Proceedings of the 4th IEEE International Workshop on Soft Computing as Transdisciplinary Science and Technology (WSTST'05)*, vol. 29, 2005, pp. 1019–1029.

[12] T. Takahama, S. Sakai, and N. Iwane, "Constrained Optimization by the ε Constrained Hybrid Algorithm of Particle Swarm Optimization and Genetic Algorithm," in *AI 2005: Advances in Artificial Intelligence*, ser. Lecture Notes in Computer Science, S. Zhang and R. Jarvis, Eds. Springer Berlin Heidelberg, 2005, vol. 3809, pp. 389–400.

[13] T. Takahama and S. Sakai, "Constrained Optimization by ε Constrained Differential Evolution with Dynamic ε-Level Control," in *Advances in Differential Evolution*, ser. Studies in Computational Intelligence, U. Chakraborty, Ed. Springer Berlin Heidelberg, 2008, vol. 143, pp. 139–154.

[14] R. M. Storn and K. V. Price, "Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces," ICSI, Berkeley, CA, Tech. Rep. TR-95-012, March 1995.

[15] K. Miettinen, *Nonlinear Multiobjective Optimization*. Boston, Massachuisetts: Kluwer Academic Publishers, 1999.

[16] M. Ehrgott, *Multicriteria Optimization*, 2nd ed. Springer, Berlin, June 2005.

[17] P. Vincke, *Multicriteria Decision-Aid*. John Wiley & Sons, New York, 1992.

[18] C. A. Coello Coello, "Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: A survey of the state of the art," *Computer methods in applied mechanics and engineering*, vol. 191, pp. 1245–1287, 2002.

[19] K. Deb, S. Agrawal, A. Pratap, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II." *IEEE Transactions Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[20] W. Peng and Q. Zhang, "A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems," in *GrC 2008*, 2008, pp. 534 –537.

[21] S. Zapotecas Martínez and C. A. Coello Coello, "A Multi-objective Particle Swarm Optimizer Based on Decomposition," in *GECCO'2011*. Dublin, Ireland: ACM Press, July 2011, pp. 69–76.

[22] Q. Zhang, A. Zhou, S. Zhao, P. N. Suganthan, W. Liu, and S. Tiwari, "Multiobjective optimization test instances for the cec 2009 special session and competition," University of Essex and Nanyang Technological University, Tech. Rep. CES-487, 2008.

[23] E. Zitzler and L. Thiele, "Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study," in *PPSN V*, A. E. Eiben, Ed. Amsterdam: Springer-Verlag, September 1998, pp. 292–301.

[24] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance Assessment of Multiobjective Optimizers: An Analysis and Review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, April 2003.

[25] P. K. Shukla, "On Gradient Based Local Search Methods in Unconstrained Evolutionary Multi-objective Optimization," in *EMO 2007*, S. Obayashi, K. Deb, C. Poloni, T. Hiroyasu, and T. Murata, Eds. Matshushima, Japan: Springer. Lecture Notes in Computer Science Vol. 4403, March 2007, pp. 96–110.

[26] P. Koch, O. Kramer, G. Rudolph, and N. Beume, "On the hybridization of sms-emoa and local search for continuous multiobjective optimization," in *GECCO'2009*. New York, NY, USA: ACM, 2009, pp. 603–610.

[27] S. Zapotecas Martínez and C. A. Coello Coello, "A Direct Local Search Mechanism for Decomposition-based Multi-Objective Evolutionary Algorithms," in *CEC'2012*. Brisbane, Australia: IEEE Press, June 2012, pp. 3431–3438.