A Direct Local Search Mechanism for Decomposition-based Multi-Objective Evolutionary Algorithms

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 recent years, the development of multi-objective evolutionary algorithms (MOEAs) hybridized with mathematical programming techniques has significantly increased. However, most of these hybrid approaches are gradient-based, and tend to require a high number of extra objective function evaluations to estimate the gradient information required. The use of direct search methods—i.e., methods that do not require gradient information—has been, however, less popular in the specialized literature (although such approaches have been used with single-objective evolutionary algorithms). This paper precisely focuses on the design of a hybrid between the well-known MOEA/D and Nelder and Mead's algorithm. Clearly, the mathematical programming technique adopted here, acts as a local search mechanism, whose goal is to improve the search performed by MOEA/D. Because of its nature, the proposed local search mechanism can be easily coupled to any other decomposition-based MOEA. Our preliminary results indicate that this sort of hybridization is quite promising for dealing with multi-objective optimization problems (MOPs) having high dimensionality (in decision variable space).**

I. Introduction

In spite of the current widespread use of evolutionary algorithms for solving MOPs, their computational cost (measured in terms of fitness function evaluations performed) remains as one of their main limitations when applied to real-world applications [3]. In order to address this issue, a variety of hybrid approaches have been proposed in the last few years, mainly focusing on the use of local search engines coupled to different types of MOEAs. Most of these hybrid approaches rely on local search engines based on gradient information [8], [2], [19], [12]. However, few researchers have attempted to hybridize direct search methods (which do not require gradient information) with a MOEA (see for example [1], [25]). Regarding Nelder and Mead's algorithm (also known as *nonlinear simplex search*) which is referred to in this work, the following hybrid algorithms have been reported in the specialized literature in recent years.

Koduru et al. [10] use the nonlinear simplex search hybridized with a Multi-objective Particle Swarm Optimizer (MOPSO). In this work, the authors adopted clustering techniques to build the simplex. The nonlinear simplex search was used as a local search engine for finding nondominated solutions in the neighborhood defined by the particle to be improved. Zapotecas and Coello [24] presented a hybridization between the well-known Nondominated Sorting Genetic Algorithm II (NSGA-II) and the nonlinear simplex search. In this work, the search was directed by an aggregating function and the simplex was constructed using a low-discrepancy sequence into a reduced domain search. Zhong et al. [29] hybridized the nonlinear simplex search and the Differential Evolution (DE) algorithm. The simplex was constructed selecting random solutions from the current population, which were then sorted according to Pareto dominance. At each iteration of the local search, a movement into the simplex was performed for generating new nondominated solutions.

Zhang et al. [28] proposed the multi-objective evolutionary algorithm based on Decomposition (MOEA/D) [28]. This approach decomposes a MOP into several single-objective optimization problems. In this way, a set of approximate solutions to the Pareto optimal set is generated by minimizing each subproblem instead of using a selection mechanism based on Pareto dominance (as the above mentioned hybrid algorithms). More recently, Zapotecas and Coello [26] proposed the Multi-Objective Nonlinear Simplex Search (MONSS), which adopts a decomposition-based approach similar to the one used by MOEA/D. This approach builds a simplex using solutions that minimize a set of neighboring subproblems, and showed the potential of nonlinear simplex search in low-dimensional MOPs.

In this paper, we propose a new memetic algorithm based on both a decomposition approach (we use MOEA/D as our global search engine) and the nonlinear simplex search method (which acts as our local search engine). As we will see later on, the solutions obtained by our proposed memetic algorithm outperform those obtained by the original MOEA/D in most of the test problems adopted in our experimental study.

The remainder of this paper is organized as follows. In Section II, we provide the basic definitions required for understanding the rest of the paper. Section III describes the proposed memetic algorithm, including a detailed explanation of the local search mechanism that we propose. Section IV presents the experimental study used for assessing the performance of our proposed memetic algorithm. In Section VI, we provide a brief discussion of our results. Finally, in Section VII, we provide our conclusions and some possible paths for future research.

II. Basic Concepts

A. Multi-Objective Optimization

An unconstrained Multi-objective Optimization Problem (MOP), can be stated as follows [1]:

$$\min_{\mathbf{x}\in\Omega} \quad \{\mathbf{F}(\mathbf{x})\} \tag{1}$$

where Ω defines the decision space and $\mathbf{F}$ is defined as the vector of objective functions:

$$\mathbf{F}:\Omega\rightarrow\mathbb{R}^k, \quad \mathbf{F}(\mathbf{x})=(f_1(\mathbf{x}),\ldots,f_k(\mathbf{x}))^T$$

where $f_i:\mathbb{R}^n\rightarrow\mathbb{R}$ is a continuous and unconstrained function.

In multi-objective optimization, it is desirable to produce a set of trade-off solutions representing the best possible compromises among the objectives (i.e., solutions such that no objective can be improved without worsening another). In order to describe the concept of optimality in which we are interested, the following definitions are introduced [13]:

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, $f_i(\mathbf{x})\leq f_i(\mathbf{y})$ and $f_i(\mathbf{x})<f_i(\mathbf{y})$ in at least one f_i for all $i=1,\ldots,k$.

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* $\mathcal{PS}$ is defined by:

$$\mathcal{PS}=\{\mathbf{x}\in\Omega|\mathbf{x}\text{ is a Pareto optimal solution}\}$$

Definition 4. The *Pareto Optimal Front* $\mathcal{PF}$ is defined by:

$$\mathcal{PF}=\{\mathbf{F}(\mathbf{x})|\mathbf{x}\in\mathcal{PS}\}$$

Thus, when solving MOPs, we are interested in finding the best possible *trade-offs* among the objectives, such that no objective can be improved without worsening another. Since the number of Pareto optimal solutions can be very large, we are also interested in obtaining a well-distributed set of solutions, since the size of our approximation (produced by a MOEA) will be normally small.

B. Decomposing Multi-Objective Optimization Problems

In the specialized literature, there are several approaches for transforming a MOP into multiple single-objective optimization subproblems [6], [13]. These approaches use a weighted vector as their search direction. In this way, and under certain assumptions (e.g., the minimum is unique, the weighting coefficients are positive, etc.), a Pareto optimal point is achieved by solving such subproblems. Therefore, an approximation of the Pareto optimal front can be achieved by decomposing a MOP into several single-objective optimization problems. Among these methods, perhaps the two most widely used are the *Tchebycheff* and the *Weighted Sum* approaches. It is worth noting, however, that the approaches based on boundary intersection have certain advantages over those based on either Tchebycheff or the Weighted Sum [4], [28]. In the following, we briefly describe a method based, precisely, on the boundary intersection approach, which is referred to in this work.

1) Penalty Boundary Intersection Approach: The Penalty Boundary Intersection (PBI)[2] approach proposed by Zhang and Li [28], uses a weighted vector $\mathbf{w}$ and a penalty value θ for minimizing both the distance to the utopian vector d_1 and the direction error to the weighted vector d_2 from the solution $\mathbf{F}(\mathbf{x})$. Therefore, the optimization problem can be stated as:

$$\text{minimize:} \quad g(\mathbf{x}|\mathbf{w},\mathbf{z}^\star)=d_1+\theta d_2 \tag{2}$$

where

$$d_1=\frac{||(\mathbf{F}(\mathbf{x})-\mathbf{z}^\star)^T\mathbf{w}||}{||\mathbf{w}||}$$

$$\text{and} \quad d_2=\left|\left|(\mathbf{F}(\mathbf{x})-\mathbf{z}^\star)-d_1\frac{\mathbf{w}}{||\mathbf{w}||}\right|\right|$$

such that $\mathbf{x}\in\Omega$ and $\mathbf{z}^\star=(z_1,\ldots,z_k)^T$, such that: $z_i=\min\{f_i(\mathbf{x})|\mathbf{x}\in\Omega\}$.

In this way, a good representation of the Pareto front can be generated by solving a set of problems defined by a well-distributed set of weighted vectors. That has been the main incentive for the development of current decomposition-based MOEAs, see for example [28], [16], [14], [27].

C. The Framework of MOEA/D

The Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) proposed by Zhang and Li [28], decomposes a MOP into several scalarization subproblems. In the following description, we assume that the PBI approach is employed, but the use of other scalarization function is also possible.

Let $W=\{\mathbf{w}_1,\ldots,\mathbf{w}_N\}$ be a set of evenly spread weighted vectors. The problem of approximation of the $\mathcal{PF}$ of a MOP can be decomposed into N scalarization subproblems by using the PBI approach and the objective function of the j^{th} subproblem will be defined by $g(\mathbf{x}|\mathbf{w}_j,\mathbf{z})$, where $\mathbf{w}_j\in\mathbb{R}^k$ and $\mathbf{z}=(z_1,\ldots,z_k)^T$ is the artificial utopian vector whose component z_i is the best value found so far for objective f_i.

In MOEA/D, a neighborhood of weighted vectors $\mathbf{w}_i$ is defined as a set of its closest weighted vectors in $\{\mathbf{w}_1,\ldots,\mathbf{w}_N\}$. The neighborhood of the i^{th} subproblem consists of all the subproblems with the weighted vectors from the neighborhood of $\mathbf{w}_i$.

At each generation t, MOEA/D with the PBI approach maintains: 1) a population of N points $\mathbf{x}_1,\ldots,\mathbf{x}_N\in\Omega$, where $\mathbf{x}_i$ is the current solution to the i^{th} subproblem; 2) $FV^1,\ldots,FV^N$, where FV^i is the F-value of $\mathbf{x}_i$, i.e., $FV^i=\mathbf{F}(\mathbf{x}_i)$ for each $i=1,\ldots,N$; 3) an external population EP, which is used to store nondominated solutions

[1] Without loss of generality, we assume minimization

[2] based on the well-known Normal Boundary Intersection (NBI) method [4]

found during the search. The general framework of MOEA/D is presented below.

```
Input:
  • a stopping criterion;
  • $N$: the number of the subproblems considered in MOEA/D;
  • a well-distributed set of weighted vectors: $\{\mathbf{w}_1, \dots, \mathbf{w}_N\}$;
  • $T$: the number of the weight vectors in the neighborhood of each weighted vector.
Output:
  • $EP$: the nondominated solutions found during the search;
  • $P$: the final population found by MOEA/D.
Step 1) Initialization
    Step 1.1) set $EP = \emptyset$
    Step 1.2) Compute the Euclidean distances between any two weighted vectors and then work out the $T$ closest weighted vectors to each weighted vector. For each $i = 1, \dots, N$, set $B(i) = \{i_1, \dots, i_T\}$ where $\mathbf{w}_{i_1}, \dots, \mathbf{w}_{i_T}$ are the $T$ closest weighted vectors to $\mathbf{w}_i$.
    Step 1.3) Generate an initial population $P = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ randomly. Set $FV^i = \mathbf{F}(\mathbf{x}_i)$.
    Step 1.4) Initialize $\mathbf{z} = (z_1 \dots, z_k)^T$ by a problem-specific method.
Step 2) Update
For $i = 1, \dots, N$, do
    Step 2.1) Reproduction: Randomly select two indexes $k, l$ from $B(i)$, and then generate a new solution $\mathbf{y}$ from $\mathbf{x}_k$ and $\mathbf{x}_l$ by using genetic operators.
    Step 2.2) Improvement: Apply a problem-specific repair/improvement heuristic on $\mathbf{y}$ to produce $\mathbf{y}'$.
    Step 2.3) Update of z: For each $j = 1, \dots, k$, if $z_j < f_j(\mathbf{x})$, then set $z_j = f_j(\mathbf{y}')$.
    Step 2.4) Update of Neighboring Solutions: For each index $j \in B(i)$, if $g(\mathbf{y}'|\mathbf{w}_j, \mathbf{z}) \leq g(\mathbf{x}|\mathbf{w}_j, \mathbf{z})$, then set $\mathbf{x}_j = \mathbf{y}'$ and $FV^j = \mathbf{F}(\mathbf{y}')$.
    Step 2.5) Update of $EP$: Remove from $EP$ all the vectors dominated by $\mathbf{F}(\mathbf{y}')$. Add $\mathbf{F}(\mathbf{y}')$ to $EP$ if no vectors in $EP$ dominate $\mathbf{F}(\mathbf{y}')$.
Step 3) Stopping Criterion: If the stopping criterion is satisfied, then stop and output $EP$ and $P$. Otherwise, go to Step 2.
```

For a detailed description of the above outlined framework, the interested reader is referred to [28].

D. The Multi-Objective Nonlinear Simplex Search

Nelder and Mead's algorithm (also known as *nonlinear simplex search*) [15], has been an efficient alternative for dealing with multidimensional and non-differentiable functions. This method, based on the original idea of Spendley et al. [20], guides the search towards a minimum value by deforming a geometric shape called *simplex* according to the three basic movements described by Nelder and Mead: *reflection*, *expansion* and *contraction*.

Recently, an extension of the nonlinear simplex search for multi-objective optimization has been proposed. The Multi-Objective Nonlinear Simplex Search (MONSS) introduced by Zapotecas and Coello [26], decomposes a MOP into several single-objective scalarization subproblems. Such subproblems are defined by a well-distributed set of weighted vectors $W = \{\mathbf{w}_1, \dots, \mathbf{w}_N\}$ which was previously defined.

At the beginning, MONSS defines a set of N random solutions $\mathcal{S} = \{\mathbf{x}_1, \dots, \mathbf{x}_N\}$ having a uniform distribution. Each vector $\mathbf{x}_i \in \mathcal{S}$ represents a solution for the i^{th} subproblem defined by the i^{th} weighted vector $\mathbf{w}_i \in W$. In this way, different subproblems are simultaneously solved and the set of solutions $\mathcal{S}$ will represent an approximation to the Pareto optimal set lengthwise of the search process. MONSS identifies different solutions along the Pareto front and the search is directed towards different non-overlapped regions (or partitions) C_i's from the set of weighted vectors W, such that, each C_i defines a neighborhood.

The nonlinear simplex search is focused on minimizing a subproblem defined by a weighted vector $\mathbf{w}_s$ which is randomly chosen from C_i. The n-simplex ($\mathbf{\Delta}$) used in the search, is defined as:

$$\mathbf{\Delta} = \{\mathbf{x}_s, \mathbf{x}_1, \dots, \mathbf{x}_n\} \tag{3}$$

such that: $\mathbf{x}_s \in \mathcal{S}$ is a minimum of $g(\mathbf{x}_s|\mathbf{w}_s, \mathbf{z}^\star)$ for the vector $\mathbf{w}_s \in W$. $\mathbf{x}_j \in \mathcal{S}$ ($j = 1, \dots, n$) represents the n solutions that minimize the subproblems defined by the n closest weighted vectors of $\mathbf{w}_s$, where n represents the number of decision variables of the MOP to be solved. In order to relax the search and to prevent the collapse of the simplex, the search direction, i.e., the vector $\mathbf{w}_s$, is changed for any other direction $\hat{\mathbf{w}}_s \in C_i$. Since any vector $\mathbf{w}_s$ from C_i can be chosen, the possible *action range* (A_r) of the search is defined by the cardinality of C_i, i.e., $A_r = |C_i|$.

In order to complete one iteration of the MONSS, the simplex search needs to iterate $n+1$ times in each defined partition C_i. As MONSS defines $m = \frac{|W|}{n+1}$ partitions, at each iteration, $|W|$ fitness function evaluations are performed. All of the new solutions found by MONSS are stored in a pool called "intensification set" ($\mathcal{I}$). At the end of each iteration, the new approximation $\mathcal{S}$ to the Pareto set is updated by selecting the solutions that minimize each subproblem taken from the union $\mathcal{I} \cup \mathcal{S}$. For a detailed description of MONSS, the interested reader is referred to [26].

III. Our Proposed Approach

A. General Framework

Our proposed multi-objective memetic algorithm, adopts MOEA/D [28] as its baseline algorithm. The local search engine is based on the MONSS framework [26], however, in order to couple it to MOEA/D and to improve the search, some modifications have been introduced. In this way, the memetic algorithm (denoted as MOEA/D+LS), explores the global search space using MOEA/D, while the local search mechanism exploits promising regions given by the same MOEA/D. Next, we present the general framework of MOEA/D+LS and the following section describes in detail the local search mechanism.

```
Input:
  • a stopping criterion;
  • $N$: the number of subproblems considered in MOEA/D+LS;
  • a well-distributed set of weighted vectors: $\{\mathbf{w}_1, \dots, \mathbf{w}_N\}$;
  • $T$: the number of weight vectors in the neighborhood of each weighted vector.
  • $R_{ls}$: the number of solutions which are replaced in the current population by the local search.
```

- A_r: the action range for the local search.

Output:

- P: the final population found by MOEA/D+LS.

Step 1) Initialization

Step 1.2) Compute the Euclidean distances between any two weighted vectors and then work out the T closest weighted vectors to each weighted vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$ where $\mathbf{w}_{i_1}, \ldots, \mathbf{w}_{i_T}$ are the T closest weighted vectors to $\mathbf{w}_i$.
Step 1.3) Generate an initial population $P = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ randomly. Set $FV^i = \mathbf{F}(\mathbf{x}_i)$.
Step 1.4) Initialize $\mathbf{z} = (z_1 \ldots, z_k)^T$ by a problem-specific method.

Step 2) Memetic Algorithm

Step 2.1) MOEA/D iteration: Perform **Step 2 of the MOEA/D** framework for obtaining P (the next population).
Step 2.2) Local Search:
If the percentage of nondominated solutions in P is less than 50% then

Step 2.2.1) Selection Mechanism: Select a solution from P as the initial search solution ($\mathbf{p}_{ini}$) according to Section III-B1.
Step 2.2.2) Build the Simplex: Build the simplex according to Section III-B2.
Step 2.2.3) Search Direction: Select the search direction for the nonlinear simplex search according to Section III-B3.
Step 2.2.4) Deform the Simplex: Perform any movement (reflection, contraction or expansion) for obtaining $\mathbf{p}_{new}$ according to Nelder and Mead's algorithm (see Section III-B4).
Step 2.2.5) Update the population: Update the population P using the new solution $\mathbf{p}_{new}$ according to the rules presented in Section III-B5.
Step 2.2.6) Stopping Criterion: If the stopping criterion is satisfied then stop and go to **Step 3**. Otherwise go to **Step 2.2.1** or **Step 2.2.3** according to the rules detailed in Section III-B6.

Else go to **Step 3**.

Step 3) Stopping Criterion: If the stopping criterion is satisfied, then stop and output P. Otherwise go to **Step 2**.

B. Local Search

MOEA/D+LS exploits the promising neighborhoods of the nondominated solutions found by MOEA/D. In the following description, let P be the set of solutions found by MOEA/D in any generation. We assume that if a solution $\mathbf{p} \in P$ is nondominated, there exists another nondominated solution $\mathbf{q} \in \Omega$ such that $||\mathbf{p} - \mathbf{q}|| < \delta$ for any small $\delta \in \mathbb{R}_+$. In other words, the probability that $\mathbf{q}$ is nondominated with respect to $\mathbf{p}$ in the neighborhood defined by δ is equal to one, which implies that $\mathbf{q}$ is also nondominated.

The local search mechanism presented here takes into account this property to obtain new nondominated solutions departing from nondominated solutions allocated in the current population P. Considering that MOEA/D tries to solve a set of subproblems, if all solutions in P are nondominated, we assume that the minimum value for each subproblem has been achieved and then, the application of the local search procedure could be inefficient. Therefore, a good strategy needs to be adopted.

The degrees of freedom of the local search depend of the process used for building the simplex, which (as we will see later on) adopts solutions from the current population. Considering that at end of the evolutionary process the population converges to a particular region of the search space (the place where the nondominated solutions are contained), the performance of the local search engine should be better when the diversity in the population is higher, i.e., when having a low number of nondominated solutions. Thus, in this work, the local search procedure is applied when the percentage of nondominated solutions in P is less than a certain percentage (we used 50% in this paper). In the following sections, we will detail the local search steps included in the above description of our proposed MOEA/D+LS.

1) Selection Mechanism: Let $P^\star \subseteq P$ be the set of nondominated solutions found by MOEA/D in any generation. Assuming that all the nondominated solutions in $P^\star$ are equally efficient, the solution $\mathbf{p}_{ini}$ which starts the local search is randomly taken from $P^\star$. Solution $\mathbf{p}_{ini}$ represents not only the initial search point, but also the simplex head from which the simplex will be built.

2) Building the Simplex: Let $\mathbf{w}_{ini}$ be the weighted vector that defines the subproblem for which the initial search solution $\mathbf{p}_{ini}$ is minimum. Let $S(\mathbf{w}_{ini})$ be the neighborhood of the n closest weighted vectors to $\mathbf{w}_{ini}$ (where n is the number of decision variables of the MOP)[3]. Then, the simplex defined as:

$$\mathbf{\Delta} = \{\mathbf{p}_{ini}, \mathbf{p}_1, \ldots, \mathbf{p}_n\}$$

is built in two different ways by using a probability P_s, according to the two following strategies:

i. Neighboring solutions: The remaining n solutions $\mathbf{p}_i \in P$ $(i = 1, \ldots, n)$ are chosen, such that, $\mathbf{p}_i$ minimizes each subproblem defined by each weighted vector in $S(\mathbf{w}_{ini})$. This is the same strategy employed for constructing the simplex used in MONSS [26].

ii. Sample solutions: The remaining n solutions $\mathbf{p}_i \in \Omega$ $(i = 1, \ldots, n)$ are generated by using a low-discrepancy sequence. The Hammersley sequence [7] is adopted in this work, to generate a well-distributed sampling of solutions in a determined search space. As in [24], we use a strategy based on the genetic analysis of a sample from the current population for reducing the search space. However, here, we compute the average $(\mathbf{m})$ and standard deviation (σ) of the chromosomes (solutions) that minimize each subproblem defined by the weighted vectors in $S(\mathbf{w}_{ini})$. In this way, the new bounds are defined by:

$$\begin{aligned} \mathbf{L}_{bound} &= \mathbf{m} - \sigma \\ \mathbf{U}_{bound} &= \mathbf{m} + \sigma \end{aligned}$$

where $\mathbf{L}_{bound}$ and $\mathbf{U}_{bound}$ are the vectors which define the lower and upper bounds of the new search space, respectively.

[3] Since the dimensionality of the simplex depends of the number of decision variables of the MOP, the population size of the MOEA needs to be larger than the number of decision variables.

> Once the search space has been reduced, the n remaining solutions are generated by means of the Hammersley sequence using as bounds $\mathbf{L}_{bound}$ and $\mathbf{U}_{bound}$.

In this work, we use $P_s = 0.3$ as the probability that the construction of the simplex using sample solutions is chosen. Otherwise, the construction using neighboring solutions is employed.

3) Defining the Search Direction: Let $B(\mathbf{w}_{ini})$ be the neighborhood of the T closest weighted vectors to $\mathbf{w}_{ini}$, such that $\mathbf{w}_{ini}$ defines the subproblem for which the initial search solution $\mathbf{p}_{ini}$ is minimum. Let $D(\mathbf{w}_{ini})$ be the A_r closest weighted vectors to $\mathbf{w}_{ini}$.

The nonlinear simplex search focuses on minimizing a subproblem defined by the weighted vector $\mathbf{w}_{obj}$, which is defined according to the following rules:

i. The farthest weighted vector in $B(\mathbf{w}_{ini})$ to $\mathbf{w}_{ini}$, if it is the first iteration of the local search,

ii. otherwise, a random weighted vector taken from $D(\mathbf{w}_{ini})$ is employed.

Note that (in *ii*) the search is relaxed defining as our action range the A_r weighted vectors closest to $\mathbf{w}_{ini}$. The idea of relaxing the search is taken from the MONSS framework. However, the neighborhood $D(\mathbf{w}_{ini})$ is used instead of a partition as in MONSS. In this work, we used $A_r = 5$.

4) Deforming the Simplex: At each iteration of the local search, the $n + 1$ vertices of the simplex $\mathbf{\Delta}$ are sorted according to their value for the subproblem that it tries to minimize (the best value is the first element). In this way, a movement into the simplex is performed for generating the new solution $\mathbf{p}_{new}$. The movements are calculated according to the equations provided by Nelder and Mead in [15]. Each movement is controlled by three scalar parameters: reflection (α), expansion (β) and contraction (γ).

The simplex search was conceived for unbounded problems. When dealing with bounded variables, the created solutions can be located outside the allowable bounds after some movements of the simplex search. In order to deal with this, we bias the new solution if any component of $\mathbf{p}_{new}$ lies outside the bounds according to:

$$\mathbf{p}_{new}^{(j)} = \begin{cases} \mathbf{L}_{bound}^{(j)} & \text{, if } \mathbf{p}_{new}^{(j)} < \mathbf{L}_{bound}^{(j)} \\ \mathbf{U}_{bound}^{(j)} & \text{, if } \mathbf{p}_{new}^{(j)} > \mathbf{U}_{bound}^{(j)} \\ \mathbf{p}_{new}^{(j)} & \text{, otherwise.} \end{cases} \tag{4}$$

where $\mathbf{L}_{bound}^{(j)}$ and $\mathbf{U}_{bound}^{(j)}$ are the lower and upper bounds of the j^{th} parameter of $\mathbf{p}_{new}$, respectively.

5) Updating the Population: The information provided by the local search engine is introduced to MOEA/D using a Lamarckian evolution scheme [23]. However, since we are dealing with MOPs, the new solution generated by the local search procedure could be better than more than one solution in the current population. For this, we adopt the following mechanism in which some solutions from the population could be replaced:

Let P be the current population reported by the MOEA. Let $\mathbf{p}_{new}$ be the solution generated by any movement of the simplex search. Let $B(\mathbf{w}_{obj})$ and $W = \{\mathbf{w}_1, \ldots, \mathbf{w}_N\}$ be the neighborhood of the T closest weighted vectors to $\mathbf{w}_{obj}$, and the well-distributed set of all weighted vectors, respectively. We define

$$Q = \begin{cases} B(\mathbf{w}_{obj}) & \text{, if } r < \delta \\ W & \text{otherwise} \end{cases}$$

where r is a random number having uniform distribution. In this work, we use $\delta = 0.9$.

The current population P is updated by replacing at most R_{ls} solutions from P such that, $g(\mathbf{p}_{new}|\mathbf{w}_i, z) < g(\mathbf{x}_i|\mathbf{w}_i, z)$, where $\mathbf{w}_i \in Q$ and $\mathbf{x}_i \in P$, such that $\mathbf{x}_i$ minimizes the subproblem defined by $\mathbf{w}_i$.

Note that the loss of diversity is avoided by replacing a maximum number of solutions from P, instead of all the solutions that minimize the subproblems defined by the complete neighborhood Q, as in MOEA/D. In our study, we set $R_{ls} = 15$ as the maximum number of solution to replace.

6) Stopping Criterion: A maximum number of fitness function evaluations E_{ls} is adopted as our stopping criterion. If the nonlinear simplex search overcomes this maximum number of evaluations, the simplex search is stopped and the evolutionary process of MOEA/D continues by going to **Step 3**. However, the search could be inefficient if the simplex has been deformed so that it has collapsed into a region where there are no local minima. According to Lagarias et al. [11] the simplex search finds a better solution in at most $n + 1$ iterations (at least in convex functions with low dimensionality). Thus, we take into account this observation and adopt a stopping criterion for reconstructing the simplex using another nondominated solution from P as simplex head. Therefore, if the simplex search does not find a minimum value in $n + 1$ iterations, we reset the search by going to **Step 2.2.1**. Otherwise, we perform other movement into the simplex using a new search direction, i.e., by going to **Step 2.2.3**.

IV. Experimental Results

A. Test Problems

In order to assess the performance of our proposed memetic algorithm, we compare its results with respect to those obtained by the original MOEA/D [28]. We adopted twelve test problems whose Pareto fronts have different characteristics including convexity, concavity, disconnections and multi-modality. The two-objective test suite of Zitzler-Deb-Thiele (ZDT) [30] (except for ZDT5, which is a binary problem) is adopted. Regarding three-objective problems, we adopted the seven unconstrained MOPs from the Deb-Thiele-Laumanns-Zitzler (DTLZ) test suite [5].

We used 30 decision variables for ZDT1 to ZTD3, while ZDT4 and ZDT6 were tested using 10 decision variables. DTLZ1 was tested using 7 decision variables. For DTLZ2 to DTLZ6, we employed 12 decision variables, while DTLZ7 was tested using 22 decision variables.

B. *Performance Measures*

The following performance measures are used to assess the performance of our proposed memetic algorithm and the original MOEA/D (i.e., the one without the local search mechanism) on the test problems adopted:

Hypervolume: The Hypervolume ($\mathcal{H}v$) measure was proposed by Zitzler [31]. This performance measure is Pareto compliant [32], and quantifies both approximation and maximum spread of nondominated solutions along the Pareto 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 (PF_k). It is mathematically stated as:

$$\mathcal{H}v(PF_k) = \Lambda\left(\bigcup_{p\in PF_k}\{\mathbf{x}|\mathbf{p}\prec\mathbf{x}\prec\mathbf{r}\}\right) \tag{5}$$

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 PF_k.

Two Set Coverage: The two Set Coverage ($\mathcal{SC}$) was proposed by Zitzler et al. [30], and it compares a set of nondominated solutions A with respect to another set B, using Pareto dominance. This performance measure is defined as:

$$\mathcal{SC}(A,B) = \frac{|\{\mathbf{b}\in B|\exists\mathbf{a}\in A : \mathbf{a}\preceq\mathbf{b}\}|}{|B|} \tag{6}$$

If all points in A dominate or are equal to all points in B, this implies that $\mathcal{SC}(A,B)=1$. Otherwise, if no point of A dominates some point in B, then $\mathcal{SC}(A,B)=0$. When $\mathcal{SC}(A,B)=1$ and $\mathcal{SC}(B,A)=0$ then, we say that A is better than B. Since the Pareto dominance relation is not symmetric, we need to calculate both $\mathcal{SC}(A,B)$ and $\mathcal{SC}(B,A)$.

TABLE I
RESULTS OF $\mathcal{H}v$ FOR MOEA/D+LS AND MOEA/D

MOP	MOEA/D+LS	MOEA/D	reference vector $\mathbf{r}$
	average (σ)	*average* (σ)	
ZDT1	**0.819246** (0.038088)	0.751315 (0.033339)	$(1.1, 1.1)^T$
ZDT2	**0.384962** (0.151212)	0.210410 (0.080132)	$(1.1, 1.1)^T$
ZDT3	**0.995692** (0.158499)	0.990212 (0.089499)	$(1.1, 1.1)^T$
ZDT4	0.169257 (0.212639)	**0.600217** (0.138989)	$(1.1, 1.1)^T$
ZDT6	**0.462559** (0.050484)	0.425904 (0.010630)	$(1.1, 1.1)^T$
DTLZ1	0.316904 (0.001091)	**0.317249** (0.000957)	$(0.7, 0.7, 0.7)^T$
DTLZ2	0.768621 (0.000466)	**0.768696** (0.000644)	$(1.1, 1.1, 1.1)^T$
DTLZ3	0.221197 (0.282045)	**0.383622** (0.245603)	$(1.1, 1.1, 1.1)^T$
DTLZ4	**0.768966** (0.000664)	0.768935 (0.000645)	$(1.1, 1.1, 1.1)^T$
DTLZ5	**0.426307** (0.000167)	0.426115 (0.000675)	$(1.1, 1.1, 1.1)^T$
DTLZ6	**0.426345** (0.000714)	0.000228 (0.001226)	$(1.1, 1.1, 1.1)^T$
DTLZ7	**1.922224** (0.012057)	1.916040 (0.016969)	$(1.1, 1.1, 6.1)^T$

TABLE II
RESULTS OF $\mathcal{SC}$ FOR MOEA/D+LS AND MOEA/D

MOP	$\mathcal{SC}$(MOEA/D+LS, MOEA/D)	$\mathcal{SC}$(MOEA/D, MOEA/D+LS)
	average (σ)	*average* (σ)
ZDT1	**0.893657** (0.122230)	0.004889 (0.011666)
ZDT2	**0.432435** (0.149436)	0.001333 (0.007180)
ZDT3	0.667901 (0.021117)	**0.690476** (0.093046)
ZDT4	0.000000 (0.000000)	**1.000000** (0.000000)
ZDT6	0.170720 (0.028694)	**0.867949** (0.036735)
DTLZ1	**0.155326** (0.165805)	0.126444 (0.093361)
DTLZ2	0.120572 (0.028892)	**0.150281** (0.031948)
DTLZ3	**0.469164** (0.376265)	0.227174 (0.260595)
DTLZ4	**0.178360** (0.033641)	0.077111 (0.019450)
DTLZ5	**0.033682** (0.022515)	0.031905 (0.022034)
DTLZ6	**1.000000** (0.000000)	0.000000 (0.000000)
DTLZ7	**0.122837** (0.021196)	0.108987 (0.016292)

V. PARAMETERS SETTINGS

As indicated before, we compared our proposed approach with respect to MOEA/D (using the PBI approach). For a fair comparison, the set of weighted vectors was the same for both algorithms, and they were generated in the same way as in [28]. For each MOP, 30 independent runs were performed with each algorithm. The parameters for both algorithms are summarized in Table III, 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 100 for all test problems. Therefore, both algorithms performed 10,000 (for the bi-objective problems) and 30,000 (for the three-objective problems) fitness function evaluations for each problem. For MOEA/D+LS, α, β and γ represent the control parameters for the reflection, expansion and contraction movements of the nonlinear simplex search, respectively. The parameters T_n, η_c, η_m, P_c and P_m represent the neighborhood size, crossover index (for Simulated Binary Crossover (SBX)), mutation index (for Polynomial-Based Mutation (PBM)), crossover rate and mutation rate, respectively. A_r, R_{ls} and E_{ls} represent the action range, the number of solutions to be replaced and the maximum number of fitness function evaluations employed by the local search, respectively.

Finally, the parameter θ, represents the penalty value used in the PBI approach for both MOEA/D+LS and MOEA/D.

TABLE III
PARAMETERS FOR MOEA/D+LS AND MOEA/D

Parameter	MOEA/D+LS	MOEA/D
N	100/300	100/300
N_{it}	100	100
T_n	20	20
η_c	20	20
η_m	20	20
P_c	1	1
P_m	$1/n$	$1/n$
α	1	–
β	2	–
γ	1/2	–
A_r	5	–
R_{ls}	15	–
E_{ls}	300	–
θ	5	5

For each MOP, the algorithms were evaluated using the two performance measures previously described (*Hypervolume* and *Two Set Coverage*). The results obtained are summarized in Tables I and II. These tables display both the $average$ and the standard deviation (σ) of each performance measure for each MOP. The reference vectors used for computing the $\mathcal{H}v$ performance measure are shown in Table I. 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 front. In the case of the statistics for the $\mathcal{SC}$ performance measure comparing pairs of algorithms—i.e. $\mathcal{SC}(A, B)$, they were obtained as average values of the comparison of all the independent runs from the first algorithm with respect to all the independent runs from the second algorithm. For an easier interpretation, the best results are presented in **boldface** for each performance measure and test problem adopted.

VI. DISCUSSION OF RESULTS

As indicated before, the results obtained by our proposed memetic algorithm (i.e., MOEA/D+LS) were compared against those produced by the original MOEA/D.

According to the results presented in Tables I and II, MOEA/D+LS had a better performance than MOEA/D in most of the MOPs adopted. These tables provide a quantitative assessment of the performance of MOEA/D+LS in terms of the $\mathcal{H}v$ and $\mathcal{SC}$ indicators. That means that the solutions obtained by MOEA/D+LS achieved a better approximation of the Pareto optimal front than those solutions obtained by MOEA/D when a low number of fitness function evaluations was adopted.

However, for ZDT4, DTLZ1, DTLZ2 and DTLZ3, the $\mathcal{H}v$ indicator showed that the local search did not improve the performance of MOEA/D. In contrast, for DTLZ2, MOEA/D was not significantly better than the memetic algorithm, and for the case of ZDT4, DTLZ1 and DTLZ3, MOEA/D+LS was significantly outperformed by MOEA/D. The poor performance of MOEA/D+LS for these problems (ZDT4, DTLZ1 and DTLZ3) is attributed to their high multi-frontality—for a detailed description of these problems see [30], [5]. The effectiveness of MONSS when dealing with unimodal optimization problems having low dimensionality has been shown in [26]. Here, we have designed a local search mechanism based on the MONSS framework for dealing with MOPs with higher dimensionality (in decision variable space). However, when dealing with multi-frontal MOPs, the convergence of the simplex search considerably slows down and may even fail.

Regarding the $\mathcal{SC}$ performance measure, MOEA/D+LS obtained better results than those produced by MOEA/D in the majority of the test problems adopted. This means that the solutions obtained by MOEA/D+LS dominated a higher portion of the solutions produced by MOEA/D. However, MOEA/D was better for ZDT3, ZDT4, ZDT6 and DTLZ2, although the ratio of solutions dominated by MOEA/D was not significantly high for DTLZ2. Although the SC performance measure benefits MOEA/D in ZDT3 and ZDT6, it is worth noting that our proposed multi-objective memetic algorithm reached better results regarding the $\mathcal{H}v$ performance measure in those problems. $\mathcal{H}v$ not only measures the convergence but also the maximum spread of solutions along the Pareto front, which is the reason why our MOEA/D+LS obtained better results regarding this performance measure. High multi-frontality, however, remains as a limitation of our proposed approach. This can be exemplified in ZDT4, in which our proposed approach was clearly outperformed by the original MOEA/D with respect to the two performance measures adopted in our study.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a hybridization of MOEA/D with a nonlinear simplex search scheme, in which the former acts as the global search engine, and the latter works as a local search engine. The local search mechanism is based on the MONSS framework, which adopts a decomposition approach similar to the one used in MOEA/D. Therefore, its use could be easily coupled within other decomposition-based MOEAs, such as those reported in [16], [14], [27]. Our proposed multi-objective memetic algorithm (called MOEA/D+LS), was found to be competitive with respect to the original MOEA/D over a set of test functions taken from the specialized literature, when performing 10,000 and 30,000 fitness function evaluations, for problems having two and three objectives, respectively. We consider that the strategy employed to hybridize the MONSS framework with MOEA/D was appropriate for dealing with the MOPs adopted here. However, we also confirmed that multi-frontality continues to be the Achilles heel of our proposed approach.

As part of our future work, we intend to focus on designing other mechanism that helps us decide whether the local search engine will be triggered or not. Regarding the performance of the nonlinear simplex search scheme, we plan to experiment with other modifications of this method—see for example [22], [17], [21]. We also plan to explore different strategies for constructing the simplex. We believe that the use of an appropriate simplex and a good hybridization strategy could be a powerful combination for solving complex and computationally expensive MOPs (see for example [28], [9]). Given the nature of the methods used here (they do not require gradient information), the use of this hybrid approach could be an efficient alternative when dealing with some real-world problems for which the gradient information is not available. Finally, we also aim to extend our hybrid approach to constrained MOPs using for example, the Complex method [18] or any variants of the nonlinear simplex search algorithm.

Acknowledgments

The first author acknowledges support from CONACyT through a scholarship to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN. The second author gratefully acknowledges support from CONACyT project no. 103570.

References

[1] S. F. Adra, I. Griffin, and P. J. Fleming. Hybrid Multiobjective Genetic Algorithm with a New Adaptive Local Search Process. In H.-G. B. et al., editor, *2005 Genetic and Evolutionary Computation Conference (GECCO'2005)*, volume 1, pages 1009–1010, New York, USA, June 2005. ACM Press.

[2] P. A. Bosman and E. D. de Jong. Combining Gradient Techniques for Numerical Multi-Objective Evolutionary Optimization. In M. K. et al., editor, *2006 Genetic and Evolutionary Computation Conference (GECCO'2006)*, volume 1, pages 627–634, Seattle, Washington, USA, July 2006. ACM Press. ISBN 1-59593-186-4.

[3] C. A. Coello Coello, G. B. Lamont, and D. A. V. Veldhuizen. *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer, 2nd edition, 2007.

[4] I. Das and J. E. Dennis. Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[5] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler. Scalable Test Problems for Evolutionary Multiobjective Optimization. In A. Abraham, L. Jain, and R. Goldberg, editors, *Evolutionary Multiobjective Optimization. Theoretical Advances and Applications*, pages 105–145. Springer, USA, 2005.

[6] M. Ehrgott. *Multicriteria Optimization*. Springer, Berlin, 2nd edition edition, June 2005.

[7] J. M. Hammersley. Monte-Carlo methods for solving multivariable problems. *Annals of the New York Academy of Science*, 86:844–874, 1960.

[8] X. Hu, Z. Huang, and Z. Wang. Hybridization of the Multi-Objective Evolutionary Algorithms and the Gradient-based Algorithms. In *Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003)*, volume 2, pages 870–877, Canberra, Australia, December 2003. IEEE Press.

[9] S. Huband, P. Hingston, L. Barone, and L. While. A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, 10(5):477–506, October 2006.

[10] P. Koduru, S. Das, and S. M. Welch. Multi-Objective Hybrid PSO Using ϵ-Fuzzy Dominance. In D. Thierens, editor, *2007 Genetic and Evolutionary Computation Conference (GECCO'2007)*, volume 1, pages 853–860, London, UK, July 2007. ACM Press.

[11] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. Convergence properties of the Nelder–Mead simplex method in low dimensions. *SIAM Journal of Optimization*, 9:112–147, 1998.

[12] A. Lara, G. Sanchez, C. A. Coello Coello, and O. Schütze. HCS: A New Local Search Strategy for Memetic Multi-Objective Evolutionary Algorithms. *IEEE Transactions on Evolutionary Computation*, 14(1):112–132, February 2010.

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

[14] N. A. Moubayed, A. Petrovski, and J. A. W. McCall. A novel smart multi-objective particle swarm optimisation using decomposition. In *PPSN (2)*, pages 1–10, 2010.

[15] J. A. Nelder and R. Mead. A Simplex Method for Function Minimization. *The Computer Journal*, 7:308–313, 1965.

[16] W. Peng and Q. Zhang. A decomposition-based multi-objective particle swarm optimization algorithm for continuous optimization problems. In *IEEE International Conference on Granular Computing, 2008. GrC 2008*, pages 534 –537, 2008.

[17] M. K. Rahman. An intelligent moving object optimization algorithm for design problems with mixed variables, mixed constraints and multiple objectives. *Structural and Multidisciplinary Optimization*, 32(1):40–58, July 2006.

[18] A. Ravindran, K. Ragsdell, and G. Reklaitis. *Engineering Optimization. Methods and Applications*. John Wiley & Sons, Inc., Hoboken, New Jersey, USA, 2006.

[19] P. K. Shukla. On gradient based local search methods in unconstrained evolutionary multi-objective optimization. In *EMO*, pages 96–110, 2006.

[20] W. Spendley, G. R. Hext, and F. R. Himsworth. Sequential Application of Simplex Designs in Optimization and Evolutionary Operation. *Technometrics*, 4(4):441–461, November 1962.

[21] V. J. Torczon. *Multidirectional search: A direct search algorithm for parallel machines*. PhD thesis, Department of Mathematical Sciences, Rice University, 1989.

[22] M. B. Trabia and X. B. Lu. A Fuzzy Adaptive Simplex Search Optimization Algorithm. *Journal of Mechanical Design*, 123:216–225, 2001.

[23] L. D. Whitley, V. S. Gordon, and K. E. Mathias. Lamarckian Evolution, The Baldwin Effect and Function Optimization. In *Proceedings of the International Conference on Evolutionary Computation. The Third Conference on Parallel Problem Solving from Nature: Parallel Problem Solving from Nature*, PPSN III, pages 6–15, London, UK, 1994. Springer-Verlag.

[24] S. Zapotecas Martínez and C. A. Coello Coello. A Proposal to Hybridize Multi-Objective Evolutionary Algorithms with Non-Gradient Mathematical Programming Techniques. In *Parallel Problem Solving from Nature–PPSN X*, volume 5199, pages 837–846. Springer. Lecture Notes in Computer Science, September 2008.

[25] S. Zapotecas Martínez and C. A. Coello Coello. A Memetic Algorithm with Non Gradient-Based Local Search Assisted by a Meta-Model. In *Parallel Problem Solving from Nature–PPSN XI*, volume 6238, pages 576–585. Springer. Lecture Notes in Computer Science, September 2010.

[26] S. Zapotecas Martínez and C. A. Coello Coello. MONSS: A Multi-Objective Nonlinear Simplex Search Algorithm. Technical Report TR:2011-09-09, Centro de Investigación y de Estudios Avanzados del IPN (CINVESTAV-IPN), México, D.F., MÉXICO, September 2011.

[27] S. Zapotecas Martínez and C. A. Coello Coello. A Multi-objective Particle Swarm Optimizer Based on Decomposition. In *Proceedings of the 13th Annual Genetic and Evolutionary Computation Conference (GECCO'2011)*, pages 69–76, Dublin, Ireland, July 2011. ACM.

[28] Q. Zhang and H. Li. MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, December 2007.

[29] X. Zhong, W. Fan, J. Lin, and Z. Zhao. Hybrid non-dominated sorting differential evolutionary algorithm with nelder-mead. In *Intelligent Systems (GCIS), 2010 Second WRI Global Congress on*, volume 1, pages 306 –311, December 2010.

[30] E. Zitzler, K. Deb, and L. Thiele. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *Evolutionary Computation*, 8(2):173–195, Summer 2000.

[31] E. Zitzler and L. Thiele. Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In A. E. Eiben, editor, *Parallel Problem Solving from Nature V*, pages 292–301, Amsterdam, September 1998. Springer-Verlag.

[32] 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*, 7(2):117–132, April 2003.