A Nonlinear Simplex Search Approach for Multi-Objective Optimization

Saúl Zapotecas Martínez Alfredo Arias Montaño Carlos A. Coello Coello

***Abstract*—This paper proposes an algorithm for dealing with nonlinear and unconstrained multi-objective optimization problems (MOPs). The proposed algorithm adopts a nonlinear simplex search scheme in order to obtain multiple approximations of the Pareto optimal set. The search is directed by a well-distributed set of weighted vectors. Each weighted vector defines a scalarization problem which is solved by deforming a simplex according to the movements described by Nelder and Mead's method. The simplex is constructed with a set of solutions which minimize different scalarization problems defined by a set of neighbor weighted vectors. The solutions found in the search are used to update a set of solutions considered to be the minima for each separate problem. In this way, the proposed algorithm collectively obtains multiple trade-offs among the different conflicting objectives, while maintaining a well distributed set of solutions along the Pareto front. The main aim of this work is to show that a well-designed strategy using just mathematical programming techniques can be competitive with respect to a state-of-the-art multi-objective evolutionary algorithm.**

I. Introduction

Mathematical programming techniques for solving multi-objective optimization problems have shown to be an effective tool in many domains, at a reasonably low computational cost. However, they have several limitations, including the fact that many of them generate a single nondominated solution per run, and that many others cannot properly handle non-convex, or disconnected Pareto fronts.

On the other hand, multi-objective evolutionary algorithms (MOEAs) have been found to offer several advantages, including generality (they require little domain information to work) and ease of use. However, they are normally computationally expensive (in terms of the number of objective function evaluations required to generate a reasonably good approximation of the Pareto front), which limits their use in some real-world applications.

The characteristics of these two types of approaches naturally motivates the idea of hybridizing them. This idea has been explored by a number of researchers using both gradient-based methods and direct search methods in combination with MOEAs (see for example [8], [10]). However, the development of multi-objective mathematical programming approaches that take ideas from MOEAs and show a similar or better performance than them has been rare (see for example [9]), and is precisely the focus of this paper.

The authors are with CINVESTAV-IPN, Departamento de Computación (Evolutionary Computation Group), Av. IPN No. 2508, Col. San Pedro Zacatenco, México, D.F., 07360, MEXICO (email: saul.zapotecas@gmail.com, aarias@computacion.cs.cinvestav.mx, ccoello@cs.cinvestav.mx). The third author is also affiliated to the UMI LAFMIA 3175 CNRS at CINVESTAV-IPN.

Here, we present a novel multi-objective optimization algorithm based on direct search methods (i.e., those that do not require gradient information). The proposed approach analyzes and exploits the properties of Nelder and Mead's method [16] (which was originally proposed for single-objective optimization) in order to generate multiple solutions along the Pareto front of a problem. The main goal of the proposed strategy is to speed up convergence by means of movements guided by mathematical programming techniques, while maintaining a reasonably good representation of the Pareto front. As we will see later on, our results indicate that our proposed approach is computationally efficient (in terms of the objective function evaluations that it performs) and produces competitive results when dealing with multi-objective optimization problems (MOPs) of low and moderate dimensionality. Our main aim is to raise the interest of people working with MOEAs to hybridize their approaches with methods such as ours, in order to combine the main advantages of these two types of multi-objective optimization algorithms.

The remainder of this paper is organized as follows. In Section II, we provide the basic background required for understanding the rest of the paper. In Section III, we describe in detail our proposed approach. In Section IV, the test problems adopted to validate our approach are described. In Section V, we show the results obtained by our proposed approach. Finally, in Section VI, we provide our conclusions and some possible paths for future research.

II. Basic Concepts

A. Multi-Objective Optimization

A continuous and unconstrained multi-objective optimization problem, can be stated as follows [1]:

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

where Ω define the decision space and F is defined as the vector of the objective functions:

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

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

In multi-objective optimization, we aim 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).

[1] Without loss of generality, we assume minimization

Thus, in order to describe the concept of optimality in which we are interested, the following definitions are introduced.

Definition 1. Let $x, y \in \Omega$, we say that x *dominates* y (denoted by $x \prec y$) if and only if, $f_i(x) \leq f_i(y)$ and $F(x) \neq F(y)$.

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

Definition 3. The *Pareto Optimal Set* $\mathcal{PS}$ is defined by:

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

and its image (i.e., $\mathcal{PF} = \{F(x) | x \in \mathcal{PS}\}$) is called *Pareto Optimal Front*.

We are interested in maximizing the number of elements of the Pareto optimal set and maintaining a well-distributed set of solutions along the Pareto front.

B. Decomposing Multi-objective Optimization Problems

In the specialized literature, there are several approaches for transforming a MOP into multiple single-objective optimization subproblems [15]. 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. Among these methods, probably the two most widely used are the *Tchebycheff* and the *Weighted Sum* approaches. However, as it has been previously discussed in [3], [24], the approaches based on boundary intersection possess certain advantages over those based on either Tchebycheff or the Weighted Sum. In the following, we introduce a decomposition approach based on the boundary intersection, which is the approach adopted in this work.

1) Penalty Boundary Intersection Approach: The Penalty Boundary Intersection (PBI) approach[2] was proposed by Zhang and Li [24], and uses a weighted vector 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 $F(x)$ (see Fig. 1). Mathematically, the PBI problem can be stated as follows:

Let $w = (w_1, \ldots, w_k)^T$ be a weighted vector, i.e., $w_i \geq 0$ for all $i = 1, \ldots, k$ and $\sum_{i=1}^{k} w_i = 1$. Then, the optimization problem is defined as:

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

such that:

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

where $x \in \mathbb{R}^n$, θ is the penalty value and $z^\star = (z_1^\star, \ldots, z_k^\star)^T$ is the utopian vector, i.e., $z^\star = \min\{f_i(x) | x \in \Omega\}$ for each $i = 1, \ldots, k$.

[2]PBI is based on the well-known Normal Boundary Intersection (NBI) method [3]

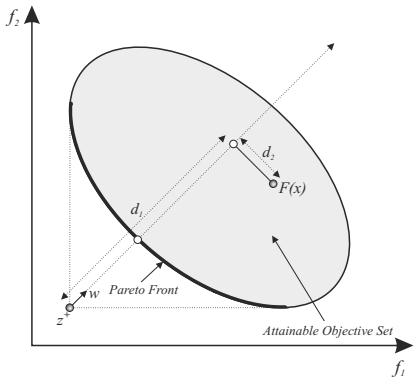


Fig. 1. Illustration of the Penalty Boundary Intersection (PBI) approach.

In this way, the PBI approach can generate a good approximation along the Pareto optimal front by defining a well-distributed set of weighted vectors.

C. The Nonlinear Simplex Search

Nelder and Mead's method [16] also known as the *Nonlinear Simplex Search* (NSS), is an algorithm based on the simplex algorithm of Spendley et al. [20], which was introduced for minimizing nonlinear and multi-dimensional unconstrained functions. While Spendley et al.'s algorithm uses regular simplexes, Nelder and Mead's method generalizes the procedure to change the shape and size of the simplex. Therefore, the convergence towards a minimum value at each iteration of the NSS method is conducted by three main movements in a geometric shape called *simplex*. The following definitions are of relevance here:

Definition 4. A *simplex* or *n-simplex* Δ is a convex hull of a set of $n + 1$ affinely independent points Δ_i $(i = 1, \ldots, n+1)$, in some Euclidean space of dimension n.

Definition 5. A simplex is called *nondegenerated*, if and only if, the vectors in the simplex denote a linearly independent set. Otherwise, the simplex is called *degenerated*, and then, the simplex will be defined in a lower dimension than n.

The full algorithm is defined stating three scalar parameters to control the movements performed in the simplex: **reflection** (α), **expansion** (γ) and **contraction** (β). At each iteration, the $n + 1$ vertices Δ_i of the simplex represent solutions which are evaluated and sorted according to: $f(\Delta_1) \leq f(\Delta_2) \leq \cdots \leq f(\Delta_{n+1})$. In this way, the movements performed in the simplex by the NSS method are defined as:

1) *Reflection*: $x_r = (1 + \alpha)x_c - \alpha\Delta_{n+1}$.
2) *Expansion*: $x_e = (1 + \alpha\gamma)x_c - \alpha\gamma\Delta_{n+1}$.
3) *Contraction*:
   a) *Outside*: $x_{co} = (1 + \alpha\beta)x_c - \alpha\beta\Delta_{n+1}$.
   b) *Inside*: $x_{ci} = (1 - \beta)x_c + \beta\Delta_{n+1}$.

where $x_c = \frac{1}{n}\sum_{i=1}^{n} \Delta_i$ is the centroid of the n best points (all vertices except for Δ_{n+1}), Δ_1 and Δ_{n+1} are the best and the worst solutions identified within the simplex, respectively. Fig. 2 shows all the possible movements made by the NSS method. At each iteration, the initial simplex is modified by one of the above movements, according to the following rules:

1. If $f(\Delta_1) \leq f(x_r) \leq f(\Delta_n)$, then $\Delta_{n+1} = x_r$.
2. If $f(x_e) < f(x_r) < f(\Delta_1)$, then $\Delta_{n+1} = x_e$, otherwise $\Delta_{n+1} = x_r$.
3. If $f(\Delta_n) \leq f(x_r) < f(\Delta_{n+1})$ and $f(x_{co}) \leq f(x_r)$, then $\Delta_{n+1} = x_{co}$.
4. If $f(x_r) \geq f(\Delta_{n+1})$ and $f(x_{ci}) < f(\Delta_{n+1})$, then $\Delta_{n+1} = x_{ci}$.

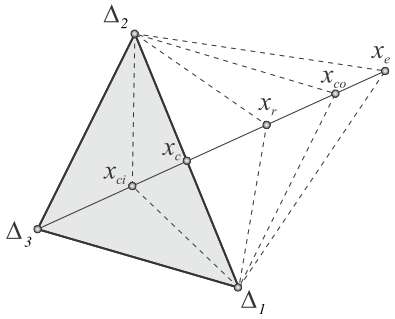


Fig. 2. Illustration of the possible movements in the simplex performed by the NNS method. The constructed simplex corresponds to an optimization problem with two decision variables, where Δ_1 and Δ_3 are the best and the worst point, respectively.

III. The Nonlinear Simplex Search for Multi-objective Optimization

A. About the Nonlinear Simplex Search

As indicated before, mathematical programming techniques are known to have several drawbacks with respect to evolutionary algorithms. The Nelder and Mead method has one more: the convergence towards an optimal value can fail when the simplexes elongate indefinitely and their shape goes to infinity in the space of simplex shapes (as, for example, in McKinnon's functions [14]). For this family of functions and others having similar features, a more appropriate strategy needs to be adopted (e.g., adjusting the control parameters, constructing in a different way the simplex, improving heuristically the movements of the NSS method, etc.). In recent years, several attempts to improve the NNS method have been reported in the literature (see for example [22], [17]). Also, different strategies for the construction of the simplex were explored in [1], [23].

The construction of the simplex plays an important role in the performance of the NSS method. To employ a degenerated simplex (i.e., to use a simplex defined in a lower dimension than the number of decision variables) in the minimization process, is not a good idea. That is because the search is restricted to find an optimal solution in a lower dimension, which avoids achieving this optimal solution if it is not allocated in the same dimensionality as the simplex [11]. On the other hand, a degenerated simplex could be used to obtain local optimal solutions, at least, in the dimensionality defined by the simplex.

In most real-world MOPs, the features of the Pareto optimal set are unknown. If the Pareto optimal set is contained in a lower dimension than the number of decision variables, then, the property that exists when using a degenerated simplex in the search could be exploited. Since the MOP is decomposed into several single-objective subproblems and assuming that each subproblem is solved throughout the search, then, the simplex could be constructed using such solutions. In this way, multiple approximate solutions to the Pareto optimal set are achieved while the search eventually converges to the region in which the Pareto set is contained.

The convergence towards a better point given by the NSS method should be achieved at most in $n+1$ iterations (at least in convex functions with low dimensionality) [11]. Thus, a considerable number of function evaluations will be used to minimize each subproblem. Therefore, a good strategy for approximating solutions to the Pareto optimal set needs to be adopted. In this work, we take into account the above observations and design an effective nonlinear simplex search approach for solving MOPs. This strategy is described next.

B. The Proposed Approach

Our proposed Nonlinear Simplex Search for Multiobjective Optimization (NSS-MO), decomposes a MOP into several single-objective scalarization subproblems. Therefore, a well-distributed set of weighted vectors $W = \{w_1, \ldots, w_N\}$ has to be previously defined. Here, we use the same method as in [24], however, other methods can be used, see for example [2].

At the beginning, a set of N solutions $\mathcal{S} = \{x_1, \ldots, x_N\}$ is randomly initialized. Each solution $x_i \in \mathcal{S}$ minimizes the i^{th} subproblem defined by the i^{th} weighted vector $w_i \in W$. In this way, different subproblems are solved by the NSS-MO algorithm and the *leading set* (i.e. the set $\mathcal{S}$) will approximate solutions towards the Pareto optimal set lengthwise of the search process. 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. That is, let $C = \{C_1, \ldots, C_m\}$ be a set of partitions from W, then, the claim is the following:

$$\bigcap_{i=1}^{m} C_i = \emptyset \text{ and } \bigcup_{i=1}^{m} C_i = W \tag{3}$$

and all the weighted vectors $w_c \in C_i$ are contiguous among themselves.

Thus, the NSS method is focused on minimizing a subproblem defined by a weighted vector w_s which is randomly chosen from C_i. The n-simplex (Δ) used in the search, is defined as:

$$\Delta = \{x_s, x_1, \ldots, x_n\} \tag{4}$$

such that: $x_s \in \mathcal{S}$ is a minimum of $g(x_s|w_s, z^\star)$ for any $w_s \in W$. $x_j \in \mathcal{S}$ represents the n solutions that minimize the

Algorithm 1 $update(W, \mathcal{S}, \mathcal{I})$

1: $\mathcal{T} = \mathcal{S} \cup \mathcal{I}$;
2: $\mathcal{R} = \emptyset$;
3: **for all** $w_i \in W$ **do**
4: $\quad \mathcal{R} = \mathcal{R} \cup \{x^\star | \min_{x^\star \in \mathcal{T}} g(x^\star | w_i, z^\star)\}$;
5: $\quad \mathcal{T} = \mathcal{T} \setminus \{x^\star\}$;
6: **end for**
7: **return** $\mathcal{R}$;

subproblems defined by the nearest n weighted vectors of w_s, where $j = 1, \ldots, n$ and n represents the number of decision variables of the MOP. After a movement made by the NSS method, it is common that the new solution obtained, x_n, leaves the search space. In order to deal with this problem, (as in [23]) we bias deterministically the boundaries. Therefore, the i^{th} bound of the new solution x_n is re-established as follows:

$$x_n^i = \begin{cases} x_{lb}^i, & \text{if } x_n^i < x_{lb}^i \\ x_{ub}^i, & \text{if } x_n^i > x_{ub}^i \end{cases} \quad (5)$$

where x_{lb}^i and x_{ub}^i are, respectively, the lower and upper bounds in the i^{th} component of the search space.

The search could be relaxed at each iteration by changing the direction vector for any other direction $\hat{w}_s \in C_i$. With this, we get an agile search into the partition C_i and we avoid collapsing the simplex search in the same direction w_s.

Here, we define $m = \frac{|W|}{n+1}$ partitions of the set W, guaranteeing at least $n+1$ iterations of the NSS method for each partition, which can be constructed using a naive modification of the well-known k-means algorithm [13]. Then, we say that one iteration of our NSS-MO has been carried out, when the NSS method iterates $n+1$ times in each defined partition C_i. Therefore, at each iteration the proposed algorithm performs $|W|$ function evaluations. All of the new solutions found in the search process are stored in a pool called *intensification set* ($\mathcal{I}$). Then, at the end of each iteration, the leading set $\mathcal{S}$ is updated using both the intensification set $\mathcal{I}$ and the weighted set W, such as it is shown in Algorithm 1. With this, the NSS method minimizes each subproblem, generating new search trajectories among the solutions of the simplex, while the updating mechanism replaces the misguided paths by selecting the best solutions according to the PBI approach, simulating the Path Relinking method [7]. In Fig. 3, we show a possible partition of the weighted set W for a MOP with three objective functions and five decision variables, i.e. defining an n-simplex with six vertices.

Summarizing, the NSS-MO algorithm can be stated as follows:

Step 1) Initialization

Step 1.1) $t = 0$ // the number of iterations
Step 1.2) Generate a well-distributed set of weighted vectors $W = \{w_1, \ldots, w_N\}$ of N.

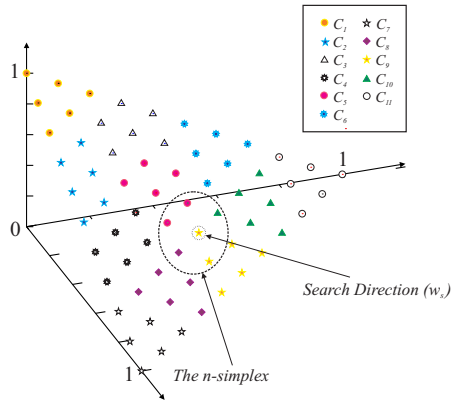


Fig. 3. Illustration of a well-distributed set of weighted vectors for a MOP with three objectives, five decision variables and 66 weighted vectors, i.e. $m = \frac{|W|}{n+1} = 11$ partitions. The n-simplex is constructed by six solutions contained in four different partitions (C_5, C_8, C_9 and C_{10}). The search is focused on the direction defined by the weighted vector w_s.

Step 1.3) Generate the *leading set* $\mathcal{S}^t = \{x_1 \ldots, x_N\}$ of N random solutions.
Step 1.4) Generate partitions: Generate $m = \frac{|w|}{n+1}$ partitions $C = \{C_1, \ldots, C_m\}$ from W (where n is the number of decision variables), such that: the eq.(3) is satisfied.

Step 2) The iteration

// *Generate the intensification set* $\mathcal{I}$
$\mathcal{I} = \emptyset$
For $i = 1, \ldots, m$, do // *for each partition* $C_i \in C$
2.1) Randomly choose $w_s \in C_i$
2.2) Apply Nelder and Mead's method:
2.2.1) Build the n**-simplex:** Construct the n-simplex from $\mathcal{S}^t$, such that: eq.(4) is satisfied.
2.2.2) Apply the NSS method: Execute the NSS method during $n+1$ iterations. At each iteration:
- Repair the bounds according to eq.(5).
- Relax the search changing the direction search w_s for any other $\hat{w}_s \in C_i$.
- Each new solution found by the NSS method is stored in the intensification set $\mathcal{I}$.

Step 3) Update the leading set: Update the *leading set* $\mathcal{S}$ using Algorithm 1. That is: $\mathcal{S}^{t+1} = update(W, \mathcal{S}^t, \mathcal{I})$
Step 4) Stopping Criteria: If $t < N_{it}$ (N_{it} is the maximum number of iterations) then $t = t + 1$ and go to **Step 2**. Otherwise, stop NSS-MO and **output:** $\mathcal{S}^{t+1}$.

IV. Test Problems

In order to assess the performance of the proposed approach, we compare its results with respect to those obtained by a state-of-the-art MOEA called MOEA/D [24]. We adopted three benchmark problems (LIS, FONSECA and DTLZ5) and an airfoil design problem as a case study. Next, we present the description of such problems.

A. Standard Test Problems

The following standard test problems were used to assess the performance of our proposed approach.

$$\text{LIS [12]:} \quad \begin{aligned} f_1(x) &= \sqrt[8]{x_1^2 + x_2^2} \\ f_2(x) &= \sqrt[4]{(x_1 - 0.5)^2 + (x_2 - 0.5)^2} \end{aligned}$$

where $x_i \in [-5, 10]$. This MOP has a concave and connected Pareto front.

$$\text{FONSECA [6]:} \quad \begin{aligned} f_1(x) &= 1 - \exp(-\textstyle\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2) \\ f_2(x) &= 1 - \exp(-\textstyle\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2) \end{aligned}$$

where $n = 3$ and $x_i \in [-4, 4]$. This MOP has a concave and connected Pareto front.

$$\text{DTLZ5 [4]:} \quad \begin{aligned} f_1(x) &= \cos(\theta_1)\cos(\theta_2)h(x) \\ f_2(x) &= \cos(\theta_1)\sin(\theta_2)h(x) \\ f_3(x) &= \sin(\theta_1)h(x) \\ \theta_1 &= \tfrac{\pi}{2}x_1 \\ \theta_2 &= \tfrac{\pi}{4(1+g(x))}(1 + 2g(x)x_2) \\ g(x) &= \textstyle\sum_{i=k}^{n}(x_i - 0.5)^2 \\ h(x) &= 1 + g(x) \end{aligned}$$

where $k = 3, n = 12$ and $x_i \in [0, 1]$. This MOP has a degenerated Pareto front formed by a curve of well-distributed solutions.

B. Airfoil Shape Optimization: A case study

Our case study consists of the multi-objective optimization of an airfoil shape problem adapted from [21] (called here MOPRW). This problem corresponds to the airfoil shape optimization of a standard-class glider, aiming to obtain an optimum performance for a sailplane. In this study the trade-off among two aerodynamic objectives is evaluated using our proposed approach, and its results are compared with respect to those obtained by MOEA/D.

1) Problem Statement: Two conflicting objective functions are defined in terms of a sailplane average weight and operating conditions [21]. They are defined as:

i) minimize: C_D/C_L
s.t. $C_L = 0.63$, $Re = 2.04 \cdot 10^6$, $M = 0.12$

ii) minimize: $C_D/C_L^{3/2}$
s.t. $C_L = 1.05$, $Re = 1.29 \cdot 10^6$, $M = 0.08$

where C_D/C_L and $C_D/C_L^{3/2}$ correspond to the inverse of the glider's gliding ratio and sink rate, respectively. Both are important performance measures for this aerodynamic optimization problem. C_D and C_L are the drag and lift coefficients.

The aim is to maximize the gliding ratio (C_L/C_D) for objective (*i*), while minimizing the sink rate in objective (*ii*). Each of these objectives is evaluated at different prescribed flight conditions, given in terms of Mach and Reynolds numbers. The aim of solving this MOP is to find a better airfoil shape, which improves a reference design.

2) Geometry Parametrization: In the present case study, the PARSEC airfoil representation [19] was adopted. Fig. 4 illustrates the 11 basic parameters used for this representation: r_{le} leading edge radius, X_{up}/X_{lo} location of maximum thickness for upper/lower surfaces, Z_{up}/Z_{lo} maximum thickness for upper/lower surfaces, Z_{xxup}/Z_{xxlo} curvature for upper/lower surfaces, at maximum thickness locations, Z_{te} trailing edge coordinate, ΔZ_{te} trailing edge thickness, α_{te} trailing edge direction, and β_{te} trailing edge wedge angle.

TABLE I
PARAMETER RANGES FOR MODIFIED PARSEC AIRFOIL REPRESENTATION

Design Variable	Lower Bound	Upper Bound
r_{leup}	0.0085	0.0126
r_{lelo}	0.0020	0.0040
α_{te}	7.0000	10.0000
β_{te}	10.0000	14.0000
Z_{te}	-0.0060	-0.0030
ΔZ_{te}	0.0025	0.0050
X_{up}	0.4100	0.4600
Z_{up}	0.1100	0.1300
Z_{xxup}	-0.9000	-0.7000
X_{lo}	0.2000	0.2600
Z_{lo}	-0.0230	-0.0150
Z_{xxlo}	0.0500	0.2000

For the present case study, the modified PARSEC geometry representation adopted allows us to define independently the leading edge radius, both for upper and lower surfaces. Thus, a total of 12 variables are used. Their allowable ranges are defined in Table I.

The PARSEC airfoil geometry representation uses a linear combination of shape functions for defining the upper and lower surfaces. These linear combinations are given by:

$$Z_{upper} = \sum_{n=1}^{6} a_n x^{\frac{n-1}{2}}, \quad Z_{lower} = \sum_{n=1}^{6} b_n x^{\frac{n-1}{2}} \qquad (6)$$

In the above equations, the coefficients a_n, and b_n are determined as functions of the 12 described geometric parameters, by solving two systems of linear equations, one for each surface. It is important to note that the geometric parameters r_{leup}/r_{lelo}, X_{up}/X_{lo}, Z_{up}/Z_{lo}, Z_{xxup}/Z_{xxlo}, Z_{te}, ΔZ_{te}, α_{te}, and β_{te} are the actual design variables in the optimization process, and that the coefficients a_n, b_n serve as intermediate variables for interpolating the airfoil's coordinates, which are used by the CFD solver (we used the Xfoil CFD code [5]) for its discretization process.

V. COMPARISON OF RESULTS

A. Performance Measures

1) Hypervolume: The *Hypervolume* ($\mathcal{H}v$) metric was proposed by Zitzler [25]. This performance measure is Pareto compliant [26] and quantifies the approximation of nondominated solutions to the Pareto optimal front. The hypervolume corresponds to the non-overlapped volume of all the hypercubes formed by a reference point r (given by

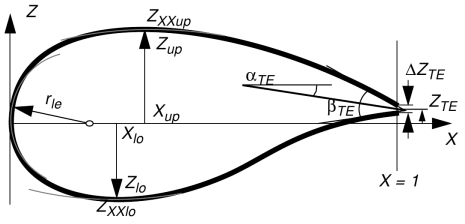


Fig. 4. PARSEC airfoil parametrization.

the user) and each solution p in the Pareto set approximation ($\mathcal{PF}_k$). It is mathematically defined as:

$$\mathcal{H}v = \Lambda\left(\bigcup_{p\in\mathcal{PF}_k}\{x|p \prec x \prec r\}\right) \quad (7)$$

where Λ denotes the Lebesgue measure and $r \in \mathbb{R}^k$ denotes a reference vector being dominated by all valid candidate solutions in PF_k.

2) Spacing: The *Spacing* ($\mathcal{S}p$) metric proposed by Schott [18], quantifies the spread of solutions in the obtained approximation of the Pareto front. This performance measure can be calculated as:

$$\mathcal{S}p = \sqrt{\frac{1}{|P|}\sum_{i=1}^{|P|}(\overline{d} - d_i)^2} \quad (8)$$

where d_i and $\overline{d}$ are defined as:

$$d_i = \min_{i,i\neq j}\sum_{k=1}^{M}|f_k^i - f_k^j|, \text{ and } \overline{d} = \frac{\sum_{i=1}^{|P|} d_i}{|P|}$$

A value of zero for this performance measure indicates that all the solutions are uniformly spread (i.e., the best possible performance).

B. Experimental Setup

As indicated before, we compared our proposed approach with respect to MOEA/D [24] (which uses the PBI approach). For each MOP, 30 independent runs were performed with each approach. The parameters for both algorithms are summarized in Table II, where N_{sol} represents the number of initial solutions (100 for bi-objective problems and 300 for three-objective problems). N_{it} represents the number of iterations (which was set to 40). Therefore, we performed $4,000$ (for the bi-objective problems) and $12,000$ (for the three-objective problems) function evaluations. For NSS-MO, α, β and γ represent the control parameters for the reflection, expansion and contraction movements of the NSS method. For MOEA/D, the parameters T_n, η_c, η_m, P_c and P_m represent the neighborhood size, crossover index, mutation index, crossover rate and mutation rate, respectively. Finally, the parameter θ, represents the penalty value used in the PBI approach for both the NSS-MO and MOEA/D.

For each MOP, the algorithms were evaluated using the two performance measures previously defined. (*Hypervolume* and *Spacing*). Regarding the hypervolume, the reference vectors adopted were: $r = (1,1)^T$ for the LIS problem, and $r = (1.1,\ldots,1.1)^T$ for the FONSECA and DTLZ5 problems, while for the airfoil design problem the reference vector $r = (0.007610, 0.005236)^T$ was employed. The results obtained are summarized in Tables III and IV. Each table displays both the $average$ and the standard deviation (σ) of each performance measure for each MOP. For an easier interpretation, the best results are presented in **boldface** for each performance measure and test problem adopted.

TABLE II
PARAMETERS FOR NSS-MO AND MOEA/D

Parameter	NSS-MO	MOEA/D
N_{sol}	100/300	100/300
N_{it}	40	40
T_n	–	30
P_c	–	1
P_m	–	$1/n$
α	1	–
β	2	–
γ	0.5	–
θ	5	5

TABLE III
RESULTS OF THE $\mathcal{H}v$ METRIC FOR NSS-MO AND MOEA/D

MOP	NSS-MO	MOEA/D
	$average$ (σ)	$average$ (σ)
LIS	**0.309713** (0.007686)	0.262384 (0.008786)
FONSECA	**0.542006** (0.001476)	0.374342 (0.007427)
DTLZ5	**0.429676** (0.000917)	0.425734 (0.001480)
MOPRW	**2.055961e-07** (3.045320e-08)	2.045719e-07 (2.511415e-08)

TABLE IV
RESULTS OF THE $\mathcal{S}p$ METRIC FOR NSS-MO AND MOEA/D

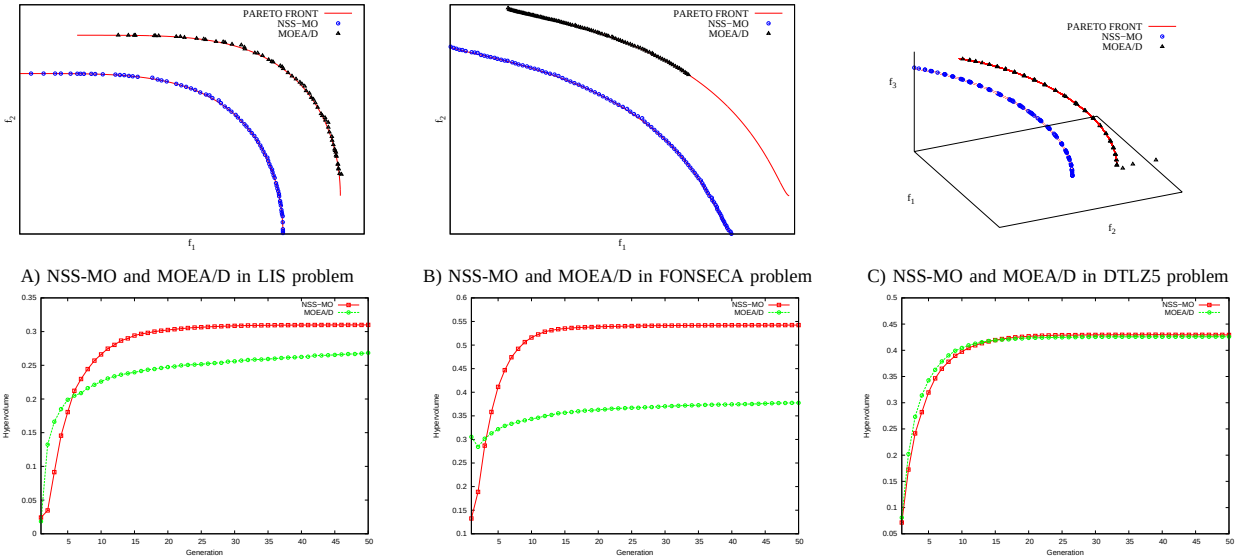
MOP	NSS-MO	MOEA-D
	$average$ (σ)	$average$ (σ)
LIS	**0.005861** (0.000812)	0.010273 (0.003768)
FONSECA	0.004454 (0.000218)	**0.003346** (0.000498)
DTLZ5	**0.007064** (0.001211)	0.034209 (0.031652)
MOPRW	1.554402e-05 (7.141540e-06)	**2.397431e-06** (2.790701e-06)

C. Results and Discussion

In this section, we present the results obtained by the proposed algorithm for each adopted test problem including the airfoil design problem. As indicated before, the results obtained by our proposed approach (NSS-MO) are compared against those produced by MOEA/D.

Table III shows the results obtained for the hypervolume ($\mathcal{H}v$) performance measure. From this table, it can be seen that the results obtained by our proposed approach outperformed MOEA/D in all the adopted MOPs. This indicates that our proposed algorithm produced a better approximation to the Pareto optimal front. Regarding the spacing ($\mathcal{S}p$) performance measure, Table IV shows that MOEA/D obtained better results than those produced by our proposed NSS-MO in the FONSECA and MOPRW problems, while NSS-MO was better in the LIS and DTLZ5 test problems. However, in our case, convergence was considered to be a more important criterion than uniform distribution, and regarding convergence, our proposed approach obtained better results in all cases.

Fig. 5 graphically shows that NSS-MO obtained both a better distribution of solutions and a better approximation



A) NSS-MO and MOEA/D in LIS problem

B) NSS-MO and MOEA/D in FONSECA problem

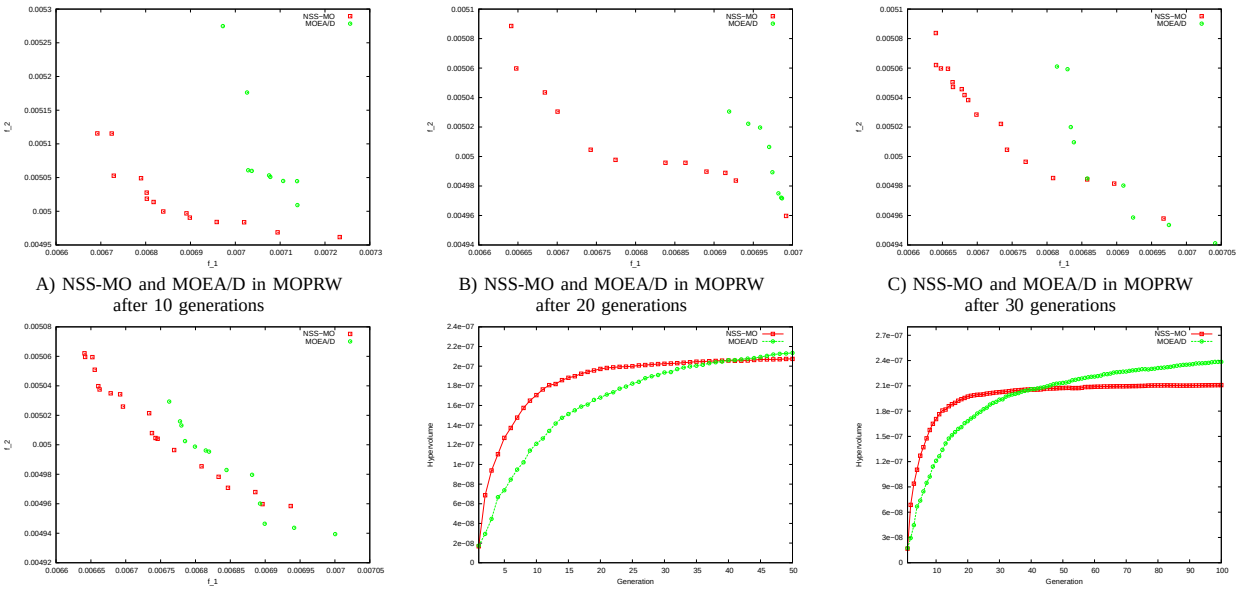
C) NSS-MO and MOEA/D in DTLZ5 problem

D) Hypervolume convergence for LIS problem

E) Hypervolume convergence for FONSECA problem

F) Hypervolume convergence for DTLZ5 problem

Fig. 5. Comparison of results for NSS-MO and MOE/D in the benchmark problems. Figures A to C show the achieved Pareto front, while Figures D to F show the convergence plot of the hypervolume for the LIS, FONSECA and DTLZ5 problems, respectively.



A) NSS-MO and MOEA/D in MOPRW after 10 generations

B) NSS-MO and MOEA/D in MOPRW after 20 generations

C) NSS-MO and MOEA/D in MOPRW after 30 generations

D) NSS-MO and MOEA/D in MOPRW after 40 generations

E) Hypervolume convergence for MOPRW problem after 50 generations

F) Hypervolume convergence for MOPRW problem after 100 generations

Fig. 6. Comparison of results for NSS-MO and MOE/D in the airfoil design problem. Figures A to D compare the achieved Pareto front for $10, 20, 30$ and 40 generations, while figures E and F show the convergence plot of the hypervolume at 50 and 100 generations, respectively.

to the Pareto optimal front in the LIS and FONSECA test problems. The differences in the distribution of solutions for DTLZ5 do not seem significant when comparing the two approaches, although MOEA/D was better in this case. It is worth noting that both approaches converged to the true Pareto front, but their approximations are shown at different positions in Fig. 5 to make them more readable. That is the reason why the axes of these plots do not have any numerical values.

Fig. 6 shows the approximation of the Pareto front generated by both NSS-MO and MOEA/D after $10, 20, 30$ and 40 iterations. In this case, NSS-MO obtained, on average, better hypervolume values than MOEA/D (see Table III), although the difference is not significant. In fact, Fig. 6 shows a particular instance in which MOEA/D obtained a better hypervolume value than our approach after 100 iterations.

The approximations to the Pareto front presented in Figs. 5 and 6, correspond to the set of nondominated solutions found by each algorithm in the run with the value nearest to the mean value of the $\mathcal{H}v$ metric for each multi-objective optimization problem adopted.

VI. Conclusions and Future Work

We have proposed a multi-objective algorithm based on mathematical programming techniques. The proposed approach was, in principle, designed for dealing with unconstrained, and unimodal problems having low and moderate dimensionality.

Our results indicate that our proposed NSS-MO algorithm outperforms MOEA/D regarding convergence in all the test problems adopted, including one from aeronautical engineering. The number of objective function evaluations in these test problems was low (4,000 for the bi-objective problems and 12,000 for the three-objective problems), which can make it a good choice for dealing with expensive objective functions. Our proposed approach has, however, some disadvantages. For example, when dealing with highly accidented search spaces, the movements of the NSS method may not be able to reach a better point during the search. Should that be the case, the step sizes (i.e., the control parameters α, β and γ) must be fine-tuned until finding a suitable search region.

In spite of the effectiveness of our proposed approach in MOPs with moderate dimensionality, our main goal is to hybridize it with a MOEA so that its use can be extended to problems of higher dimensionality and with highly accidented search spaces. The idea would be to use a MOEA for locating promising regions of the search space, and then apply NSS-MO for exploiting those regions in an efficient way. We believe that this sort of multi-objective memetic algorithm could be a powerful engine for solving complex and computationally expensive MOPs.

Acknowledgements

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 acknowledges support from IPN and CONACyT to pursue graduate studies at the Computer Science Department of CINVESTAV-IPN The third author gratefully acknowledges support from CONACyT project no. 103570.

References

[1] R. Chelouah and P. Siarry. Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. *European Journal of Operational Research*, 148(2):335–348, July 2003.

[2] I. Das. *Nonlinear Muicriteria Optimization and Robust Optimality*. PhD thesis, Rice University, Houston, Texas, 1997.

[3] 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.

[4] 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.

[5] M. Drela. XFOIL: An Analysis and Design System for Low Reynolds Number Aerodynamics. In *Conference on Low Reynolds Number Aerodynamics*, University Of Notre Dame, IN, June 1989.

[6] C. M. Fonseca and P. J. Fleming. Multiobjective Genetic Algorithms Made Easy: Selection, Sharing, and Mating Restriction. In *Proceedings of the First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications*, pages 42–52, Sheffield, UK, September 1995. IEE.

[7] F. Glover. Tabu search and adaptive memory programing – Advances, applications and challenges. In *Interfaces in Computer Science and Operations Research*, pages 1–75. Kluwer Academic Publishers, 1996.

[8] C. K. Goh, Y. S. Ong, K. C. Tan, and E. J. Teoh. An Investigation on Evolutionary Gradient Search for Multi-Objective Optimization. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 3742–3747, Hong Kong, June 2008. IEEE Service Center.

[9] A. G. Hernández-Díaz, C. A. Coello Coello, F. Pérez, R. Caballero, J. Molina, and L. V. Santana-Quintero. Seeding the Initial Population of a Multi-Objective Evolutionary Algorithm using Gradient-Based Information. In *2008 Congress on Evolutionary Computation (CEC'2008)*, pages 1617–1624, Hong Kong, June 2008. IEEE Service Center.

[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] J. Lis and A. E. Eiben. A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In T. Fukuda and T. Furuhashi, editors, *Proceedings of the 1996 International Conference on Evolutionary Computation*, pages 59–64, Nagoya, Japan, 1996. IEEE.

[13] J. B. MacQueen. Some Methods for Classification and Analysis of Multivariate Observations. In *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[14] K. I. M. McKinnon. Convergence of the Nelder–Mead Simplex Method to a Nonstationary Point. *SIAM Journal on Optimization*, 9(1):148–158, 1998.

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

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

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

[19] H. Sobieczky. Parametric Airfoils and Wings. In K. Fuji and G. S. Dulikravich, editors, *Notes on Numerical Fluid Mechanics, Vol.. 68*, pages 71–88, Wiesbaden, 1998. Vieweg Verlag.

[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] A. Szöllös, M. Smíd, and J. Hájek. Aerodynamic optimization via multi-objective micro-genetic algorithm with range adaptation, knowledge-based reinitialization, crowding and epsilon-dominance. *Advances in Engineering Software*, 40(6):419–430, 2009.

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

[23] S. Zapotecas Martínez and C. A. Coello Coello. A Proposal to Hybridize Multi-Objective Evolutionary Algorithms with Non-Gradient Mathematical Programming Techniques. In G. Rudolph, T. Jansen, S. Lucas, C. Poloni, and N. Beume, editors, *Parallel Problem Solving from Nature–PPSN X*, volume 5199, pages 837–846. Springer. Lecture Notes in Computer Science, Dortmund, Germany, September 2008.

[24] 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.

[25] 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.

[26] 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.