MONSS: A Multi-Objective Nonlinear Simplex Search

Saúl Zapotecas Martínez and Carlos A. Coello Coello

CINVESTAV-IPN (Evolutionary Computation Group)
Departamento de Computación
México D.F. 07300, MÉXICO
saul.zapotecas@gmail.com, ccoello@cs.cinvestav.mx

Abstract. This paper present a novel methodology 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, which define each, a scalarization problem, that is solved by deforming a simplex according to the movements described by Nelder and Mead's algorithm. Considering a MOP with n decision variables, the simplex is constructed using $n+1$ solutions which minimize different scalarization problems defined by $n+1$ neighbor weighted vectors. All 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. In this work, we 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 against which we compare our results.

Keywords: Multi-objective optimization, Nonlinear Simplex Search.

1 Introduction

In engineering and scientific applications commonly there exist problems can be stated as *multi-objective optimization problems* (MOPs). Such problems define a vector function whose elements represent the objective functions, which, represent a mathematical description of performance criteria. As the measures of the objectives usually are in conflict with each other, no unique best solution may exist, and then, good trade-offs among the objectives, which are obtained by using the definition of Pareto optimality, should be achieved. Such definition will provide us to obtain not one, but a set of (Pareto) optimal solutions (the as well know Pareto optimal set, PS).

Throughout the years, several mathematical programming techniques for dealing with MOPs have been proposed. Most of theses methods, transform a MOP into a single-objective scalarization function, in which, the objective is an aggregation of all objective functions f_i's (the well known scalarization approaches). Once the aggregating function is formulated, a mathematical programming method is employed for finding a Pareto solution. These mathematical methods have shown to be an effective tool in many domains, at a reasonably low computational cost. However, they have several limitations, including the fact of obtaining a single Pareto solution per run, and that most of them cannot properly deal with nonconvex, multi-modal or non-differentiable optimization problems. That has motivated the development of stochastic methods, such as the so-called *multi-objective evolutionary algorithms* (MOEAs), which, for their simplicity and ease of use have became very popular and applicable in many optimization problems, see for example [1].

Based on certain mathematical foundations, and assuming some assumptions, the development of multi-objective programming techniques has been encouraged since early days of multi-objective optimization. In the specialized literature, there exist more than 30 multi-objective programming techniques, and in the last few years the design of new methods have drawn the interest of some researchers. Recently, Fliege et al. [2] proposed an extension of Newton's method for unconstrained multi-objective optimization. Fischer and Shukla presented an algorithm based on Levenberg-Marquardt method to solve unconstrained MOPs [3]. Although the existence of this sort of methods dates back more than three decades (see for example [4]). Unfortunately when the function gradient is not available, the above mentioned methods become impractical, and then, a directed search technique needs to be adopted (i.e. a method that do not require gradient information).

The use of directed search methods is scarce in the multi-objective context, although some researchers have used them as local search operators into MOEAs (see for example [5–7]). However, to the previous knowledge of the authors, there is not a full methodology to approximate multiple solutions towards the Pareto set (maintaining a good representation of the Pareto front) by using just non-gradient mathematical programming techniques. One of the main reasons of the shortage in such strategies, is that it is not efficient to approximate different solutions towards different parts of the Pareto front. These drawbacks have naturally motivated the idea of hybridizing either gradient or non-gradient mathematical programming techniques with MOEAs. 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 [8]), and it is precisely the focus of this work.

In this paper, we present a novel methodology for dealing with unconstrained MOPs based on direct search methods. The proposed approach analyzes and exploits the properties of Nelder and Mead's method [9] (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 approximation by means of movements guided by mathematical programming techniques, while maintaining a reasonably good representation of the Pareto front. Preliminary studies show that the proposed approach is computationally efficient (in terms of the objective function evaluations that it performs) and produces competitive results when dealing with MOPs of low and moderate dimensionality. As it will seen after, our proposed approach showed competitive results when it was compared with a current state-of-the-art MOEA.

The remainder of this paper is organized as follows. In Section 2, we introduce the basic concepts required for understanding the rest of the paper. Section 3 shows the original template of the Nelder and Mead's algorithm adopted in this work. In Section 4, we describe in detail our proposed approach. Section 5 shows the results obtained by our proposed approach. Finally, in Section 6, we provide our conclusions and some possible paths for future research.

2 Multi-objective Optimization

Without loss of generality, an unconstrained multi-objective optimization problem (MOP), can be stated as [1]:

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

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

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

where $f_i : \mathbb{R}^n \to \mathbb{R}$ is an 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). Therefore, in order to describe the concept of optimality in which we are interested, the following definitions are introduced [10].

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

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

Definition 3. The *Pareto optimal set PS* is defined by:

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

Definition 4. For a Pareto optimal set PS, the *Pareto optimal front PF* is defined as:

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

[1] assuming minimization

As in most of multi-objective algorithm, 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.

3 The Nonlinear Simplex Search

Nelder and Mead's method [9] also known as the *Nonlinear Simplex Search* (NSS), is an algorithm based on the simplex algorithm of Spendley et al. [11], 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 in this work.

Definition 4. A *simplex* or *n-simplex* $\boldsymbol{\Delta}$ is a convex hull of a set of $n+1$ affinely independent points $\boldsymbol{\Delta}_i$ $(i = 1, \ldots, n{+}1)$, 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 Nelder and Mead's method is fully 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: $f(\boldsymbol{\Delta}_1) \leq f(\boldsymbol{\Delta}_2) \leq \cdots \leq f(\boldsymbol{\Delta}_{n+1})$. In this way, the movements performed in the simplex by the NSS method are defined as:

1. *Reflection*: $\boldsymbol{x}_r = (1 + \alpha)\boldsymbol{\Delta}_c - \alpha\boldsymbol{\Delta}_{n+1}$.
2. *Expansion*: $\boldsymbol{x}_e = (1 + \alpha\gamma)\boldsymbol{\Delta}_c - \alpha\gamma\boldsymbol{\Delta}_{n+1}$.
3. *Contraction*:
   (a) *Outside*: $\boldsymbol{x}_{co} = (1 + \alpha\beta)\boldsymbol{x}_c - \alpha\beta\boldsymbol{\Delta}_{n+1}$.
   (b) *Inside*: $\boldsymbol{x}_{ci} = (1 - \beta)\boldsymbol{x}_c + \beta\boldsymbol{\Delta}_{n+1}$.

where $\boldsymbol{x}_c = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\Delta}_i$ is the centroid of the n best points (all vertices except for $\boldsymbol{\Delta}_{n+1}$), $\boldsymbol{\Delta}_1$ and $\boldsymbol{\Delta}_{n+1}$ are the best and the worst solutions identified within the simplex, respectively. Figure 1 shows all the possible movements performed 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 $f(\boldsymbol{\Delta}_1) \leq f(\boldsymbol{x}_r) \leq f(\boldsymbol{\Delta}_n)$, then $\boldsymbol{\Delta}_{n+1} = \boldsymbol{x}_r$.
2. If $f(\boldsymbol{x}_e) < f(\boldsymbol{x}_r) < f(\boldsymbol{\Delta}_1)$, then $\boldsymbol{\Delta}_{n+1} = \boldsymbol{x}_e$, otherwise $\boldsymbol{\Delta}_{n+1} = \boldsymbol{x}_r$.
3. If $f(\boldsymbol{\Delta}_n) \leq f(\boldsymbol{x}_r) < f(\boldsymbol{\Delta}_{n+1})$ and $f(\boldsymbol{x}_{co}) \leq f(\boldsymbol{x}_r)$, then $\boldsymbol{\Delta}_{n+1} = \boldsymbol{x}_{co}$.
4. If $f(\boldsymbol{x}_r) \geq f(\boldsymbol{\Delta}_{n+1})$ and $f(\boldsymbol{x}_{ci}) < f(\boldsymbol{\Delta}_{n+1})$, then $\boldsymbol{\Delta}_{n+1} = \boldsymbol{x}_{ci}$.

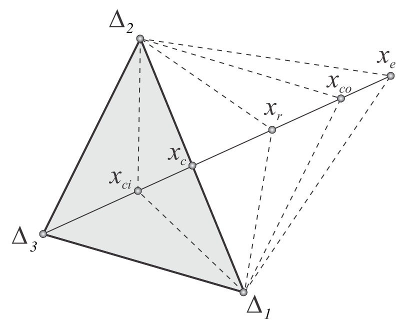


Fig. 1. Illustration of the possible movements in the simplex performed by the NSS method. The constructed simplex corresponds to an optimization problem with two decision variables, where Δ_1 and Δ_3 are the best and the worst points, respectively.

4 The Nonlinear Simplex Search for Unconstrained Multi-Objective Optimization

4.1 Decomposing Multi-objective Optimization Problem

There are several approaches for transforming a MOP into a single-objective optimization subproblem [12, 10, 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 solution is achieved by solving such subproblem. 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 [14, 15], the approaches based on boundary intersection possess certain advantages over those based on either Tchebycheff or the Weighted Sum. In the following, a boundary intersection approach adopted in this work is described.

The Penalty Boundary Intersection: The Penalty Boundary Intersection (PBI) approach[2] was proposed by Zhang and Li [15], and uses a weighted vector $\boldsymbol{w}$

[2] PBI is based on the well-known Normal Boundary Intersection (NBI) method [14]

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 $\boldsymbol{F}(\boldsymbol{x})$. Mathematically, the PBI problem can be formulated as follows:

Let $\boldsymbol{w} = (w_1, \ldots, w_k)^T$ 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:

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

such that:

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

where $\boldsymbol{x} \in \mathbb{R}^n$, θ is the penalty value and $\boldsymbol{z}^\star = (z_1^\star, \ldots, z_k^\star)^T$ is the utopian vector, i.e., $\boldsymbol{z}^\star = \min\{f_i(\boldsymbol{x})|\boldsymbol{x} \in \Omega\}$ for each $i = 1, \ldots, k$. Figure 2 illustrates the PBI for a bi-objective optimization problem.

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 weighted vectors, which define the search direction in the optimization process. This strategy is employed in this work, and its mode of use is described in Section 4.3.

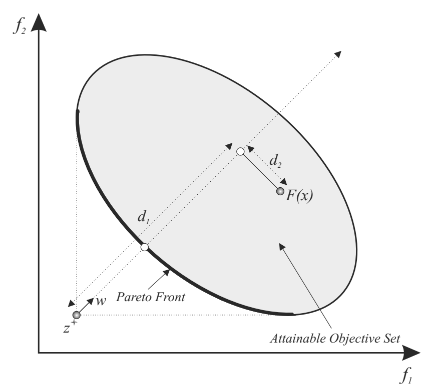


Fig. 2. Illustration of the Penalty Boundary Intersection (PBI) approach.

4.2 About the Nonlinear Simplex Search and Multi-objective Optimization Problems

Mathematical programming techniques are know to have several limitations compared with respect to evolutionary algorithms. As mentioned before, most of these strategies are designed to deal with convex functions and usually require the gradient information. Nelder and Mead's method does not require the gradient information, instead of this, the NSS algorithm trusts in obtaining a better solution by deforming a simplex shape along the search process. Nonetheless, the Nelder and Mead's method possess a strong disadvantage: 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 [16]). 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, modifying the movements into the simplex, etc.). In recent years, several attempts to improve the NSS method have been reported in the literature (see for example [17–20]). However, for its inherent nature (based on heuristic movements), these modifies implement badly or even fails for certain optimization problems. But not just these improvements have been reported in the literature, but also different strategies for the construction of the simplex also have been explored by several researchers (see for example [21, 6]).

The construction of the simplex plays an important role in the performance of the simplex search 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 an appropriate 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 [22]. However, the use of a degenerated simplex could obtain local minima, 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 optimization subproblems and assuming that each subproblem is solved throughout the search, then, the simplex could be constructed using such solutions. In this way, multiple trade-off solutions are achieved while the search eventually converges to the region in which the Pareto optimal set is contained.

The convergence towards a better point given in the Nelder and Mead's method should be achieved at most in $n + 1$ iterations (at least in convex functions with low dimensionality) [22]. Thus, for solving each subproblem (of the decomposition) a considerable number of function evaluations could be required. Therefore, an appropriate strategy for approximating solutions to the Pareto optimal set needs to be adopted.

The above observations are taken into account and they are used to design an effective nonlinear simplex search approach for unconstrained multi-objective optimization. The proposed methodology is described in the next section.

4.3 The Multi-Objective Nonlinear Simplex Search

The proposed *Multi-objective Nonlinear Simplex Search* (MONSS) decomposes a MOP into several single-objective scalarization subproblems. Therefore, a well-distributed set of weighted vectors $W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N\}$ has to be previously defined. In this work, we use the same method as in [15], however, other methods can be used, see for example [23].

At the beginning, a set of N solutions $\mathcal{S} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ having an uniform distribution is randomly initialized. Each vector $\boldsymbol{x}_i \in \mathcal{S}$ represent a solution for the i^{th} subproblem defined by the i^{th} weighted vector $\boldsymbol{w}_i \in W$. In this way, different subproblems are simultaneously solved by the MONSS algorithm and the set of solutions $\mathcal{S}$ will constitute an approximate to the Pareto optimal set lengthwise of the search process. In order to find different solutions along the Pareto front, 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 $\boldsymbol{w}_c \in C_i$ are contiguous among themselves.

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

$$\boldsymbol{\Delta} = \{\boldsymbol{x}_s, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\} \tag{4}$$

such that: $\boldsymbol{x}_s \in \mathcal{S}$ is a minimum of $g(\boldsymbol{x}_s|\boldsymbol{w}_s, \boldsymbol{z}^\star)$ for any $\boldsymbol{w}_s \in W$. $\boldsymbol{x}_j \in \mathcal{S}$ represents the n solutions that minimize the subproblems defined by the nearest n weighted vectors of $\boldsymbol{w}_s$, 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, $\boldsymbol{x}_n$, leaves the search space. In order to deal with this problem, (as in [6]) we bias deterministically the boundaries. Therefore, the i^{th} bound of the new solution $\boldsymbol{x}_n$ is re-established as follows:

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

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

To speed up the convergence towards the Pareto set, the search is relaxed at each iteration by changing the direction vector for any other direction $\hat{\boldsymbol{w}}_s \in C_i$. In this way, an agile search into the partition C_i is performed and collapsing the simplex search in the same direction $\boldsymbol{w}_s$ is avoided. 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 [24].

One iteration of the MONSS is carried out, when the simplex search 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* ($\mathcal{I}$). At the end of each iteration, the 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.

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

Input:
W: A well-distributed set of weighted vectors.
$\mathcal{I}$: The intensification set.
$\mathcal{S}$: The current approximation to the Pareto set.
Output:
$\mathcal{R}$: An approximation to the Pareto front.

begin
    $\mathcal{T} = \mathcal{S} \cup \mathcal{I}$;
    $\mathcal{R} = \emptyset$;
    forall the $\boldsymbol{w}_i \in W$ **do**
        $\mathcal{R} = \mathcal{R} \cup \{\boldsymbol{x}^\star | \min_{\boldsymbol{x}^\star \in \mathcal{T}} g(\boldsymbol{x}^\star | \boldsymbol{w}_i, \boldsymbol{z}^\star)\}$;
        $\mathcal{T} = \mathcal{T} \setminus \{\boldsymbol{x}^\star\}$;
    end
    return $\mathcal{R}$;
end

In this way, the simplex search 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 [25]. In Figure 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 MONSS algorithm can be stated as in Algorithm 2.

5 Experimental Studies

5.1 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, the well know "*Multi-Objective Evolutionary Algorithm based on Decomposition*" (MOEA/D), which has shown a good performance compared with respect to other MOEAs [15]. Similar to MONSS, MOEA/D decomposes a MOP into several scalarization problems. However, instead of using mathematical programming methods, MOEA/D uses genetic operators to approximate solutions to the Pareto set (for more details see [15]).

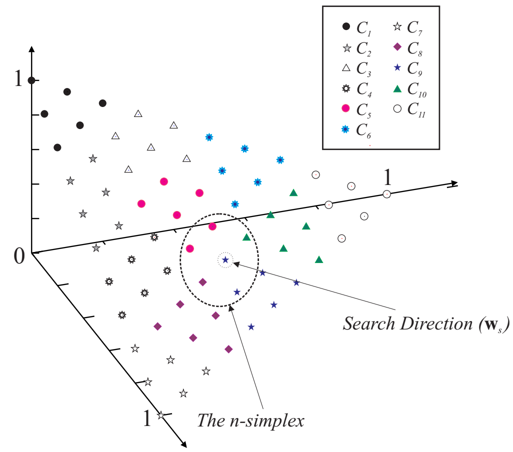


Fig. 3. Illustration of a well-distributed set of weighted vectors for a MOP with three objectives, five decision variables and 66 weighted vectors, i.e. $m = \frac{|W|}{n+1} = 11$ partitions. The n-simplex is constructed by six solutions contained in four different partitions (C_5, C_8, C_9 and C_{10}).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

In our experiments, we adopted ten MOPs with two and three objectives. The different characteristics in their Pareto optimal front and the definition of such problems are summarized in Table 1.

5.2 Performance Measures

In order to assess the performance of our proposed approach, we compared with respect to MOEA/D by using the following performance measures.

Hypervolume: The Hypervolume ($\mathcal{H}v$) metric was proposed by Zitzler [34]. This performance measure is Pareto compliant [35], and quantifies both approximation and distribution of nondominated solutions along the Pareto front. The hypervolume corresponds to the non-overlapped volume of all the hypercubes formed by a reference point $\boldsymbol{r}$ (given by the user) and each solution $\boldsymbol{p}$ in the Pareto set approximation (PF_k). It is mathematically stated as:

$$\mathcal{H}v(PF_k) = \Lambda\left(\bigcup_{p \in PF_k} \{\boldsymbol{x} | \boldsymbol{p} \prec \boldsymbol{x} \prec \boldsymbol{r}\}\right) \qquad (6)$$

where Λ denotes the Lebesgue measure and $\boldsymbol{r} \in \mathbb{R}^k$ 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. [36], and it compares a set of non-dominated solutions A with respect to another set B, using Pareto dominance. This performance measure is defined as:

Algorithm 2: The flowchart of the *multi-objective nonlinear simplex search* algorithm

Input:
$W = \{\boldsymbol{w}_1, \ldots, \boldsymbol{w}_N\}$: A set of N weighted vectors.
max_{it}: A maximum number of iterations.
Output:
$\mathcal{S}$: An approximation to the Pareto front.

begin
- $t = 0$;
- **Generate initial solutions:** Generate a set $\mathcal{S}^t = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ of N random solutions;
- **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;
- **while** $t < max_{it}$ **do**
  - **for** $i = 0$ *to* m **do**
    - **Randomly choose** $\boldsymbol{w}_s \in C_i$;
    - **Apply Simplex Search method:**
      - *a)* **Build the n-simplex:** Construct the n-simplex from $\mathcal{S}^t$, such that: eq.(4) is satisfied.
      - *b)* **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 search direction $\boldsymbol{w}_s$ for any other $\hat{\boldsymbol{w}}_s \in C_i$.
        - * Each new solution generated by any movements of the NSS method is stored in the intensification set $\mathcal{I}$.
  - **end**
  - **Update the leading set:** Update the set $\mathcal{S}$ using Algorithm 1. That is: $\mathcal{S}^{t+1} = update(W, \mathcal{S}^t, \mathcal{I})$;
  - $t = t + 1$;
- **end**
- **return** $\mathcal{S}^t$;

end

$$\mathcal{SC}(A, B) = \frac{|\{\boldsymbol{b} \in B | \exists \boldsymbol{a} \in A : \boldsymbol{a} \preceq \boldsymbol{b}\}|}{|B|} \tag{7}$$

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 $\mathcal{SC}(B, A)$.

5.3 Parameter Settings

As indicated before, we compared our proposed approach with respect to MOEA/D [15] (using the PBI approach). For a fair comparison, the set of weighted vectors

MOP	Obj	Definition	PF features	MOP	Obj	Definition	PF features
DEB2 [26]	2	$f_1(\boldsymbol{x}) = x_1$ $f_2(\boldsymbol{x}, g(\boldsymbol{x})) = g(\boldsymbol{x}), \cdot h(\boldsymbol{x})$ and: $g(\boldsymbol{x}) = 1 + 10x_2$ $h(\boldsymbol{x}) = 1 - (f_1(\boldsymbol{x})/g(\boldsymbol{x}))^2 - \frac{f_1(\boldsymbol{x})}{g(\boldsymbol{x})} \times$ $\sin(12\pi f_1(\boldsymbol{x}))$ $x_i \in [0, 1]$	Nonconvex Disconnected	MUR [27]	2	$f_1(\boldsymbol{x}) = 2\sqrt{x_1}$ $f_2(\boldsymbol{x}) = x_1(1 + x_2) + 5$ $x_1 \in [1, 4], x_2 \in [1, 2]$	Convex Connected
DTLZ5 [28]	3	$f_1(\boldsymbol{x}) = \cos(\theta_1)\cos(\theta_2)h(\boldsymbol{x})$ $f_2(\boldsymbol{x}) = \cos\theta_1 \sin(\theta_2)h(\boldsymbol{x})$ $f_3(\boldsymbol{x}) = \sin(\frac{\pi}{2}x_1^{\alpha})h(\boldsymbol{x})$ $g(\boldsymbol{x}) = \sum_{i=3}^{n}(x_i - 0.5)^2$ $h(\boldsymbol{x}) = (1 + g(\boldsymbol{x}))$ $\alpha = \pi$ $x_i \in [0, 1], n = 12$	Nonconvex Connected	REN1 [29]	2	$f_1(\boldsymbol{x}) = \frac{1}{x_1^2 + x_2^2 + 1}$ $f_2(\boldsymbol{x}) = x_1^2 + 3x_2^2 + 1$ $x_i \in [-3, 3]$	Nonconvex Connected
FON2 [30]	2	$f_1(\boldsymbol{x}) = 1 - \exp(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2)$ $f_2(\boldsymbol{x}) = 1 - \exp(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2)$ $x_i \in [-4, 4]$	Nonconvex Connected	REN2 [29]	2	$f_1(\boldsymbol{x}) = x_1 + x_2 + 1$ $f_2(\boldsymbol{x}) = x_1^2 + 2x_2^2 - 1$ $x_i \in [-3, 3]$	Nonconvex Connected
LAU [31]	2	$f_1(\boldsymbol{x}) = x_1^2 + x_2^2$ $f_2(\boldsymbol{x}) = (x_1 + 2)^2 - x_2^2$ $x_i \in [-50, 50]$	Convex Connected	VNT2 [32]	3	$f_1(\boldsymbol{x}) = \frac{(x_1-2)^2}{2} + \frac{(x_2+1)^2}{13} + 3$ $f_2(\boldsymbol{x}) = \frac{(x_1+x_2-3)^2}{36} + \frac{(-x_1+x_2+2)^2}{8} - 17$ $f_3(\boldsymbol{x}) = \frac{(x_1+2x_2-1)^2}{175} + \frac{(2x_2-x_1)^2}{17} - 13$ $x_i \in [-4, 4]$	Nonconvex Connected
LIS [33]	2	$f_1(\boldsymbol{x}) = \sqrt[8]{x_1^2 + x_2^2}$ $f_2(\boldsymbol{x}) = \sqrt[4]{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}$ $x_i \in [-5, 10]$	Nonconvex Connected	VNT3 [32]	3	$f_1(\boldsymbol{x}) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$ $f_2(\boldsymbol{x}) = \frac{(3x_1-2x_2+4)^2}{8} + \frac{(x_1-x_2+1)^2}{27} + 15$ $f_3(\boldsymbol{x}) = \frac{1}{(x_1^2+x_2^2+1)} - 1.1\exp(-x_1^2 - x_2^2)$ $x_i \in [-3, 3]$	Nonconvex Connected

Table 1. Test problems

was the same for both algorithms. For each MOP, 30 independent runs were performed with each approach. The parameters for both algorithms are summarized in Table 2, where N_{sol} 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 40 for all test problems. Therefore, both algorithms performed $4,000$ (for the bi-objective problems) and $12,000$ (for the three-objective problems) function evaluations for each problem. For MONSS, α, β and γ represent the control parameters for the reflection, expansion and contraction movements of the NSS method, respectively. 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 MONSS and MOEA/D.

Parameter	MONSS	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	1/2	–
θ	5	5

Table 2. Parameters for MONSS and MOEA/D

For each MOP, the algorithms were evaluated using the two performance measures previously defined. (*Hypervolume* and *Two Set Coverage*). The results obtained are summarized in Table 3. The table displays 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 4. These vectors are established near 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 distribution is reported when the algorithms converge along the Pareto front. In the case of the statistics for the $\mathcal{SC}$ 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.

5.4 Discussion of results

The main goal of the simulation is to verify the effectiveness of the nonlinear simplex search when dealing with MOPs. As indicated before, the results obtained by our proposed approach (MONSS) were compared against those produced by MOEA/D.

Table 3 shows the results obtained for both the Hypervolume ($\mathcal{H}v$) and the Two Set Coverage ($\mathcal{SC}$) performance measures. From this table, it can be seen that the results obtained by the MONSS outperformed to MOEA/D in most of the adopted test problems. This means that the proposed approach reached a better approximation and distribution of solutions along the Pareto front. The exception was VNT2, where MOEA/D obtained a better value in the $\mathcal{H}v$ metric. However, given the small difference in this performance measure, we consider that MONSS was not significantly outperformed by MOEA/D in this problem. Thereby, our proposed approach became as competitive as MOEA/D.

Regarding the $\mathcal{SC}$ performance measure, MONSS obtained better results compared against those produced by MOEA/D in the majority of the test problems. This means that, the solutions obtained by MONSS dominated a higher ratio of solutions produced by MOEA/D. However, MOEA/D was better for DTLZ5 and REN1 problems, though the ratio of solutions dominated by MOEA/D was not significantly high. Although the SC performance measure benefits to MOEA/D in these problems, it is worth noting that our proposed approach reached better results in the $\mathcal{H}v$ performance measure. $\mathcal{H}v$ metric not only measures the convergence but also the distribution of solution along the Pareto front, that is a reason why the MONSS obtained better result in $\mathcal{H}v$ metric for DTL5 and REN1 problems, even when it was outperformed by MOEA/D in $\mathcal{SC}$ metric.

Finally, Figures 4 and 5 show the hypervolume convergence at each iteration of the algorithms. From these graphics, it is possible see the performance for both algorithms (MONSS and MOEA/D) was similar in most of the cases, and in some one more, MONSS approximated faster solutions to the Pareto front than MOEA/D. With that figures, we validated the effectiveness of our methodology when dealing MOPs with low and moderate dimensionality.

6 Conclusions and Future Work

We have proposed a novel methodology based on just mathematical programming techniques for approximating solutions along the Pareto front of a MOP. The proposed approach was, in principle, designed for dealing with unconstrained, and unimodal problems having low and moderate dimensionality (2, 3 and 12 decision variables).

Our results indicate that our proposed MONSS outperforms MOEA/D regarding convergence in most of the test problems adopted. The number of objective function evaluations in these test problems was restricted to 4,000 for the bi-objective problems and 12,000 for the three-objective problems (i.e. a low

MOP	$\mathcal{H}v$(MONSS)	$\mathcal{H}v$(MOEA/D)	$\mathcal{SC}$(MONSS, MOEA/D)	$\mathcal{SC}$(MOEA/D, MONSS)
	average (σ)	*average* (σ)	*average* (σ)	*average* (σ)
DEB2	**0.981552** (0.004504)	0.969845 (0.049164)	**0.190446** (0.053016)	0.146296 (0.035893)
DTLZ5	**0.429676** (0.000917)	0.426429 (0.001175)	0.210250 (0.019020)	**0.311705** (0.051739)
FON2	**0.542006** (0.001476)	0.539159 (0.001406)	**0.354962** (0.090241)	0.116333 (0.030275)
LAU	**13.934542** (0.008218)	13.868946 (0.029341)	**0.072572** (0.060321)	0.056333 (0.028459)
LIS	**0.309713** (0.007686)	0.259479 (0.009430)	**0.340798** (0.124927)	0.097992 (0.045691)
MUR	**3.141629** (0.003791)	3.140806 (0.001290)	**0.147827** (0.058459)	0.092632 (0.011971)
REN1	**3.612650** (0.000958)	3.596241 (0.019682)	0.105443 (0.053141)	**0.146599** (0.042929)
REN2	**18.925039** (0.016614)	18.918943 (0.023277)	**0.026274** (0.022809)	0.013468 (0.006563)
VNT2	2.113570 (0.003068)	**2.114601** (0.002688)	**0.080900** (0.014464)	0.057426 (0.011687)
VNT3	**11.685911** (0.013195)	11.599974 (0.018481)	**0.029109** (0.012831)	0.000501 (0.001278)

Table 3. Results of $\mathcal{H}v$ and $\mathcal{SC}$ performance measures for MONSS and MOEA/D

MOP	$\boldsymbol{r}$	MOP	$\boldsymbol{r}$
DEB2	$(1.1, 1.1)^T$	MUR	$(4.1, 4.1)^T$
DTLZ5	$(1.1, 1.1, 1.1)^T$	REN1	$(37.1, 1.1)^T$
FON2	$(1.1, 1.1)^T$	REN2	$(-1.9, 2.1)^T$
LAU	$(4.1, 4.1)^T$	VNT2	$(4.5, -16.0, -11.5)^T$
LIS	$(1, 1)^T$	VNT3	$(8.5, 17.5, 0.5)^T$

Table 4. Reference vectors for $\mathcal{H}v$ performance metric

number of evaluations), which can make it a good choice for dealing with expensive objective functions. With that we are show that it is possible design a competitive algorithm by using just directed search methods. 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.

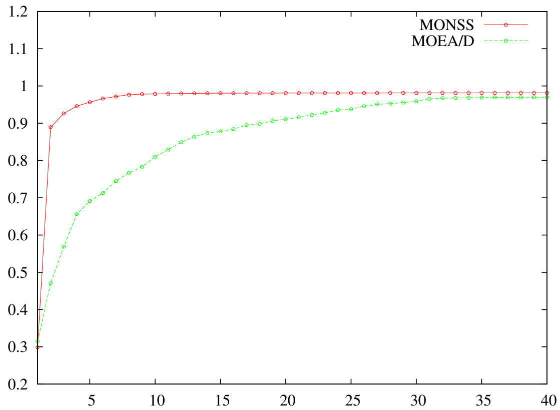
In spite of the effectiveness of our proposed approach in MOPs with low and moderate dimensionality, our main goal is to hybridize it with a MOEA so that its use can be extended to problems of higher dimensionality and with highly accidented search spaces. The idea would be to use a MOEA for locating promising regions of the search space, and then apply MONSS 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.

References

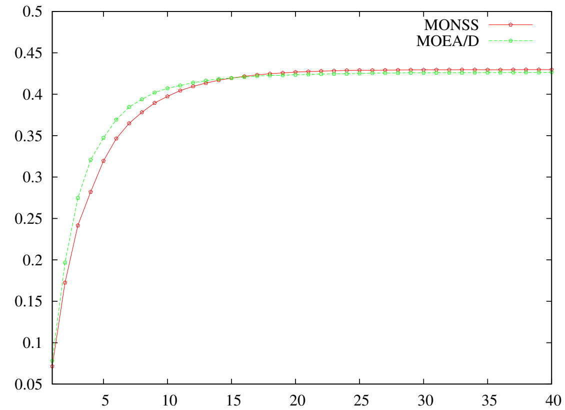
1. Coello Coello, C.A., Lamont, G.B., eds.: Applications of Multi-Objective Evolutionary Algorithms. World Scientific, Singapore (2004) ISBN 981-256-106-4.
2. Fliege, J., Graña Drummond, L.M., Svaiter, B.F.: Newton's Method for Multiobjective Optimization. SIAM Journal on Optimization **20**(2) (2009) 602–626
3. Fischer, A., Shukla, P.K.: A levenberg-marquardt algorithm for unconstrained multicriteria optimization. Oper. Res. Lett. **36**(5) (2008) 643–646
4. Mukai, H.: Algorithms for Multicriterion Optimization. IEEE Transactions on Automatic Control **25**(2) (1980) 177–186
5. Koduru, P., Das, S., Welch, S.M.: Multi-Objective Hybrid PSO Using ϵ-Fuzzy Dominance. In Thierens, D., ed.: 2007 Genetic and Evolutionary Computation Conference (GECCO'2007). Volume 1., London, UK, ACM Press (2007) 853–860
6. Zapotecas Martínez, S., Coello Coello, C.A.: A Proposal to Hybridize Multi-Objective Evolutionary Algorithms with Non-Gradient Mathematical Programming Techniques. In Rudolph, G., Jansen, T., Lucas, S., Poloni, C., Beume, N., eds.: Parallel Problem Solving from Nature–PPSN X. Volume 5199. Springer. Lecture Notes in Computer Science, Dortmund, Germany (2008) 837–846
7. Zapotecas Martínez, S., Coello Coello, C.A.: A Memetic Algorithm with Non Gradient-Based Local Search Assisted by a Meta-Model. In Schaefer, R., Cotta, C., Kolodziej, J., Rudolph, G., eds.: Parallel Problem Solving from Nature–PPSN XI. Volume 6238 of Lecture Notes in Computer Science., Kraków, Poland, Springer (2010) 576–585
8. Hernández-Díaz, A.G., Coello Coello, C.A., Pérez, F., Caballero, R., Molina, J., Santana-Quintero, L.V.: Seeding the Initial Population of a Multi-Objective Evolutionary Algorithm using Gradient-Based Information. In: 2008 Congress on Evolutionary Computation (CEC'2008), Hong Kong, IEEE Service Center (2008) 1617–1624
9. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. The Computer Journal **7** (1965) 308–313
10. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston, Massachusetts (1999)
11. Spendley, W., Hext, G.R., Himsworth, F.R.: Sequential Application of Simplex Designs in Optimization and Evolutionary Operation. Technometrics **4**(4) (1962) 441–461
12. Vincke, P.: Multicriteria Decision-Aid. John Wiley & Sons, New York (1992)
13. Ehrgott, M.: Multicriteria Optimization. 2nd edition edn. Springer, Berlin (2005)
14. Das, I., Dennis, J.E.: Normal-boundary intersection: a new method for generating Pareto optimal points in multicriteria optimization problems. SIAM Journal on Optimization **8**(3) (1998) 631–657
15. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation **11**(6) (2007) 712–731

16. McKinnon, K.I.M.: Convergence of the Nelder–Mead Simplex Method to a Nonstationary Point. SIAM Journal on Optimization **9**(1) (1998) 148–158
17. Rahman, M.K.: An intelligent moving object optimization algorithm for design problems with mixed variables, mixed constraints and multiple objectives. Structural and Multidisciplinary Optimization **32**(1) (2006) 40–58
18. Byatt, D.: A Convergent Variants of the Nelder-Mead Algorithm. Master's thesis, University of Canterbury (2000)
19. Trabia, M.B., Lu, X.B.: A Fuzzy Adaptive Simplex Search Optimization Algorithm. Journal of Mechanical Design **123** (2001) 216–225
20. Árpád Bűrmen, Puhan, J., Tuma, T.: Grid Restrained Nelder-Mead Algorithm. Computational Optimization and Applications **34** (2006) 359–375
21. Chelouah, R., Siarry, P.: Genetic and Nelder-Mead algorithms hybridized for a more accurate global optimization of continuous multiminima functions. European Journal of Operational Research **148**(2) (2003) 335–348
22. Lagarias, J.C., Reeds, J.A., Wright, M.H., Wright, P.E.: Convergence properties of the Nelder–Mead simplex method in low dimensions. SIAM Journal of Optimization **9** (1998) 112–147
23. Das, I.: Nonlinear Multicriteria Optimization and Robust Optimality. PhD thesis, Rice University, Houston, Texas (1997)
24. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability. Volume 1., University of California Press (1967) 281–297
25. Glover, F.: Tabu search and adaptive memory programing – Advances, applications and challenges. In: Interfaces in Computer Science and Operations Research, Kluwer Academic Publishers (1996) 1–75
26. Deb, K.: Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolutionary Computation **7**(3) (1999) 205–230
27. Murata, T., Ishibuchi, H.: MOGA: Multi-Objective Genetic Algorithms. In: Proceedings of the 2nd IEEE International Conference on Evolutionary Computing, Perth, Australia (1995) 289–294
28. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In Abraham, A., Jain, L., Goldberg, R., eds.: Evolutionary Multiobjective Optimization. Theoretical Advances and Applications. Springer, USA (2005) 105–145
29. Valenzuela-Rendón, M., Uresti-Charre, E.: A Non-Generational Genetic Algorithm for Multiobjective Optimization. In Bäck, T., ed.: Proceedings of the Seventh International Conference on Genetic Algorithms, San Mateo, California, Michigan State University, Morgan Kaufmann Publishers (1997) 658–665
30. Fonseca, C.M., Fleming, P.J.: 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, Sheffield, UK, IEE (1995) 42–52
31. Laumanns, M., Rudolph, G., Schwefel, H.P.: A Spatial Predator-Prey Approach to Multi-Objective Optimization: A Preliminary Study. In Eiben, A.E., Schoenauer, M., Schwefel, H.P., eds.: Parallel Problem Solving From Nature — PPSN V, Amsterdam, Holland, Springer-Verlag (1998) 241–249
32. Viennet, R., Fontiex, C., Marc, I.: Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set. International Journal of Systems Science **27**(2) (1996) 255–260

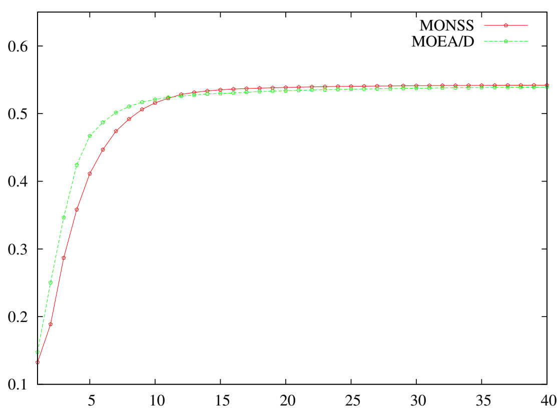
33. Lis, J., Eiben, A.E.: A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In Fukuda, T., Furuhashi, T., eds.: Proceedings of the 1996 International Conference on Evolutionary Computation, Nagoya, Japan, IEEE (1996) 59–64
34. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In Eiben, A.E., ed.: Parallel Problem Solving from Nature V, Amsterdam, Springer-Verlag (1998) 292–301
35. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G.: Performance Assessment of Multiobjective Optimizers: An Analysis and Review. IEEE Transactions on Evolutionary Computation **7**(2) (2003) 117–132
36. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation **8**(2) (2000) 173–195



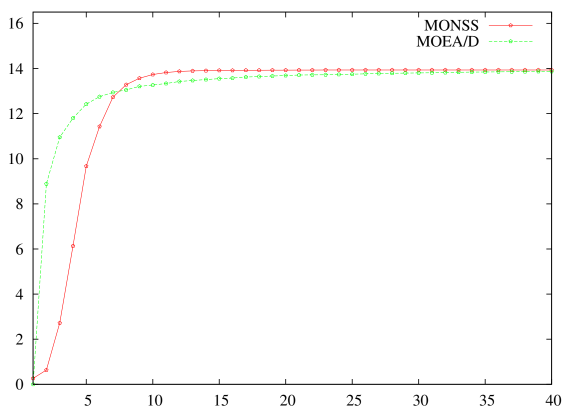
A) Hypervolume convergence for DEB2 problem



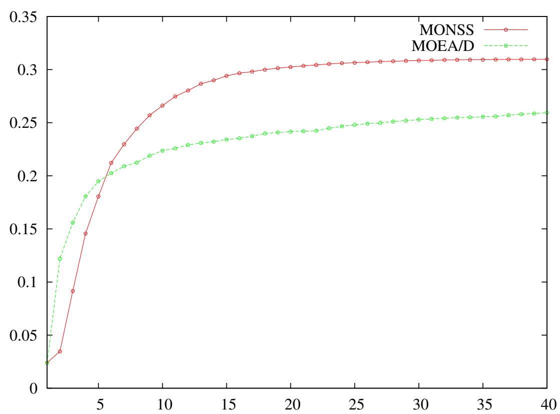
B) Hypervolume convergence for DTLZ5 problem



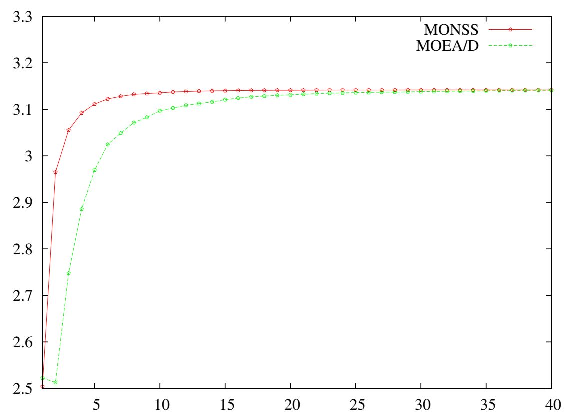
C) Hypervolume convergence for FON2 problem



D) Hypervolume convergence for LAU problem

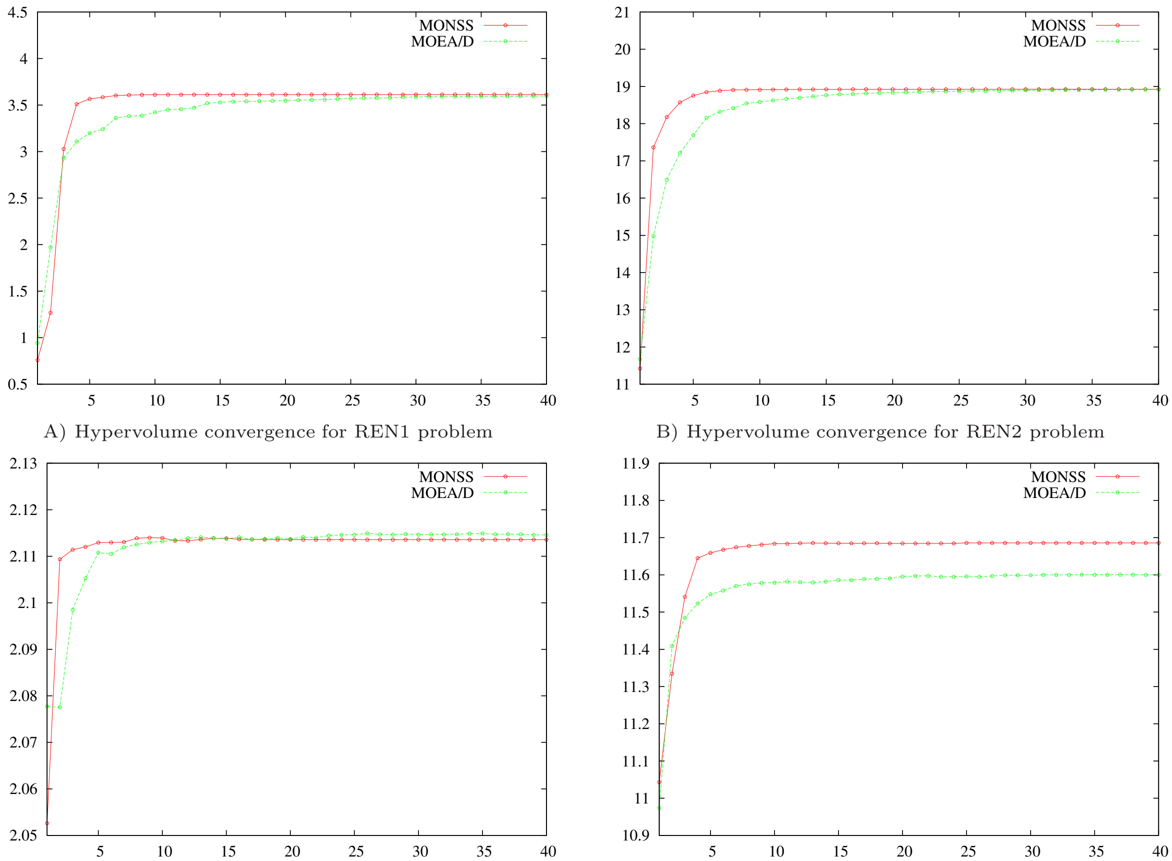


E) Hypervolume convergence for LIS problem



F) Hypervolume convergence for MUR problem

Fig. 4. Convergence graphic for MONSS and MOEA/D algorithms in DEB2, DTLZ5, FON2, LAU, LIS and MUR problems



A) Hypervolume convergence for REN1 problem

B) Hypervolume convergence for REN2 problem

C) Hypervolume convergence for VNT2 problem

D) Hypervolume convergence for VNT3 problem

Fig. 5. Convergence graphic for MONSS and MOEA/D algorithms in REN1, REN2, VNT2 and VNT3 problems