Noname manuscript No.
(will be inserted by the editor)

MONSS: A Multi-Objective Nonlinear Simplex Search Approach

Saúl Zapotecas-Martínez · Carlos A. Coello Coello

Received: date / Accepted: date

Abstract This paper present a novel methodology for dealing with unconstrained nonlinear multi-objective optimization problems (MOPs). The proposed algorithm adopts a nonlinear simplex search scheme in order to obtain multiple elements of the Pareto optimal set. The search is directed by a well-distributed set of weight vectors, which define each, a scalarization problem, that is solved by deforming a simplex according to the movements described by Nelder and Mead's method. 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 weight 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 paper, 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, Multi-objective nonlinear simplex search, Multi-objective direct search methods

1 Introduction

In engineering and scientific applications, it is common to find problems having two or more (often conflicting) objective functions. They are called Multi-objective Optimization Problems (MOPs) and, because of the conflict among their objectives, it is normally the case that they have more than one solution. Pareto optimality is the most commonly adopted concept to determine the solutions of an MOP. When

The second author gratefully acknowledges support from CONACyT project no. 103570.

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

applying this concept, a set of solutions called *Pareto optimal set* (PS) is produced. The solutions contained in this set represent the best possible trade-offs among the objectives of the MOP being solved (i.e., PS contains solutions in which no objective can be improved without worsening another one).

Over the years, several mathematical programming techniques for dealing with MOPs have been proposed. Most of these methods transform a MOP into a single-objective scalarization function, in which the objective to be optimized consists of the aggregation of all the (original) objective functions f_i's (these are the well-known scalarization approaches). Once the aggregating function is formulated, a mathematical programming method is employed for finding a Pareto optimal solution. These mathematical methods have shown to be effective tools in many domains, at a reasonably low computational cost. However, they have several limitations, including the fact that they obtain a single Pareto optimal 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) [6], which, because of their simplicity and ease of use, have become very popular in a wide variety of domains [5].

The development of mathematical programming techniques for solving multi-objective problems has been a very active area of research for many years, giving rise to a wide variety of approaches [10, 23, 24, 34]. Recently, several powerful approaches that rely on gradient information, have been proposed. For example, Fliege et. al [12] proposed an extension of Newton's method for unconstrained multi-objective optimization. Fischer and Shukla [11] presented an algorithm based on the Levenberg-Marquardt method to solve unconstrained MOPs. Unfortunately, when the gradient information of the objectives is not available, such methods become impractical and it is necessary to look for alternative approaches—for example, direct search methods.

The use of direct search methods has been scarce in the multi-objective context, although some researchers have used them as local search operators coupled to MOEAs—see for example [17, 36, 37]. Nevertheless, the existence of multi-objective direct search methods to approximate multiple solutions to the Pareto optimal set (maintaining a good distribution of solutions along the Pareto optimal front) using direct search methods that are not based on metaheuristics, have been rare in the specialized literature—see for example [1]. The main reason for the shortage in such strategies, is that it is not efficient to approximate different solutions along the Pareto optimal front using mathematical programming techniques. 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 [16], and such is precisely the focus of the work reported here.

In this paper, we present a new method for dealing with unconstrained MOPs based on a direct search method. The proposed approach analyzes and exploits the properties of Nelder and Mead's method [26] (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 the search by means of movements guided by mathematical programming tech-

niques, while maintaining a reasonably good representation of the Pareto optimal front. The study presented here indicates that the proposed approach is computationally efficient (in terms of the number of objective function evaluations that it performs) and produces competitive results when dealing with MOPs of low and moderate dimensionality. As will be seen later on, our proposed approach produces competitive results with respect to those of a state-of-the-art MOEA.

2 Basic Concepts

2.1 Multi-objective Optimization

Assuming minimization, a continuous multi-objective optimization problem (MOP), can be stated as:

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

where $\Omega \subset \mathbb{R}^n$ defines the decision space and $\mathbf{F}$ is defined as the vector of the objective functions:

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

where each $f_j : \Omega \to \mathbb{R}, j = 1, \ldots, k$ represents the function to be minimized. In this paper we consider the box-constrained case, i.e., $\Omega = \prod_{i=1}^{n}[a_j, b_j]$. Therefore, each variable vector $\mathbf{x} = (x_1, \ldots, x_n)^T \in \Omega$ is such that $a_i < x_i < b_i, i = 1, \ldots, n$.

In multi-objective optimization, it is desirable to obtain 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 [23].

Definition 1 Let $\mathbf{x}, \mathbf{y} \in \Omega$, we say that $\mathbf{x}$ *dominates* $\mathbf{y}$ (denoted by $\mathbf{x} \prec \mathbf{y}$) with respect to eq. (1) if and only if:

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

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

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

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

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

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

As in most of multi-objective algorithms, we are interested in finding the maximum number of elements of the Pareto optimal set, while maintaining a well-distributed set of solutions along the Pareto front.

2.2 Decomposing a Multi-objective Optimization Problem

It is well-known that a Pareto optimal solution to the problem defined in eq. (1), under certain conditions, could be an optimal solution of a scalar optimization problem in which the objective is an aggregation of all the functions f_i's. Many scalar approaches have been proposed to aggregate the objectives of a MOP. Among them, the Tchebycheff approach is one of most widely used methods reported in the specialized literature. In the following, we describe the Tchebycheff problem which is adopted in this paper. Note however, that other scalarization approaches could also be easily coupled to this work—see for example those presented in [10, 23, 34].

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

$$\text{Minimize} \quad g(\mathbf{x}|\mathbf{w},\mathbf{z}) = \max_{1\leq j\leq k}\{w_j|f_j(\mathbf{x}) - z_j|\} \tag{2}$$

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

For each Pareto optimal point $\mathbf{x}^\star$ there exists a weighting vector $\mathbf{w}$ such that $\mathbf{x}^\star$ is the optimum solution of eq. (2) and each optimal solution of eq. (2) is a Pareto optimal solution of eq. (1). An appropriate representation of the Pareto front could be reached by solving different scalarization problems. Such problems can be defined by a set of well-distributed weight vectors, which define the search direction in the optimization process. This strategy is employed in this work, and its form of use will be described in Section 4.2.

3 The Nonlinear Simplex Search

Nelder and Mead's method [26] also known as the *Nonlinear Simplex Search* (NSS) or *Downhill Simplex Search*, is an algorithm based on the simplex algorithm of Spendley et al. [29], which was introduced for minimizing continuous and multidimensional unconstrained optimization problems. 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. The following definitions are of relevance in this work [15].

Definition 5 Let $\mathbf{\Delta} = \{\Delta_1,\ldots,\Delta_{n+1}\}$ be the set of $n+1$ points in $\mathbb{R}^n$, such that $\Delta_2 - \Delta_1,\ldots,\Delta_{n+1} - \Delta_1$ are linearly independent. Then, the simplex or n-simplex with vertices $\mathbf{\Delta}$ is defined by the convex hull of $\mathbf{\Delta}$, i.e., the set of points:

$$\mathbf{\Delta}^n = \left\{ t_1\Delta_1 + \cdots + t_{n+1}\Delta_{n+1} : \sum_{i=1}^{n+1} t_i = 1 \text{ and } t_i \geq 0 \text{ for all } i \in \{1,\ldots,n+1\} \right\}$$

If the vertices of the simplex are all mutually equidistant, then the simplex is said to be regular. Thus, in two dimensions, a regular simplex is an equilateral triangle, while in three dimensions a regular simplex is a regular tetrahedron.

Definition 6 A simplex is called *nondegenerate*, if and only if, the n vectors: $\Delta_2 - \Delta_1, \ldots, \Delta_{n+1} - \Delta_1$ are linearly independent. Otherwise, the simplex is called *degenerate*, and then, the simplex will be defined in a lower dimension than n.

The convergence towards a minimum value at each iteration of Nelder and Mead's method is conducted by four main movements in a nondegenerate simplex. The method is fully defined stating four scalar parameters to control the movements performed in the simplex: **reflection** (ρ), **expansion** (χ), **contraction** (γ) and **shrinkage** (σ). According to Nelder and Mead, these parameters should satisfy:

$$\rho > 0, \quad \chi > 1, \quad \chi > \rho, \quad 0 < \gamma < 1 \quad \text{and} \quad 0 < \sigma < 1 \tag{3}$$

Actually, there is no method that can be used to establish this set of parameters. However, the nearly universal choices used in Nelder and Mead's method are [26]:

$$\rho = 1, \quad \chi = 2, \quad \gamma = \frac{1}{2} \quad \text{and} \quad \sigma = \frac{1}{2} \tag{4}$$

Let $\mathbf{\Delta} = \{\mathbf{\Delta}_1, \mathbf{\Delta}_2, \ldots, \mathbf{\Delta}_{n+1}\}$ be the set of vertices that define a nondegenerate simplex, such that the vertices are sorted according to the function value, i.e., $f(\mathbf{\Delta}_1) < \cdots < f(\mathbf{\Delta}_{n+1})$. Then, the transformations performed into the simplex by the Nelder and Mead method, are defined as:

1. Reflection: $\mathbf{x}_r = (1+\rho)\mathbf{x}_c - \rho\mathbf{\Delta}_{n+1}$, see Fig. 1.
2. Expansion: $\mathbf{x}_e = (1+\rho\gamma)\mathbf{x}_c - \rho\chi\mathbf{\Delta}_{n+1}$, see Fig. 1.
3. Contraction:
   (a) *Outside*: $\mathbf{x}_{oc} = (1+\rho\gamma)\mathbf{x}_c - \rho\gamma\mathbf{\Delta}_{n+1}$, see Fig. 1.
   (b) *Inside*: $\mathbf{x}_{ic} = (1-\gamma)\mathbf{x}_c + \gamma\mathbf{\Delta}_{n+1}$, see Fig. 1.
4. Shrinkage: Each vertex of the simplex is transformed by the geometric shrinkage defined by: $\mathbf{\Delta}_i = \mathbf{\Delta}_1 + \sigma(\mathbf{\Delta}_i - \mathbf{\Delta}_1)$, $i = 2, \ldots, n+1$, and the new vertices are evaluated, see Fig. 1.

where $\mathbf{x}_c = \frac{1}{n}\sum_{i=1}^{n} \mathbf{\Delta}_i$ is the centroid of the n best points (all vertices except for $\mathbf{\Delta}_{n+1}$), $\mathbf{\Delta}_1$ and $\mathbf{\Delta}_{n+1}$ are the best and the worst solutions identified within the simplex, respectively.

At each iteration, the simplex is modified by one of the above movements, according to the following rules:

1. If $f(\mathbf{\Delta}_1) \leq f(\mathbf{x}_r) \leq f(\mathbf{\Delta}_n)$, then $\mathbf{\Delta}_{n+1} = \mathbf{x}_r$.
2. If $f(\mathbf{x}_e) < f(\mathbf{x}_r) < f(\mathbf{\Delta}_1)$, then $\mathbf{\Delta}_{n+1} = \mathbf{x}_e$, otherwise $\mathbf{\Delta}_{n+1} = \mathbf{x}_r$.
3. If $f(\mathbf{\Delta}_n) \leq f(\mathbf{x}_r) < f(\mathbf{\Delta}_{n+1})$ and $f(\mathbf{x}_{oc}) \leq f(\mathbf{x}_r)$, then $\mathbf{\Delta}_{n+1} = \mathbf{x}_{oc}$; otherwise, perform a shrinkage.
4. If $f(\mathbf{x}_r) \geq f(\mathbf{\Delta}_{n+1})$ and $f(\mathbf{x}_{ic}) < f(\mathbf{\Delta}_{n+1})$, then $\mathbf{\Delta}_{n+1} = \mathbf{x}_{ic}$; otherwise, perform a shrinkage.

The stopping criterion employed by Nelder and Mead, and commonly adopted in many optimization problems is defined by:

$$\sqrt{\frac{1}{n+1}\sum_{i=1}^{n+1}\left(f(\mathbf{\Delta}_i) - \overline{f}\right)^2} \leq \varepsilon \tag{5}$$

where $\overline{f} = \frac{1}{n+1}\sum_{i=1}^{n+1} f(\mathbf{\Delta}_i)$ and ε is a predefined constant.

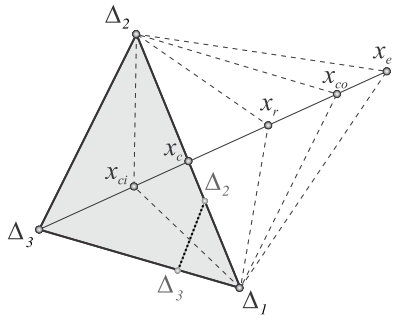


Fig. 1 Illustration of the possible movements in the simplex performed by the NSS method. The constructed simplex corresponds to an optimization problem with two decision variables, where Δ_1 and Δ_3 are the best and the worst points, respectively.

4 The Nonlinear Simplex Search for Multi-Objective Optimization

4.1 About Nonlinear Simplex Search and Multi-objective Optimization Problems

Traditional mathematical programming techniques for solving optimization problems have shown to be an effective tool in many science and engineering fields. Most of these strategies have been designed to deal with convex functions and usually require the gradient of the function. Being a direct search method, Nelder and Mead's method has the advantage of not requiring gradient information. Instead, the NSS algorithm aims at obtaining a better solution by deforming a simplex shape along the search process. Nonetheless, Nelder and Mead's method has an important 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 [22]). For this family of functions and others having similar features, a more appropriate strategy needs to be adopted (e.g., adjusting the control parameters, modifying the movements into the simplex, etc.). In recent years, several attempts to improve the simplex search algorithm have been reported in the specialized literature—see e.g. [2, 3, 27, 30, 31]. However, due to its inherent nature (based on heuristic movements), several of these variants of the simplex search normally produce additional problems, and, in some cases, they eventually fail.

In addition to any changes to the NSS algorithm itself, it is also possible to propose different strategies for constructing the simplex, and several researchers have reported work in that direction—see e.g. [4, 36]. The construction of the simplex plays an important role in the performance of the NSS algorithm. For example, to use a degenerated simplex (i.e., a simplex defined in lower dimensionality than the number of decision variables) in the minimization process, is inappropriate. The reason is that in such case, the search is restricted to find an optimal solution in lower dimension, which avoids achieving this optimal solution if it is not located in the same dimension of the optimization problem [30].

In most real-world MOPs, the features of the PS are unknown. However, under some conditions, it can be induced from the Karush-Kuhn-Tucker conditions that the PS of a continuous MOP with k objectives forms a $(k-1)$-dimensional piecewise continuous manifold in decision variable space [23, 28]. This regularity

property of continuous MOPs could be exploited in the search. In our case, the search is directed by a well-distributed set of weight vectors, each of which defines a scalarization problem by using the Tchebycheff approach. Since the problem defined in eq. (2) is continuous of $\mathbf{w}$, an optimal solution of the problem defined by a weight vector $\mathbf{w}^i$ should be close to the one defined by other weight vector $\mathbf{w}^j$ $(i \neq j)$, if $\mathbf{w}^i$ and $\mathbf{w}^j$ are close to each other. Therefore, assuming that each subproblem is solved throughout the search, any information of the solution that minimizes contiguous problems could be useful in the search. Thus, the vertices of the simplex could be constructed by using solutions that minimize contiguous problems.

The stopping criterion employed by Nelder and Mead (eq. (5)), has the advantage of linking the size of the simplex with an approximation of the local curvature of the objective function. In fact, a local minimum is often obtained for a high curvature of the objective function. Conversely, a minimum located in a flat valley of the objective function carries less information. Therefore, it does not make much sense to transform the simplex which would be almost flat. However, satisfying the stopping criterion of eq. (5) could require several transformations of the simplex, which in practice, could consume a considerable number of objective function evaluations and the simplex could eventually collapse in a single direction, i.e., in a unique region of the PS. Lagarias et al. [18] showed that the convergence towards a better point in the Nelder and Mead method is achieved at most in $n+1$ iterations (at least in convex problems with low dimensionality). Therefore, this property of convergence could be exploited in order to design an appropriate strategy for approximating solutions to the PS.

In addition, the execution of the shrinkage step in the NSS algorithm could become inefficient. This can be mainly caused by two facts:

1. Once the simplex is transformed by the shrinkage step, n vertices are generated and all of them need to be evaluated. Thus, when the dimension of the MOP is high, the number of objective function evaluations will significantly increase.
2. Since the shrinkage step reduces the simplex volume, the search is then restricted to a small portion of the search space. Therefore, the risk of collapsing the simplex in a specific region of the search space increases. Consequently, the diversity of solutions along the Pareto front tends to reduce, which is a disadvantage for the decision maker in multi-objective optimization.

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

4.2 The Multi-Objective Nonlinear Simplex Search

Our proposed *Multi-objective Nonlinear Simplex Search* (MONSS) decomposes a MOP into several single-objective scalarization subproblems. In the following description, we use the Tchebycheff approach, although any other scalarization approach can serve for the same purpose.

Let $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ be a well-distributed set of weight vectors. The problem to approximate solutions to the Pareto front of the problem defined by eq. (1), can be treated by solving a set of N scalar optimization subproblems in which

problem (1) is decomposed. Therefore, (assuming the Tchebycheff approach) the i^{th} optimization problem consists in minimizing:

$$g(\mathbf{x}|\mathbf{w}^i, \mathbf{z}) = \max_{1\leq j\leq k}\{w_j^i|f_j(\mathbf{x}) - z_j|\} \tag{6}$$

where $\mathbf{w}^i = (w_1^i, \ldots, w_k^i)^T$ is a weight vector, i.e., $w_j^i \geq 0$ for all $j \in \{1, \ldots, k\}$ and $\sum_{j=1}^k w_j^i = 1$. $\mathbf{z} = (z_1, \ldots, z_k)^T$ is the reference vector whose component z_j is the minimum value found so far for the objective f_j for all $j \in \{1, \ldots, k\}$.

At each iteration, MONSS minimizes simultaneously these N optimization subproblems. Therefore, MONSS generates a solution $\mathbf{x}^i$ that minimizes $g(\mathbf{x}^i|\mathbf{w}^i, \mathbf{z}^\star)$ Initially, a set of N solutions $\mathcal{S} = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ having an uniform distribution is randomly initialized. Each vector $\mathbf{x}^i \in \mathcal{S}$ represents a solution for the i^{th} subproblem defined by the i^{th} weight vector $\mathbf{w}^i \in W$. In this way, different subproblems are simultaneously solved by the MONSS algorithm and the set of solutions $\mathcal{S}$ will constitute an approximation to the PS 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 weight 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:

$$\bigcap_{i=1}^{m} C_i = \emptyset \text{ and } \bigcup_{i=1}^{m} C_i = W \tag{7}$$

and all the weight vectors $\mathbf{w}^c \in C_i$ are contiguous among themselves.

The simplex search is focused on minimizing a subproblem defined by a weight vector $\mathbf{w}^s$ which is randomly chosen from each partition C_i. The n-simplex ($\boldsymbol{\Delta}$) used in the search, is defined by the vertices:

$$\boldsymbol{\Delta} = \{\mathbf{x}^s, \mathbf{x}^1, \ldots, \mathbf{x}^n\} \tag{8}$$

such that: $\mathbf{x}^s$ is the solution in $\mathcal{S}$ with minimum value for the problem $g(\mathbf{x}^s|\mathbf{w}^s, \mathbf{z})$. The remaining vertices of the simplex $\mathbf{x}^j$ ($\mathbf{x}^s \neq \mathbf{x}^j$, for all $j \in \{1, \ldots, n\}$) are chosen by selecting n different solutions from $\mathcal{S}$, such that minimize the subproblems defined by the closest n weight vectors to $\mathbf{w}^s$ (in terms of the Euclidean distance), where n represents the number of decision variables of the MOP, see Fig. 2.

The movements in the simplex are carried out according to Section 3. However, considering the discussion in Section 4.1, MONSS omits the shrinkage step in the search. In the NSS algorithm, it is common that after any movement the new solution $\mathbf{x}^{new} = (x_1^{new}, \ldots, x_n^{new})^T$ leaves the allowed search space Ω. In order to deal with this problem, (as in [36]) we deterministically bias the boundaries. Therefore, the j^{th} component of the new solution $\mathbf{x}^{new}$ is re-established as follows:

$$x_j^{new} = \begin{cases} x_j^{lb}, & \text{if } x_j^{new} < x_j^{lb} \\ x_j^{ub}, & \text{if } x_j^{new} > x_j^{up} \end{cases} \tag{9}$$

where x_j^{lb} and x_j^{ub} denote the lower and upper bounds for the j^{th} component of the search space, for $j = 1, \ldots, n$. In fact, for each movement performed by the NSS algorithm, a new nondegenerate simplex is generated [18]. Therefore, the strategy adopted to re-establish the new solution could degenerate the simplex. However,

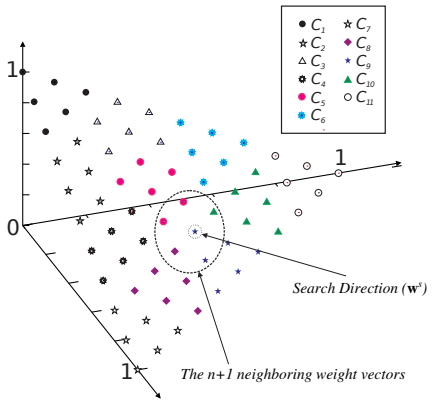


Fig. 2 Illustration of a well-distributed set of weight vectors for a MOP with three objectives, five decision variables and 66 weight vectors, i.e., $m = \frac{|W|}{n+1} = 11$ partitions. The n-simplex is constructed by six solutions that minimize the subproblems defined by six neighboring weight vectors contained in four different partitions (C_5, C_8, C_9 and C_{10}). The search is focused on the direction defined by the weight vector $\mathbf{w}^s$.

in practice, it could achieve optimal solutions (at least) in the dimension in which the simplex is defined [30].

In order to speed up the approximation to the Pareto front, the search direction $\mathbf{w}^s$ is changed (at each movement of the NSS) for any other direction $\hat{\mathbf{w}} \in C_i$. Note that the problem by eq. (2) is continuous of $\mathbf{w}$, therefore, the optimal solution of $g(\mathbf{x}|\mathbf{w}^i, \mathbf{z})$ should be close to that of $g(\mathbf{x}|\mathbf{w}^j, \mathbf{z})$ if $\mathbf{w}^i$ and $\mathbf{w}^j$ are close to each other. Since the partition C_i has contiguous weight vectors, any information about the problems defined with weight vectors $\mathbf{w}^c \in C_i$ should be helpful for the search.

In this way, an agile search into the partition C_i is performed and a collapse of the simplex search in the same direction $\mathbf{w}^s$ is avoided. Additionally, the number of movements in the simplex is limited to $n+1$. This criterion is taken according to the discussion presented in Section 4.1. Here, we define $m = \frac{|W|}{n+1}$ partitions of the set W, guaranteeing at least $n+1$ movements in the simplex for each partition. Such partitions can be easily constructed by using a naive modification of the well-known k-means algorithm [21].

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 approach 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 weight set W, such as it is shown in Algorithm 1.

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 Tchebycheff approach, simulating the Path Relinking method [14]. Summarizing, the MONSS algorithm can be stated as shown in Algorithm 2.

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

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

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

Input:
$W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$: A set of N weight vectors.
T_{max}: A maximum number of iterations.
Output:
$\mathcal{S}$: An approximation to the Pareto front.

begin
  $t = 0$;
  Generate initial solutions: Generate a set $\mathcal{S}^t = \{\mathbf{x}^1, \ldots, \mathbf{x}^N\}$ 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: eq. (7) is satisfied;
  while $t < T_{max}$ **do**
    for $i = 0$ *to* m **do**
      Randomly choose $\mathbf{w}^s \in C_i$;
      Apply Simplex Search method:
      a) **Build the** n**-simplex:** Construct the n-simplex from $\mathcal{S}^t$, such that: eq. (8) is satisfied.
      b) **Apply the NSS method:** Execute the NSS method during $n+1$ iterations. At each iteration:
        * Repair the bounds according to eq. (9).
        * Relax the search changing the search direction $\mathbf{w}^s$ for any other $\hat{\mathbf{w}} \in C_i$.
        * 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

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

5 Experimental Study

To assess the performance of our proposed MONSS algorithm, we compare its results with respect to those obtained by the *Multi-Objective Evolutionary Algorithm*

based on Decomposition (MOEA/D) [38]. MOEA/D is a state-of-the-art MOEA that has shown its superiority with respect to older state-of-the-art MOEAs such as NSGA-II [8] and SPEA2 [41]—see [38]. In our study, we only compare our proposed approach with respect to MOEA/D for two main reasons: 1) MOEA/D is an algorithm that decomposes the MOP into several single-objective optimization problems (as or proposed approach); and 2) the proposed approach does not pretend to be an evolutionary algorithm, instead of this, we study the use of Nelder and Mead's algorithm to solve MOPs with different characteristics. Therefore, a more detailed discussion can be carried out by focusing in analyzing the results achieved by our MONSS algorithm. In then next section, we present the test problems and the performance assessment measures adopted in our comparative study.

5.1 Test Problems and Performance Assessment

In our experiments, we adopted 20 MOPs with two and three objectives having different characteristics in their Pareto optimal fronts such as: convexity, concavity and discontinuity. Ten of these MOPs have been taken from different authors and the definitions of such MOPs are summarized in Table 1 (in this work, we call to all of them *traditional test problems*). For the sake of a more comprehensive study, we also adopt the ten MOPs with complicate Pareto optimal sets proposed by Zhang et al. [39], which constitute the well-known CEC'2009 test suite. Due to space limitations, the description of these problems is omitted. However, the interested reader is referred to [39] for the details about such test problems.

In order to assess the performance of our proposed approach on the test problems adopted, we compare it with respect to MOEA/D using the following performance measures:

Hypervolume (I_H): The I_H performance measure was proposed by [42]. This performance measure is Pareto compliant [43], and quantifies both convergence and spread of nondominated solutions along the Pareto optimal front. The hypervolume corresponds to the non-overlapped volume of all the hypercubes formed by a reference point $\mathbf{r}$ (given by the user) and each solution $\mathbf{p}$ in the Pareto set approximation (P). It is mathematically stated as:

$$I_H(P) = \Lambda\left(\bigcup_{\mathbf{p}\in P}\{\mathbf{x}|\mathbf{p}\prec\mathbf{x}\prec\mathbf{r}\}\right) \tag{10}$$

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

Two Set Coverage (I_C): The I_C performance measure was proposed by Zitzler et al. [40], 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:

$$I_C(A,B) = \frac{|\{\mathbf{b}\in B|\exists\mathbf{a}\in A:\mathbf{a}\preceq\mathbf{b}\}|}{|B|} \tag{11}$$

Table 1 Traditional test problems

MOP	Definition	MOP	Definition
DEB2 [7]	$f_1(\mathbf{x}) = x_1$ $f_2(\mathbf{x}, g(\mathbf{x})) = g(\mathbf{x}), \cdot h(\mathbf{x})$ and: $g(\mathbf{x}) = 1 + 10x_2$ $h(\mathbf{x}) = 1 - (f_1(\mathbf{x})/g(\mathbf{x}))^2 - \frac{f_1(\mathbf{x})}{g(\mathbf{x})} \times \sin(12\pi f_1(\mathbf{x}))$ $x_i \in [0, 1]$	MUR [25]	$f_1(\mathbf{x}) = 2\sqrt{x_1}$ $f_2(\mathbf{x}) = x_1(1 + x_2) + 5$ $x_1 \in [1, 4], x_2 \in [1, 2]$
DTLZ5 [9]	$f_1(\mathbf{x}) = \cos(\theta_1)\cos(\theta_2)h(\mathbf{x})$ $f_2(\mathbf{x}) = \cos\theta_1 \sin(\theta_2)h(\mathbf{x})$ $f_3(\mathbf{x}) = \sin(\frac{\pi}{2}x_1^\alpha)h(\mathbf{x})$ $g(\mathbf{x}) = \sum_{i=3}^{n}(x_i - 0.5)^2$ $h(\mathbf{x}) = (1 + g(\mathbf{x}))$ $\alpha = \pi$ $x_i \in [0, 1], n = 12$	REN1 [32]	$f_1(\mathbf{x}) = \frac{1}{x_1^2+x_2^2+1}$ $f_2(\mathbf{x}) = x_1^2 + 3x_2^2 + 1$ $x_i \in [-3, 3]$
FON2 [13]	$f_1(\mathbf{x}) = 1 - \exp(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2)$ $f_2(\mathbf{x}) = 1 - \exp(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2)$ $x_i \in [-4, 4], n = 3$	REN2 [32]	$f_1(\mathbf{x}) = x_1 + x_2 + 1$ $f_2(\mathbf{x}) = x_1^2 + 2x_2^2 - 1$ $x_i \in [-3, 3]$
LAU [19]	$f_1(\mathbf{x}) = x_1^2 + x_2^2$ $f_2(\mathbf{x}) = (x_1 + 2)^2 - x_2^2$ $x_i \in [-50, 50]$	VNT2 [33]	$f_1(\mathbf{x}) = \frac{(x_1-2)^2}{2} + \frac{(x_2+1)^2}{13} + 3$ $f_2(\mathbf{x}) = \frac{(x_1+x_2-3)^2}{36} + \frac{(-x_1+x_2+2)^2}{8} - 17$ $f_3(\mathbf{x}) = \frac{(x_1+2x_2-1)^2}{175} + \frac{(2x_2-x_1)^2}{17} - 13$ $x_i \in [-4, 4]$
LIS [20]	$f_1(\mathbf{x}) = \sqrt[8]{x_1^2 + x_2^2}$ $f_2(\mathbf{x}) = \sqrt[4]{(x_1 - 0.5)^2 + (x_2 - 0.5)^2}$ $x_i \in [-5, 10]$	VNT3 [33]	$f_1(\mathbf{x}) = 0.5(x_1^2 + x_2^2) + \sin(x_1^2 + x_2^2)$ $f_2(\mathbf{x}) = \frac{(3x_1-2x_2+4)^2}{8} + \frac{(x_1-x_2+1)^2}{27} + 15$ $f_3(\mathbf{x}) = \frac{1}{(x_1^2+x_2^2+1)} - 1.1\exp(-x_1^2 - x_2^2)$ $x_i \in [-3, 3]$

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

5.2 Experimental Setup

As indicated before, we compared the results obtained by our proposed approach with respect to those achieved by MOEA/D [38]. For a fair comparison, we used the Tchebycheff approach and the same weight vectors for both algorithms. The weight vectors were generated as in [38], i.e., the settings of N and $W = \{\mathbf{w}^1, \ldots, \mathbf{w}^N\}$ are controlled by a parameter H. More precisely, $\mathbf{w}^1, \ldots, \mathbf{w}^N$ are all the weight vectors in which each individual weight w_j^i ($i = 1, \ldots, N$ and $j = 1, \ldots, k$) takes a value from:

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

Therefore, the number of such vectors in W is given by $N = C_{H+k-1}^{k-1}$, where k is the number of objective functions. Here, we use $H = 99$ (for two-objective

Table 2 Parameters for MONSS and MOEA/D

Parameter	MONSS	MOEA/D
N_{sol}	100/300	100/300
T_{max}	40/200	40/200
T_n	–	30
P_c	–	1
P_m	–	$1/n$
ρ	1	–
χ	2	–
γ	1/2	–

problems) and $H = 23$ (for three-objective problems), i.e., 100 and 300 weight vectors for MOPs having two and three objectives, respectively,

For each MOP, 30 independent runs were performed with each algorithm. The parameters for both algorithms are summarized in Table 2, where N_{sol} represents the number of initial solutions (100 for two-objective problems and 300 for three-objective problems). T_{max} represents the maximum number of iterations. For traditional test problems (i.e., the problems in Table 1) we used $T_{max} = 40$, while for the CEC'2009 test problems (denoted as UF1-UF10 in Table 3) we adopted $T_{max} = 200$. Therefore, both algorithms performed 4,000 (for traditional test problems) and 20,000 (for the CEC'2009 test problems) objective function evaluations, in problems having two objectives. In the case of the three-objective problems, both algorithms performed 12,000 (for traditional test problems) and 60,000 (for the CEC'2009 test problems) objective function evaluations. The CEC'2009 test suite was tested using six decision variables.

For MONSS, ρ, χ and γ represent the control parameters for the reflection, expansion and contraction movements, 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.

For each MOP, the algorithms were evaluated using the I_H and I_C performance measures. The results obtained are summarized in Tables 3 and 4. These tables display both the *average* and the standard deviation (σ) of the I_H and I_C performance measures for each MOP. The reference vectors used for computing the I_H performance measure are shown in Table 3. 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 distribution is reported when the algorithms converge along the Pareto front. In the case of the statistics for the I_C metric comparing pairs of algorithms (i.e., $I_C(A, B)$ and $I_C(B, A)$), 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.

In order to determine if there were significant differences among the results obtained by the algorithms (MONSS and MOEA/D), we adopted the Wilcoxon rank sum [35]. This null hypothesis test computes the probability that given two independent samples A and B, they come from distributions that have *equal medians*. The two samples of data are assumed to come from continuous distributions that are identical except possibly for a location shift, but are otherwise arbitrary.

In Table 3, P is the probability of rejecting the null hypothesis: *"medians are equal"*. Therefore, small values of P cast doubt on the validity of the null hypothesis. In our study, we performed the Wilcoxon rank sum using a P-value=0.05 as significance level. H=0 indicates that the null hypothesis cannot be rejected at the 5% level. H=1 indicates that the null hypothesis can be rejected at the 5% level.

5.3 Discussion of results

The main goal of the simulations performed in this study, is to verify the effectiveness of our proposed MONSS when dealing with MOPs with different characteristics. As indicated before, the results obtained by our proposed approach were compared against those produced by MOEA/D. Tables 3 and 4 show the results obtained by the algorithms in both the traditional and the CEC'2009 test problems, for the I_H and I_C performance measures. These results are separately discussed in the next sections for a better interpretation.

5.3.1 Traditional test problems

In Table 3, it can be seen that the results obtained by our proposed MONSS outperformed those achieved by MOEA/D in most of the traditional test problems (i.e., the MOPs described in Table 1). This table provides a quantitative assessment of the performance of MONSS in terms of the I_H performance measure. This means that the solutions obtained by MONSS achieved a better approximation and spread of solutions along the Pareto front than those produced by MOEA/D. The exception was DTLZ5 where MOEA/D obtained a better I_H value. However, according to Wilcoxon's rank sum, MOEA/D was not significant better than MONSS for this problem, i.e., the performance of both algorithms was very similar. For LAU, LIS, REN2 and VNT2 test problems, the null hypothesis test showed significant differences among the samples produced by MONSS and MOEA/D. That means (in addition with the I_H values) that the performance of our proposed approach for these problems was significantly better than the one achieved by MOEA/D. Although for DEB2, FON2, MUR, REN1 and VNT3, the I_H value indicates a better performance for MONSS, Wilcoxon's rank sum did not show significant differences, i.e., MONSS was not significantly better than MOEA/D. However, MONNS was very competitive for these problems. To validate our results, Fig. 3 shows the I_H convergence plots for these problems. In this figure, we can see the performance between MONSS and MOEA/D along each iteration. From these plots, it is possible to observe that in most cases, MONSS quickly achieves a better value for the I_H performance measure.

Regarding the I_C performance measure (Table 4), our proposed MONSS obtained better results when compared against those produced by MOEA/D in most of these test problems. This means that the solutions obtained by MONSS dominated a higher ratio of solutions produced by MOEA/D. However, MOEA/D obtained a better performance in terms of the I_C performance measure for DTLZ5. However, as can be seen in Table 4, the ratio of solutions dominated by MOEA/D was not significantly high. The use of the I_C performance measure also corroborates the results obtained by the I_H performance measure and the Wilcoxon rank

sum in the DTLZ5 test problem, i.e., the performance of both algorithms was very similar.

5.3.2 CEC'2009 test problems

Regarding the CEC'2009 test problems (UF1-UF10 in Tables 3 and 4), we can see that our proposed MONSS obtained a better value in the I_H performance measure in most of the CEC'2009 test problems. The exceptions were the problems UF5 and UF6, where MOEA/D obtained a better value in the I_H performance measure. The Wilcoxon rank sum showed significant differences among the samples produced by MONSS and MOEA/D in the majority of these problems. The exception was UF10, where the null hypothesis test did not show significant difference. Therefore, although MONSS obtained a better I_H value for this problem, it was not significantly better than MOEA/D. However, for UF1-UF4, UF7 and UF9, according to Wilcoxon's rank sum, our proposed MONSS obtained significantly better results than those achieved by MOEA/D. That means that a better approximation and spread of solutions along the Pareto front were obtained by MONSS in comparison with those achieved by MOEA/D. In fact, for UF5 and UF6, MONSS was significantly outperformed by MOEA/D. The main reason for the poor performance of MONSS in these problems is attributed to the difficult landscapes that UF5 and UF6 possess—see [39]. Fig. 4 shows the I_H converge plots for the CEC'2009 test problems. This figure shows the performance between MONSS and MOEA/D along the iterations. In fact, the poor performance of MONSS for UF5 and UF6 can be seen in Fig. 4 E) and F), respectively.

With respect to the I_C performance measure, MONSS obtained better results in comparison with those achieved by MOEA/D in most of these test problems. Therefore, the solution obtained by MONSS dominated a higher ratio of solutions produced by MOEA/D. The exception were UF5 and UF6, in which MOEA/D produced more solution that dominate to those obtained by MONSS. This also corroborates the results obtained for the indicator I_H in UF5 and UF6, where MONSS was significantly outperformed by MOEA/D. However, in spite of being an algorithm based on a direct search method, MONSS became very competitive in comparison with a powerful state-of-the-art MOEA (i.e. MOEA/D), and in most cases, the performance of MONSS significantly outperformed the one achieved by MOEA/D.

6 Conclusions and Future Work

We have proposed a new method based on the use of mathematical programming techniques for approximating solutions along the Pareto front of a MOP. The proposed approach was, in principle, designed for dealing with box-constrained multi-objective optimization problems having low and moderate dimensionality (between 2 and 12 decision variables).

Our results indicate that our proposed MONSS outperforms a state-of-the-art multi-objective evolutionary algorithm (MOEA/D) in terms of convergence in most of the test problems adopted. The number of objective function evaluations in these test problems was restricted to 4,000 (for traditional test problems) and 20,000 (for the CEC'2009 test problems) for the two-objective problems. In the

Table 3 Results corresponding to the I_H performance measure for MONSS and MOEA/D

MOP	MONSS I_H (σ)	MOEA/D I_H (σ)	$P(H)$	reference vector ($\mathbf{r}$)
DEB2	**0.983980** (0.000253)	0.980044 (0.018523)	0.145286 (0)	$(1.1, 1.1)^T$
DTLZ5	0.434615 (0.001253)	**0.437779** (0.000623)	0.145286 (0)	$(1.1, 1.1, 1.1)^T$
FON2	**0.545732** (0.000916)	0.542475 (0.001483)	0.137323 (0)	$(1.1, 1.1)^T$
LAU	**14.052315** (0.004976)	13.970412 (0.029470)	**0.000000 (1)**	$(4.1, 4.1)^T$
LIS	**0.315755** (0.002831)	0.260127 (0.009704)	**0.000000 (1)**	$(1.0, 1.0)^T$
MUR	**3.144421** (0.000428)	3.143494 (0.000980)	0.761822 (0)	$(4.1, 4.1)^T$
REN1	**3.613325** (0.000627)	3.598263 (0.019908)	0.520139 (0)	$(37.1, 1.1)^T$
REN2	**19.013226** (0.002713)	19.011884 (0.004712)	**0.000000 (1)**	$(-1.9, 2.1)^T$
VNT2	**2.165237** (0.000895)	2.163965 (0.001096)	**0.000002 (1)**	$(4.5, -16.0, -11.5)^T$
VNT3	**11.735691** (0.013317)	11.723060 (0.035286)	0.579294 (0)	$(8.5, 17.5, 0.5)^T$
UF1	**0.807203** (0.018045)	0.700705 (0.056916)	**0.000000 (1)**	$(1.1, 1.1)^T$
UF2	**0.838738** (0.018344)	0.829808 (0.011885)	**0.007288 (1)**	$(1.1, 1.1)^T$
UF3	**0.543851** (0.094089)	0.454699 (0.089608)	**0.000253 (1)**	$(1.1, 1.1)^T$
UF4	**0.443920** (0.010838)	0.429724 (0.010334)	**0.000012 (1)**	$(1.1, 1.1)^T$
UF5	0.038956 (0.047168)	**0.221446** (0.090002)	**0.000000 (1)**	$(1.1, 1.1)^T$
UF6	0.059901 (0.076414)	**0.193941** (0.100058)	**0.000004 (1)**	$(1.1, 1.1)^T$
UF7	**0.652922** (0.042732)	0.418607 (0.145031)	**0.000000 (1)**	$(1.1, 1.1)^T$
UF8	**0.704555** (0.010813)	0.668449 (0.020402)	**0.000000 (1)**	$(1.1, 1.1, 1.1)^T$
UF9	**1.050053** (0.006325)	0.961896 (0.084149)	**0.000000 (1)**	$(1.1, 1.1, 1.1)^T$
UF10	**0.325869** (0.096517)	0.295680 (0.117077)	0.304177 (0)	$(1.1, 1.1, 1.1)^T$

case of three objectives, the search was restricted to 12,000 (for traditional test problems) and 60,000 (for the CEC'2009 test problems) objective function evaluations. The good results obtained by our proposed approach with this relatively low number of objective function evaluations suggest that it can be a good choice for dealing with expensive objective functions.

The main motivation for the work presented here has been to show that it is possible to design a competitive multi-objective optimization algorithm using only direct search methods, and without relying on metaheuristic search mechanisms. It is, however, also clear to us that our proposed approach has some disadvantages with respect to multi-objective metaheuristics. The main ones have to do with the difficulties of the NSS method for moving in highly accidented search spaces. It is possible, however, to improve the performance of our proposed approach in such cases by varying the step sizes (i.e., the control parameters ρ, χ and γ) until finding a suitable region of the search space in which the NSS movements can be properly conducted. This is, however, an issue that deserves further research.

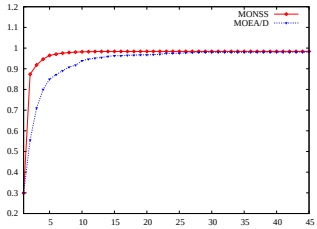
Table 4 Results corresponding to the I_C performance measure for MONSS and MOEA/D

MOP	I_C(MONSS, MOEA/D) (σ)	I_C(MOEA/D, MONSS) (σ)
DEB2	**0.068682** (0.022168)	0.028632 (0.011789)
DTLZ5	0.057920 (0.004841)	**0.065374** (0.048562)
FON2	**0.226095** (0.074152)	0.045333 (0.021868)
LAU	**0.093033** (0.081053)	0.047000 (0.025710)
LIS	**0.348161** (0.134523)	0.010101 (0.013040)
MUR	**0.033010** (0.013697)	0.013000 (0.009000)
REN1	**0.059347** (0.047529)	0.043000 (0.027586)
REN2	**0.005000** (0.006191)	0.004333 (0.005588)
VNT2	**0.044133** (0.013296)	0.018711 (0.006107)
VNT3	**0.231700** (0.041898)	0.035242 (0.010910)
UF1	**0.328315** (0.101923)	0.092295 (0.069648)
UF2	**0.532979** (0.058161)	0.042355 (0.019300)
UF3	**0.173409** (0.247981)	0.114337 (0.114386)
UF4	**0.303535** (0.079261)	0.235692 (0.091731)
UF5	0.025000 (0.098953)	**0.813333** (0.199555)
UF6	0.140000 (0.156205)	**0.156667** (0.272899)
UF7	**0.196863** (0.077797)	0.052480 (0.108817)
UF8	**0.125916** (0.040859)	0.102632 (0.020653)
UF9	**0.235295** (0.073415)	0.072039 (0.027979)
UF10	**0.236593** (0.154094)	0.117121 (0.103371)

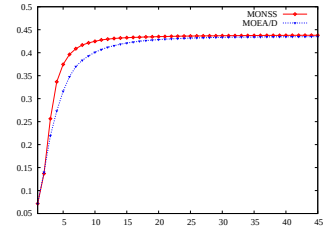
Motivated by the limitations of our proposed approach, we intend to hybridize it with a multi-objective evolutionary algorithm, so that it can be applied to multi-objective optimization problems of higher dimensionality and highly accidented search spaces. The idea of this hybridization is to use a multi-objective evolutionary algorithm to locate the promising regions of the search space and then adopt our MONSS algorithm to exploit such regions in an efficient manner. We believe that this sort of multi-objective memetic algorithm could be a powerful tool for solving complex and computationally expensive multi-objective optimization problems in an efficient and effective manner.

References

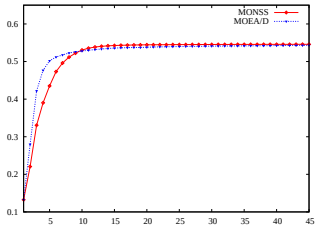
1. Audet, C., Savard, G., Zghal, W.: A mesh adaptive direct search algorithm for multiobjective optimization. European Journal of Operational Research **204**(3), 545–556 (2010)
2. Árpád Bűrmen, Puhan, J., Tuma, T.: Grid Restrained Nelder-Mead Algorithm. Computational Optimization and Applications **34**, 359–375 (2006)



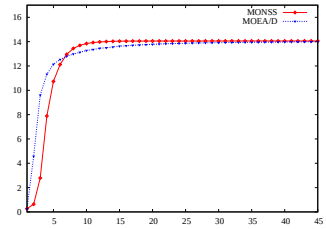
A) I_H convergence for DEB2 problem



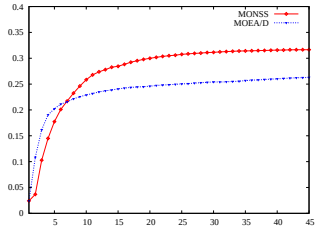
B) I_H convergence for DTLZ5 problem



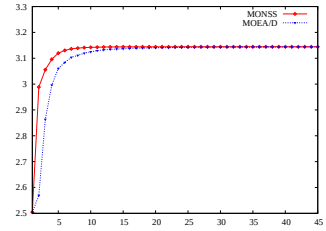
C) I_H convergence for FON2 problem



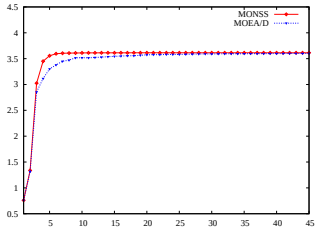
D) I_H convergence for LAU problem



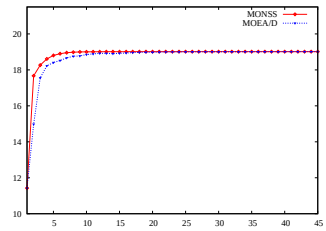
E) I_H convergence for LIS problem



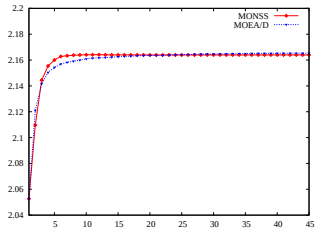
F) I_H convergence for MUR problem



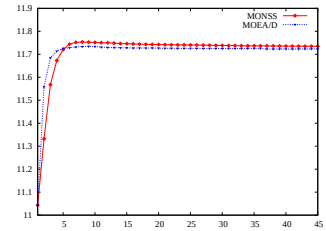
G) I_H convergence for REN1 problem



H) I_H convergence for REN2 problem

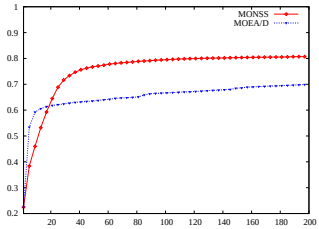


I) I_H convergence for VNT2 problem

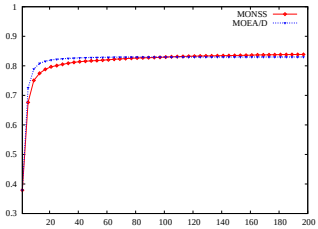


J) I_H convergence for VNT3 problem

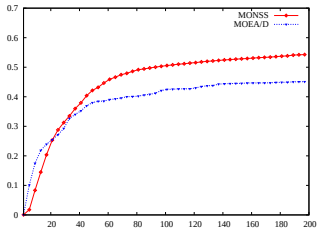
Fig. 3 I_H convergence plots for MONSS and MOEA/D in the test problems DEB2, DTLZ5, FON2, LAU, LIS, MUR, REN1, REN2, VNT2 and VNT3. The x axis and y axis show the number of iterations and the I_H value, respectively



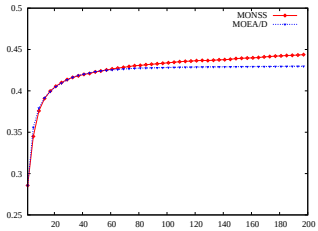
A) I_H convergence for UF1 problem



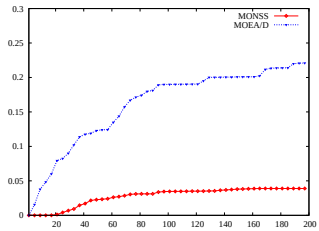
B) I_H convergence for UF2 problem



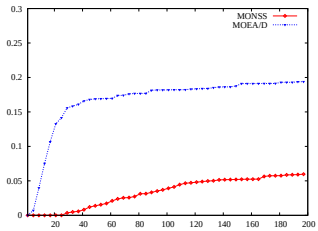
C) I_H convergence for UF3 problem



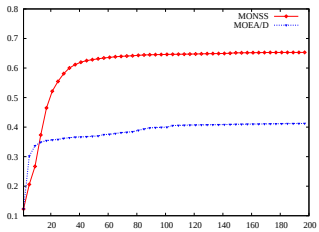
D) I_H convergence for UF4 problem



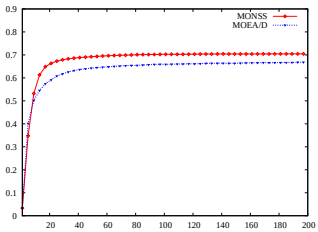
E) I_H convergence for UF5 problem



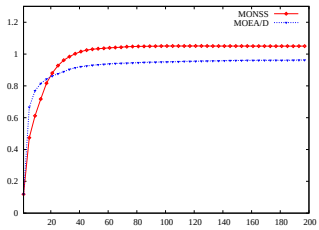
F) I_H convergence for UF6 problem



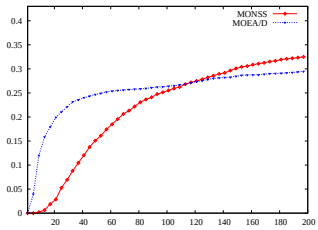
G) I_H convergence for UF7 problem



H) I_H convergence for UF8 problem



I) I_H convergence for UF9 problem



J) I_H convergence for UF10 problem

Fig. 4 Convergence plot for MONSS and MOEA/D in the test problems UF1-UF10. The x axis and y axis show the number of iterations and the I_H value, respectively

3. Byatt, D.: A Convergent Variants of the Nelder-Mead Algorithm. Master's thesis, University of Canterbury (2000)
4. 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), 335–348 (2003)
5. Coello Coello, C.A., Lamont, G.B. (eds.): Applications of Multi-Objective Evolutionary Algorithms. World Scientific, Singapore (2004). ISBN 981-256-106-4
6. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-Objective Problems, second edn. Springer, New York (2007). ISBN 978-0-387-33254-3
7. Deb, K.: Multi-Objective Genetic Algorithms: Problem Difficulties and Construction of Test Problems. Evolutionary Computation **7**(3), 205–230 (1999)
8. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
9. Deb, K., Thiele, L., Laumanns, M., Zitzler, E.: Scalable Test Problems for Evolutionary Multiobjective Optimization. In: A. Abraham, L. Jain, R. Goldberg (eds.) Evolutionary Multiobjective Optimization. Theoretical Advances and Applications, pp. 105–145. Springer, USA (2005)
10. Ehrgott, M.: Multicriteria Optimization, 2nd edition edn. Springer, Berlin (2005)
11. Fischer, A., Shukla, P.K.: A levenberg-marquardt algorithm for unconstrained multicriteria optimization. Oper. Res. Lett. **36**(5), 643–646 (2008)
12. Fliege, J., Graña Drummond, L.M., Svaiter, B.F.: Newton's Method for Multiobjective Optimization. SIAM Journal on Optimization **20**(2), 602–626 (2009)
13. 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, pp. 42–52. IEE, Sheffield, UK (1995)
14. Glover, F.: Tabu search and adaptive memory programing – Advances, applications and challenges. In: Interfaces in Computer Science and Operations Research, pp. 1–75. Kluwer Academic Publishers (1996)
15. Hatcher, A.: Algebraic Topology, 1 edn. Cambridge University Press (2001)
16. 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), pp. 1617–1624. IEEE Service Center, Hong Kong (2008)
17. Koduru, P., Das, S., Welch, S.M.: Multi-Objective Hybrid PSO Using ϵ-Fuzzy Dominance. In: D. Thierens (ed.) 2007 Genetic and Evolutionary Computation Conference (GECCO'2007), vol. 1, pp. 853–860. ACM Press, London, UK (2007)
18. 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**, 112–147 (1998)
19. Laumanns, M., Rudolph, G., Schwefel, H.P.: A Spatial Predator-Prey Approach to Multi-Objective Optimization: A Preliminary Study. In: A.E. Eiben, M. Schoenauer, H.P. Schwefel (eds.) Parallel Problem Solving From Nature — PPSN V, pp. 241–249. Springer-Verlag, Amsterdam, Holland (1998)
20. Lis, J., Eiben, A.E.: A Multi-Sexual Genetic Algorithm for Multiobjective Optimization. In: T. Fukuda, T. Furuhashi (eds.) Proceedings of the 1996 International Conference on Evolutionary Computation, pp. 59–64. IEEE, Nagoya, Japan (1996)
21. MacQueen, J.B.: Some Methods for Classification and Analysis of Multivariate Observations. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297. University of California Press (1967)
22. McKinnon, K.I.M.: Convergence of the Nelder–Mead Simplex Method to a Nonstationary Point. SIAM Journal on Optimization **9**(1), 148–158 (1998)
23. Miettinen, K.: Nonlinear Multiobjective Optimization. Kluwer Academic Publishers, Boston, Massachusetts (1999)
24. Mukai, H.: Algorithms for Multicriterion Optimization. IEEE Transactions on Automatic Control **25**(2), 177–186 (1980)
25. Murata, T., Ishibuchi, H.: MOGA: Multi-Objective Genetic Algorithms. In: Proceedings of the 2nd IEEE International Conference on Evolutionary Computing, pp. 289–294. Perth, Australia (1995)

26. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. The Computer Journal **7**, 308–313 (1965)
27. 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), 40–58 (2006)
28. Schütze, O., Mostaghim, S., Dellnitz, M., Teich, J.: Covering pareto sets by multilevel evolutionary subdivision techniques. In: Proceedings of the 2nd international conference on Evolutionary multi-criterion optimization, EMO'03, pp. 118–132. Springer-Verlag, Berlin, Heidelberg (2003)
29. Spendley, W., Hext, G.R., Himsworth, F.R.: Sequential Application of Simplex Designs in Optimization and Evolutionary Operation. Technometrics **4**(4), 441–461 (1962)
30. Torczon, V.J.: Multi-directional search: A direct search algorithm for parallel machines. Ph.D. thesis, Rice University (1989)
31. Trabia, M.B., Lu, X.B.: A Fuzzy Adaptive Simplex Search Optimization Algorithm. Journal of Mechanical Design **123**, 216–225 (2001)
32. Valenzuela-Rendón, M., Uresti-Charre, E.: A Non-Generational Genetic Algorithm for Multiobjective Optimization. In: T. Bäck (ed.) Proceedings of the Seventh International Conference on Genetic Algorithms, pp. 658–665. Michigan State University, Morgan Kaufmann Publishers, San Mateo, California (1997)
33. Viennet, R., Fontiex, C., Marc, I.: Multicriteria Optimization Using a Genetic Algorithm for Determining a Pareto Set. International Journal of Systems Science **27**(2), 255–260 (1996)
34. Vincke, P.: Multicriteria Decision-Aid. John Wiley & Sons, New York (1992)
35. Wilcoxon, F.: Individual comparisons by ranking methods. Biometrics Bulletin **1**(6), 80–83 (1945)
36. Zapotecas Martínez, S., Coello Coello, C.A.: A Proposal to Hybridize Multi-Objective Evolutionary Algorithms with Non-Gradient Mathematical Programming Techniques. In: G. Rudolph, T. Jansen, S. Lucas, C. Poloni, N. Beume (eds.) Parallel Problem Solving from Nature–PPSN X, vol. 5199, pp. 837–846. Springer. Lecture Notes in Computer Science, Dortmund, Germany (2008)
37. Zapotecas Martínez, S., Coello Coello, C.A.: A Memetic Algorithm with Non Gradient-Based Local Search Assisted by a Meta-Model. In: R. Schaefer, C. Cotta, J. Kolodziej, G. Rudolph (eds.) Parallel Problem Solving from Nature–PPSN XI, *Lecture Notes in Computer Science*, vol. 6238, pp. 576–585. Springer, Kraków, Poland (2010)
38. Zhang, Q., Li, H.: MOEA/D: A Multiobjective Evolutionary Algorithm Based on Decomposition. IEEE Transactions on Evolutionary Computation **11**(6), 712–731 (2007)
39. Zhang, Q., Zhou, A., Zhao, S., Suganthan, P.N., Liu, W., Tiwari, S.: Multiobjective optimization test instances for the cec 2009 special session and competition. Tech. Rep. CES-487, University of Essex and Nanyang Technological University (2008)
40. Zitzler, E., Deb, K., Thiele, L.: Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. Evolutionary Computation **8**(2), 173–195 (2000)
41. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm. In: K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, T. Fogarty (eds.) EUROGEN 2001. Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, pp. 95–100. Athens, Greece (2001)
42. Zitzler, E., Thiele, L.: Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Study. In: A.E. Eiben (ed.) Parallel Problem Solving from Nature V, pp. 292–301. Springer-Verlag, Amsterdam (1998)
43. 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), 117–132 (2003)