Yokohama Publishers
ISSN 1349-8169 ONLINE JOURNAL

Pacific Journal of Optimization
Vol. 16, No. 1, pp. 39 – 65, 2020

A NONLINEAR SCALARIZATION METHOD FOR MULTI-OBJECTIVE OPTIMIZATION PROBLEMS*

Qiang Long, Lin Jiang† and Guoquan Li

Abstract: This paper presents population-based linear and nonlinear scalarization method for solving multi-objective optimization problems (MOPs). Firstly, we extend the weighted sum method by generating a set of different weights and solving a series of corresponding scalar optimization problems. This mechanism obtains an approximation of all Pareto solutions. However, the extended weighted sum method only works for convex MOPs. For nonconvex MOPs, nonlinear scalarization mechanisms have to be considered. Therefore, the weighted sum method is additionally extended to nonlinear case and a nonlinear scalarization method for nonconvex MOPs is proposed. It turns out that the proposed method not only works for nonconvex MOPs but also for MOPs with disconnected Pareto frontiers. Comprehensively numerical experiments are presented with the results and analysis showing that the proposed methods are efficient in solving various kinds of MOPs.

Key words: *multiple objective optimization, weighted sum method, nonlinear scalarization method, pareto solutions*

Mathematics Subject Classification: *90C31, 49M37, 49Mxxp*

1 Introduction

Multi-objective optimization has extensive applications in engineering and management [8, 20, 22]. Most real-world optimization problems have multiple objectives, which can be modelled as multi-objective optimization problems (MOPs). However, due to the theoretical and computational challenges, it is not easy to solve MOPs. Therefore, multi-objective optimization has attracted a wide range of research over the last few decades.

Broadly, methods for MOPs can be categorized into three types: direct, indirect and hybrid. Population-based metaheuristic methods, such as genetic algorithm and evolutionary strategy, lend themselves to direct methods. Their iterative unit is a population instead of a single point, so they can obtain the entire set of Pareto solutions or a representative subset of it. Some typical multi-objective evolutionary algorithms (MOEAs) can be found in the works of Deb et al. [5] and Long et al. [25, 26]. Indirect methods, mainly mean scalarization methods, reformulate MOP to a single-objective optimization problem. Normally, the Pareto solution of MOP and the optimal solution of single-objective optimization problems are corresponded. Typical indirect methods are the weighted sum method [10, 11], $\varepsilon-$constraint

*This work is supported by the National Natural Science Foundation of China (Grants No. 11671062 and 11871128).

†Corresponding Author.

method [17], Tchebycheff approach [21,28], normal-boundary intersection method [4,23] and Pascoletti-Serafini approach [6]. In a single run, direct methods can find an approximation of the set of Pareto solutions, while indirect methods only get one Pareto solution. Hybrid methods combine advantages of direct and indirect methods. They are based on scalar transformation and take into account heuristic ideas at the same time. A typical hybrid method is MOEA/D [38].

For indirect methods, previous research mainly focused on scalar techniques rather than on how to find an approximation of the set of Pareto solutions. This paper tries to fill this gap by extending the scalarization methods, such as the weighted sum method, to population-based. An intuitive strategy is to run a scalarization method many times using different parameters. For example, one can apply a set of different weights to the weighed sum method, and each weight will end up with a Pareto solution. However, many difficulties exist, such as how to choose weights in order to get uniformly and comprehensively distributed Pareto solutions. Furthermore, the weighted sum method works well only on convex MOPs. For nonconvex ones, nonlinear mechanisms have to be considered. Motivated by these issues, in this paper, we are going to tackle the following topics:

(1) Extend the weighted sum method to population-based and apply it to convex MOPs;

(2) Design a population-based nonlinear scalarization method and apply it to nonconvex MOPs;

(3) Study numerical performances of the proposed linear and nonlinear scalarization methods.

The rest of this paper is organized as follows: In Section 2, some basic definitions and theories of multi-objective optimization are reviewed. In Section 3, we present an extended weighted sum method. In Section 4, a nonlinear scalarization method for nonconvex MOP is designed, and some theoretical analysis are also presented. In Section 5, some implementation details of the proposed algorithms are provided. In Section 6, numerical experiments are considered. Section 7 concludes the paper.

2 Preliminaries

The general mathematical model of the constrained multi-objective optimization problem is as follows,

$$\text{(CMOP)} \quad \begin{cases} \text{Minimize} & \boldsymbol{F}(\boldsymbol{x}) \\ \text{Subject to} & g_i(\boldsymbol{x}) \le 0, \quad i = 1, \dots, p \\ & h_j(\boldsymbol{x}) = 0, \quad j = 1, \dots, q \\ & \boldsymbol{x} \in X, \end{cases} \tag{2.1}$$

where $\boldsymbol{F} : \mathbb{R}^n \mapsto \mathbb{R}^m$ $(\boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \dots, f_m(\boldsymbol{x}))^T)$ is a vector-valued function, $f_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \dots, m$; $g_i : \mathbb{R}^n \mapsto \mathbb{R}$, $i = 1, \dots, p$; $h_j : \mathbb{R}^n \mapsto \mathbb{R}$, $j = 1, \dots, q$ are Lipschitz continuous functions. $X = \{\boldsymbol{x} \in \mathbb{R}^n \mid l_i \le x_i \le u_i\} \subset \mathbb{R}^n$ is a box set, $\boldsymbol{l} = (l_1, l_2, \dots, l_n)^T$ and $\boldsymbol{u} = (u_1, u_2, \dots, u_n)^T$ are lower and upper bounds, respectively. Denote feasible set

$$\Omega = \{\boldsymbol{x} \in X \mid g_i(\boldsymbol{x}) \le 0,\ i = 1, \dots, p;\ h_j(\boldsymbol{x}) = 0,\ j = 1, \dots, q\},$$

then Problem (2.1) can be simplified as

$$\begin{cases} \text{Minimize} & \boldsymbol{F}(\boldsymbol{x}) \\ \text{Subject to} & \boldsymbol{x} \in \Omega. \end{cases} \tag{2.2}$$

In multi-objective optimization, we call feasible set Ω as the *decision variable space* and its image set $\boldsymbol{F}(\Omega) = \{y = \boldsymbol{F}(\boldsymbol{x}) \mid x \in \Omega\}$ as the *objective function value space*. In the following, some definitions and theorems are reviewed.

Given two vectors

$$\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^T \text{ and } \boldsymbol{z} = (z_1, z_2, \ldots, z_m)^T \in \mathbb{R}^m,$$

then

$$\boldsymbol{y} = \boldsymbol{z} \Leftrightarrow y_i = z_i \text{ for all } i = 1, 2, \ldots, m;$$
$$\boldsymbol{y} \leq \boldsymbol{z} \Leftrightarrow y_i \leq z_i \text{ for all } i = 1, 2, \ldots, m;$$
$$\boldsymbol{y} \prec \boldsymbol{z} \Leftrightarrow y_i < z_i \text{ for all } i = 1, 2, \ldots, m;$$
$$\boldsymbol{y} \preceq \boldsymbol{z} \Leftrightarrow y_i \leq z_i \text{ for all } i = 1, 2, \ldots, m, \text{ and } \boldsymbol{y} \neq \boldsymbol{z}.$$

" $\geq$ ", "$\succ$" and "$\succeq$" can be defined similarly. In this paper, if $\boldsymbol{y} \preceq \boldsymbol{z}$, we say $\boldsymbol{y}$ *dominates* $\boldsymbol{z}$ or $\boldsymbol{z}$ *is dominated by* $\boldsymbol{y}$.

Definition 2.1. *Suppose that* $\mathbf{y} \subseteq \mathbb{R}^m$ *and* $\mathbf{y}^* \in Y$. *If* $\mathbf{y}^* \leq y$ *for any* $\mathbf{y} \in Y$, *then* $\mathbf{y}^*$ *is called an* absolutely minimal point *of* Y.

In the sense of minimization, absolutely minimal point is an ideal point but may not exist.

Definition 2.2. *Let* $\mathbf{y} \in \mathbb{R}^m$ *and* $\mathbf{y}^* \in Y$. *If there is no* $\mathbf{y} \in Y$ *such that*

$$\mathbf{y} \preceq \mathbf{y}^* \ (\textit{or } \mathbf{y} \prec \mathbf{y}^*),$$

then $\mathbf{y}^*$ *is called an* efficient point *(or* weakly efficient point*) of* Y.

The sets of absolutely minimal points, efficient points and weakly efficient points of Y are denoted as Y_{ab}, Y_{ep} and Y_{wp}, respectively. Obviously, we have $Y_{ab} \subset Y_{ep} \subset Y_{wp}$

Definition 2.3. *Suppose that* $\mathbf{x}^* \in \Omega$. *If* $\mathbf{F}(\mathbf{x}^*) \leq \mathbf{F}(\mathbf{x})$, *for any* $\mathbf{x} \in \Omega$, $\mathbf{x}^*$ *is called an* absolutely minimal solution *of Problem (2.2). The set of absolutely minimal solution is denoted as* Ω_{as}.

The concept of the absolutely minimal solution is a direct generalization of that in single-objective optimization. It is an ideal solution but may not exist for most cases.

Definition 2.4. *Suppose that* $\mathbf{x}^* \in \Omega$. *If there is no* $\mathbf{x} \in \Omega$ *such that* $\mathbf{F}(\mathbf{x}) \preceq \mathbf{F}(\mathbf{x}^*)$ (*or* $\mathbf{F}(\mathbf{x}) \prec \mathbf{F}(\mathbf{x}^*)$), *i.e.* $\mathbf{F}(\mathbf{x}^*)$ *is an efficient point (or weakly efficient point) of the objective function value space* $\mathbf{F}(\Omega)$, *then* $\mathbf{x}^*$ *is called an* efficient solution *(or* weakly efficient solution*) of Problem (2.2). The sets of efficient solutions and weakly efficient solutions are denoted as* Ω_{es} *and* Ω_{ws}, *respectively.*

Another name of the efficient solution is *Pareto solution*, which is introduced by T.C. Koopmans in 1951 [14]. The meaning of Pareto solution is that, if $\boldsymbol{x}^* \in \Omega_{es}$, then there is no feasible solution $\boldsymbol{x} \in \Omega$, such that any $f_i(\boldsymbol{x})$ of $\boldsymbol{F}(\boldsymbol{x})$ is not worse than that of $\boldsymbol{F}(\boldsymbol{x}^*)$ and there is at least one $i_0 \in \{1, 2, \ldots, m\}$ such that $f_{i_0}(\boldsymbol{x}) < f_{i_0}(\boldsymbol{x}^*)$. In other words, $\boldsymbol{x}^*$ is the best solution in the sense of "$\preceq$". Another intuitive interpretation of Pareto solution is that it cannot be improved with respect to any objective without worsening at least one of

the others. Weakly efficient solution is introduced by S. Karlin in 1960 [13], it means that if $\boldsymbol{x}^* \in \Omega_{ws}$, then there is no feasible solution $\boldsymbol{x} \in \Omega$, such that any $f_i(\boldsymbol{x})$ of $\boldsymbol{F}(\boldsymbol{x})$ is strictly better than that of $\boldsymbol{F}(\boldsymbol{x}^*)$. In other words, $\boldsymbol{x}^*$ is the best solution in the sense of "$\prec$". The set of Pareto solutions is denoted by $\mathcal{P}^*$. Its image set $\boldsymbol{F}(\mathcal{P}^*)$ is called the *Pareto frontier*, denoted by $\mathcal{PF}^*$.

3 Extended Weighted Sum Method

In this section, we consider linear scalarization methods, more specifically, the weighted sum method. Firstly, the relationship between optimal solutions of the scalarization problem and (weakly) efficient solutions of the original MOP is theoretically reviewed, then an extended weighted sum method is presented.

For Problem (2.2), consider the following scalar optimization problem

$$\text{(SOP)} \quad \begin{cases} \text{Minimize} & \sum\limits_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) \\ \text{Subject to} & \boldsymbol{x} \in \Omega, \end{cases} \tag{3.1}$$

where $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T \in \Lambda^+$ (or Λ^{++}) is a scalar vector. We call Problem (3.1) a *weighted sum scalarization* of Problem (2.2). Here

$$\Lambda^+ = \lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T \mid \lambda_i \geq 0 \text{ and } \sum_{i=1}^{m} \lambda_i = 1\},$$

and

$$\Lambda^{++} = \{\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)^T \mid \lambda_i > 0 \text{ and } \sum_{i=1}^{m} \lambda_i = 1\}.$$

In the following, we theoretically analyze the relationship between Problem (2.2) and (3.1). For the sake of convenience, denote

$$\Phi_\lambda(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}).$$

Definition 3.1. *Suppose that $\Omega \in \mathbb{R}^n$ is a convex set, $\mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T$ is a vector-valued function, if all $f_i(\mathbf{x})$, $i = 1, 2, \ldots, m$, are (strictly) convex on Ω, then we call $\mathbf{F}(\mathbf{x})$ an* m-dimensional (strictly) convex vector-valued function *on Ω.*

Definition 3.2. *If the feasible set Ω is convex, and the multi-objective function $\mathbf{F}(\mathbf{x})$ is a convex vector-valued function on Ω, then we call Problem (2.2) a* convex MOP.

It is clear that $\Phi_\lambda(\boldsymbol{x})$ is convex if $\boldsymbol{F}(\boldsymbol{x})$ is convex. Therefore, Problem (3.1) is a convex problem if Problem (2.2) is a convex problem.

Theorem 3.3. For a given $\lambda \in \Lambda^{++}$ (or Λ^+), the optimal solution of Problem (3.1) is an efficient (or weakly efficient) solution of Problem (2.2).

Theorem 3.4. If Problem (2.2) is convex, then for any efficient solution (or weakly efficient solution) $\boldsymbol{x}^*$, there exist a $\lambda \in \Lambda^{++}$ (or $\lambda \in \Lambda^+$), such that $\boldsymbol{x}^*$ is an optimal solution of Problem (3.1).

Proofs of Theorems 3.3 and 3.4 can be found in the book written by Ehrgott [7]. These two theorems reveal that for a convex MOP, there is an one-to-one relationship between the weakly efficient solution of Problem (2.2) and the optimal solution of Problem (3.1). Based on this sense, we design the following extended weighted sum method.

Algorithm 1: Extended weighted sum method (EWSM)

Input: Problem parameters: f_i, g_i, h_i and X, number of solutions: N
Output: Pareto solutions: $\mathcal{P}$, Pareto frontier: $\mathcal{PF}$
// The main loops;

Step 1: Generate N weights $\lambda \in \Lambda^+$, store them in $\bar{\Lambda}$, so $\bar{\Lambda} \subset \Lambda^+$.

Step 2: For each $\lambda \in \bar{\Lambda}$, globally solve Problem (3.1), the obtained optimal solution $\boldsymbol{x}_\lambda^*$ is an weakly efficient solution of Problem (2.2), store $\boldsymbol{x}_\lambda^*$ in $\mathcal{P}$.

Step 3: Compute set $F_{\bar{\Lambda}} = \{\boldsymbol{F}(\boldsymbol{x}_\lambda^*) \mid \lambda \in \bar{\Lambda}\}$, then $F_{\bar{\Lambda}}$ is an approximate Pareto frontier of Problem (2.2), let $\mathcal{PF} = F_{\bar{\Lambda}}$.

The following are some remarks about Algorithm 1:

(1) In Step 1, approaches to construct the finite subset $\bar{\Lambda}$ are various. Two intuitive approaches are presented here: (i) all $\lambda \in \Lambda^+$ consist of a simplex, so λ can be uniformly picked on this simplex; (ii) randomly pick finite number of $\lambda \in \mathbb{R}_+^m$, then normalize them to construct $\bar{\Lambda}$. More details about constructing $\bar{\Lambda}$ refers to Section 6.

(2) In Step 2, a global optimization method is needed to solve Problem (3.1). If Problem (2.2) is convex (so is Problem (3.1)), Problem (3.1) can be efficiently solved use any convex optimization solver. If objective functions of Problem (2.2) are nonconvex and complicated, which may lead a tough Problem (3.1), it will be not easy to solve this problem using Algorithm 1.

(3) For different $\lambda \in \bar{\Lambda}$, Problems (3.1) are independent with each other, so the parallel computing mechanism can be introduced to Algorithm 1, which will dramatically increase its efficiency. Section 6 will introduce more implementation details.

The geometrical explanation of the extended weighted sum method is given as follows. As shown in Figure 1(b), $\boldsymbol{F}(\Omega)$ is the image set of $\boldsymbol{F}(\boldsymbol{x})$ on $\Omega \subset \mathbb{R}^n$. For $\lambda \in \Lambda^+$,

$$\Phi_\lambda(\boldsymbol{x}) = \sum_{i=1}^{m} \lambda_i f_i(\boldsymbol{x}) \triangleq \sum_{i=1}^{m} \lambda_i f_i = \lambda^T f$$

is a linear function of $f = (f_1, f_2, \ldots, f_m)^T$, where $f_i = f_i(\boldsymbol{x})$, $i = 1, 2, \ldots, m$. Therefore, solving Problem (2) equals to minimize a linear function on the image set $\boldsymbol{F}(\Omega)$, i.e.,

$$\begin{cases} \text{Minimize} & \lambda^T f \\ \text{Subject to} & f \in \boldsymbol{F}(\Omega). \end{cases} \tag{3.2}$$

If f_λ^* solves Problem (3.2) for $\lambda \in \Lambda^+$, then f_λ^* must be a point in the Pareto frontier. Meanwhile, the corresponding $\boldsymbol{x}_\lambda^*$, i.e., $f_\lambda^* = \boldsymbol{F}(\boldsymbol{x}_\lambda^*)$, is an efficient solution. Although Problem (3.2) is simpler than Problem (3.1), we cannot directly work on Problem (3.2) because, first of all, the image set $\boldsymbol{F}(\Omega)$ cannot be exactly calculated; and second of all, even a solution f_λ^* is obtained, we have to solve the nonlinear equation $\boldsymbol{F}(\boldsymbol{x}) = f_\lambda^*$ to get the corresponding efficient solution $\boldsymbol{x}_\lambda^*$, which is a complex or even impossible task.

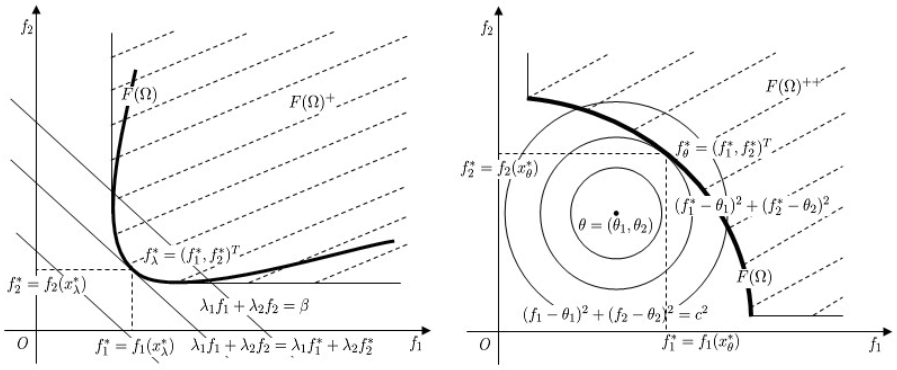
From the geometrical explanation, we can easily observe that the extended linear scalarization method works only on the problem whose image set is convex on the Pareto frontier, i.e.,

$$\boldsymbol{F}(\Omega)^+ = \{f + d \mid f \in \boldsymbol{F}(\Omega) \text{ and } d \in \mathbb{R}^m_+\}$$

is convex. Here

$$\mathbb{R}^m_+ = \{\boldsymbol{x} = (x_1, x_2, \ldots, x_m) \mid x_i \geq 0,\ i = 1, 2, \ldots, m\}.$$

If $\boldsymbol{F}(\Omega)^+$ is nonconvex (e.g., 1(b)), only the boundary point of Pareto frontier can be obtained using Algorithm 1. However, inspired by Problem (3.2), we can nonlinearly scalarize the multi-objective function.



(a) Extended weighted sum method (b) Nonlinear scalarization method

Figure 1: Geometrical meaning of linear and nonlinear scalarization methods.

4 Nonlinear Scalarization Method

Nonlinear scalarization method is nothing but changing the linear objective function in Problem (3.2) into nonlinear function. A naive thought is to use a quadratic function, more specifically, m-dimensional sphere, as objective function. Therefore, we can construct the following problem

$$\begin{cases} \text{Minimize} & \sum_{i=1}^{m} (f_i - \theta_i)^2 \\ \text{Subject to} & f \in \boldsymbol{F}(\Omega), \end{cases} \tag{4.1}$$

where $f = (f_1, f_2, \ldots, f_m)^T$ and $\theta = (\theta_1, \theta_2, \ldots, \theta_m)^T \in \mathbb{R}^m$. The geometrical explanation of Problem (4.1) is shown in Figure **??**. From the figure, it is possible to solve nonconvex MOPs using the nonlinear scalarization techniques. Like Problem (3.2), directly working on Problem (4.1) is out of option, but taking into account $f_i = f_i(\boldsymbol{x})$ $i = 1, 2, \ldots, m$, it can be transformed into the following problem,

$$\begin{cases} \text{Minimize} & \sum_{i=1}^{m} (f_i(\boldsymbol{x}) - \theta_i)^2 \\ \text{Subject to} & \boldsymbol{x} \in \Omega. \end{cases} \tag{4.2}$$

In the following, we discuss the relationship between the efficient solution of Problem (2.2) and the global optimal solution of Problem (4.2). For the sake of convenience, denote the objective function of Problem (4.2) as

$$\Psi_\theta(\boldsymbol{x}) = \sum_{i=1}^{m}(f_i(\boldsymbol{x}) - \theta_i)^2.$$

Suppose that $\bar{F}^* = (\bar{f}_1^*, \bar{f}_2^*, \ldots, \bar{f}_m^*)^T$, where

$$\bar{f}_i^* = \min_{x\in\Omega}\ f_i(\boldsymbol{x}), \quad i = 1, 2, \ldots, m.$$

In paper [4], $\bar{F}^*$ is called a *shadow minimum* or *utopia point*. Construct a set as

$$\bar{\Theta} = \{\theta = (\theta_1, \theta_2, \ldots, \theta_m)^T \mid \theta_i \le \bar{f}_i^*\},$$

the element of $\bar{\Theta}$ is called a *referential point*. We can have the following theorem.

Theorem 4.1. For a given $\theta \in \bar{\Theta}$, if $\boldsymbol{x}^*$ is a global minimal solution of Problem (4.2), then $\boldsymbol{x}^*$ must be an efficient solution of Problem (2.2).

Proof. Assume that $\boldsymbol{x}^*$ is a global minimal solution of Problem (4.2) but not an efficient solution of Problem (2.2), then there exists $\bar{\boldsymbol{x}} \in \Omega$ such that $\boldsymbol{F}(\bar{\boldsymbol{x}}) \preceq \boldsymbol{F}(\boldsymbol{x}^*)$, i.e.,

$$f_i(\bar{\boldsymbol{x}}) \le f_i(\boldsymbol{x}^*)\ \ i = 1, 2, \ldots, m;\ \text{and}\ \exists\ i_0\ \text{s.t.}\ f_{i_0}(\bar{\boldsymbol{x}}) < f_{i_0}(\boldsymbol{x}^*).$$

So we have

$$(f_{i_0}(\bar{\boldsymbol{x}}) - \theta_{i_0})^2 < (f_{i_0}(\boldsymbol{x}^*) - \theta_{i_0})^2,$$

which yields

$$\sum_{i=1}^{m}(f_i(\bar{\boldsymbol{x}}) - \theta_i)^2 < \sum_{i=1}^{m}(f_i(\boldsymbol{x}^*) - \theta_i)^2,$$

i.e.,

$$\Psi_\theta(\bar{\boldsymbol{x}}) < \Psi_\theta(\boldsymbol{x}^*).$$

This contradicts to $\boldsymbol{x}^*$ is globally minimal, which proves the theorem. □

Remark 4.2. *Theorem 4.1 can be taken as a generalization of Theorem 3.3 for nonlinear scalarization. However, Theorem 3.4 cannot be generalized; we cannot obtain all efficient solutions through picking θ all over $\bar{\Theta}$.*

Based on Problem (4.2) and Theorem 4.1, we propose the following algorithm.

Algorithm 2: Nonlinear scalarization method (NSM)

Input: Problem parameters: f_i, g_i, h_i and X, number of solutions: N
Output: Pareto solutions: $\mathcal{P}$, Pareto frontier: $\mathcal{PF}$
// The main loops;

Step 1: Successively solve

$$\bar{f}_i^* = \min_{x\in\Omega} f_i(\boldsymbol{x}), \quad i = 1, 2, \ldots, m,$$

and then construct set

$$\bar{\Theta} = \{\theta = (\theta_1, \theta_2, \ldots, \theta_m)^T \in \mathbb{R}^m \mid \theta_i \leq \bar{f}_i^*\}.$$

Step 2: Choose N referential points $\theta \in \bar{\Theta}$, store in $\hat{\Theta}$, so $\hat{\Theta} \subset \bar{\Theta}$.

Step 3: For each $\theta \in \hat{\Theta}$, globally solve Problem (4.2), the global minimal solution $\boldsymbol{x}_\theta^*$ is an efficient solution of Problem (2.2), store $\boldsymbol{x}_\theta^*$ in $\mathcal{P}$.

Step 4: Compute $F_{\hat{\Theta}} = \{\boldsymbol{F}(\boldsymbol{x}_\theta^*) \mid \theta \in \hat{\Theta}\}$, then $F_{\hat{\Theta}}$ is an approximate Pareto frontier of Problem (2.2), let $\mathcal{PF} = F_{\hat{\Theta}}$.

The following are some remarks of Algorithm 2:

(1) In Step 1, completely solving these global optimization problems is not necessary since what we really need are just lower bounds of $f_i(\boldsymbol{x})$, $i = 1, 2, \ldots, m$, so a reasonable guess of their lower bounds is enough. More practically, assume that $f_i(\boldsymbol{x}) \geq 0$, $i = 1, 2, \ldots, m$ (if any one of them is not satisfied, we can always move it parallel without changing the efficient solutions of the original problem), then we can let $\bar{\Theta} = -\mathbb{R}_{++}^m$.

(2) In Step 2, $\bar{\Theta}$ is a lowerly unbounded set, so elements of $\hat{\Theta}$ should be chosen from its upper boundary.

(3) In Step 3, a global optimization solver is needed as well as in Algorithm 1, so it could be numerically difficult if the objective function of Problem (4.2) is complicated. In this situation, the original MOP is not suitable to be solved by this algorithm.

In Step 1 of Algorithm 2, we restrict $\theta \in \bar{\Theta}$ in order to guarantee that the global minimal solution of Problem (4.2) is an efficient solution of Problem (2.1). However, based on Remark 4.2, if only choose $\theta \in \bar{\Theta}$, we may never reach some parts of the Pareto frontier. Actually, if $\theta \notin \bar{\Theta}$ but properly chosen, we can also obtain an efficient solution. This is analyzed as follows.

Construct set

$$\bar{\Theta}' = \{\theta = (\theta_1, \theta_2, \ldots, \theta_m) \in \mathbb{R}^m \mid \exists\ i_0 \in \{1, 2, \ldots, m\},\ \text{s.t. } \theta_{i_0} < \bar{f}_{i_0}^*\},$$

obviously $\bar{\Theta} \subset \bar{\Theta}'$. For a given $\theta \in \bar{\Theta}'$, assume we have $i_1 \in \{1, 2, \ldots, m\}$, such that $\theta_{i_1} < \bar{f}_{i_1}^*$, in the proof of Theorem 4.1, if we just have $i_0 = i_1$, i.e.,

$$\theta_{i_0} < \bar{f}_{i_0}^* \leq f_{i_0}(\bar{\boldsymbol{x}}) < f_{i_0}(\boldsymbol{x}^*) \tag{4.3}$$

and

$$(f_{i_0}(\bar{\boldsymbol{x}}) - \theta_{i_0})^2 - (f_{i_0}(\boldsymbol{x}^*) - \theta_{i_0})^2 < \sum_{i=1, i\neq i_0}^{m} (f_i(\boldsymbol{x}^*) - \theta_i)^2 - \sum_{i=1, i\neq i_0}^{m} (f_i(\bar{\boldsymbol{x}}) - \theta_i)^2, \tag{4.4}$$

we can still have

$$\Psi_\theta(\bar{\boldsymbol{x}}) < \Psi_\theta(\boldsymbol{x}^*),$$

which yields that $\boldsymbol{x}^*$ is an efficient solution. Of course, these conditions cannot be checked in advance, if $\theta \in \bar{\Theta}'$, but conditions (4.3) or (4.4) cannot be satisfied, the obtained global minimal solution $\boldsymbol{x}^*$ may not be an efficient solution. But we can use a non-dominated sorting [5] to exclude these points. Based on this observation, we propose the following slack nonlinear scalarization method.

Algorithm 3: Slack nonlinear scalarization method (SNSM)

Input: Problem parameters: f_i, g_i, h_i and X, number of solutions: N
Output: Pareto solutions: $\mathcal{P}$, Pareto frontier: $\mathcal{PF}$
// The main loops;

Step 1: Successively solve

$$\bar{f}_i^* = \min_{x\in\Omega} f_i(\boldsymbol{x}) \quad i = 1, 2, \ldots, m,$$

and then construct set

$$\bar{\Theta}' = \{\theta = (\theta_1, \theta_2, \ldots, \theta_m) \in \mathbb{R}^m \mid \exists\ i_0 \in \{1, 2, \ldots, m\},\ \text{s.t. } \theta_{i_0} < \bar{f}_{i_0}^*\},$$

Step 2: Choose N referential points $\bar{\Theta}'$, store in $\hat{\Theta}'$, so $\hat{\Theta}' \subset \bar{\Theta}'$.

Step 3: For each $\theta \in \hat{\Theta}'$, globally solve the corresponding Problem (4.2), suppose that $\boldsymbol{x}_\theta^*$ is the global minimal solution, store in $S_{\hat{\Theta}'}$.

Step 4: Compute $F_{\hat{\Theta}'} = \{\boldsymbol{F}(\boldsymbol{x}_\theta^*) \mid \theta \in \hat{\Theta}'\}$.

Step 5: Successively check each $y \in F_{\hat{\Theta}'}$, if y is non-dominated , then store y in $\mathcal{PF}$ and its corresponding $\boldsymbol{x}_\theta^*$ in $\mathcal{P}$.

Step 5 of Algorithm 3 is actually a non-dominated ranking [5], here we pick the first Pareto frontier. The numerical comparison of Algorithm NSM and SNSM is presented in Section 6.1.

5 Implementation

In this section, we explain some implementation details of the proposed algorithms, including generating weights λ in Algorithm EWSM, generating referential points θ in Algorithm NSM and SNSM, and the global optimization solver for scalar optimization problems.

5.1 Generating λ in Algorithm EWSM

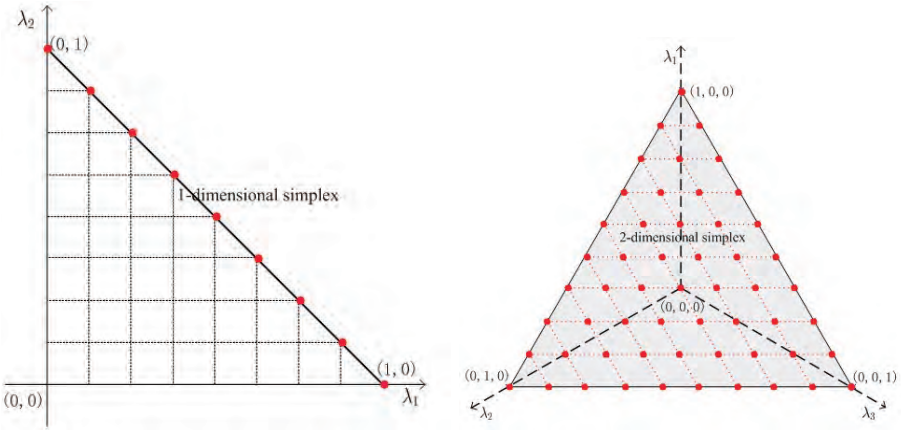
In Algorithm EWSM, all $\lambda \in \Lambda^+$ consist of a unit simplex, so finite number of weights should be uniformly generated in this unit simplex. The simplest strategy is to pick them randomly. For example, randomly choose $\lambda' \in [a, b]^m$, and then normalize λ' to be λ, i.e., $\lambda = \lambda' / \sum_{k=1}^m \lambda'_k$. Random strategy is simple and easy to implement, but it cannot guarantee uniformity, specially when ratio N/m is small.

In this paper, we apply a method called *systematic approach* to generate λ. This method was first introduced by Das and Dennis [4] and then applied by Deb [5]. It picks points in a normalized hyperplane (an $(m-1)$-dimensional unit simplex, which is equally inclined to all

axis and has an intercept of one on each axis). If p divisions are considered along each axis, the total number of different weights N for problem with m objective functions is given by

$$N = \begin{pmatrix} m+p-1 \\ p \end{pmatrix}.$$

Detial steps about the systematic approach can be found in the work of Das and Dennis [4], here we give two examples. Figure 2(a) demonstrates weights when $m = 2$, $p = 8$, there are $N = 9$ different weights; and Figure 2(b) demonstrates weights when $m = 3$, $p = 9$, there are $N = 55$ different weights. It can be observe that all the weights are uniformly distributed in the unit simplex.



(a) $m = 2$, $p = 8$, $N = 9$ (b) $m = 3$, $p = 9$, $N = 55$

Figure 2: Generating weights using systematic method.

5.2 Generating θ in Algorithm NSM and SNSM

In Algorithm NSM, θ is generated in $\bar{\Theta}$. Figure 4 illustrates two strategies to generate θ when $m = 2$. Here, we have

$$\boldsymbol{x}^*_{f_1} = \arg\min_{\boldsymbol{x}\in\Omega} f_1(\boldsymbol{x}),\ \ \bar{f}^*_1 = f_1(\boldsymbol{x}^*_{f_1})$$

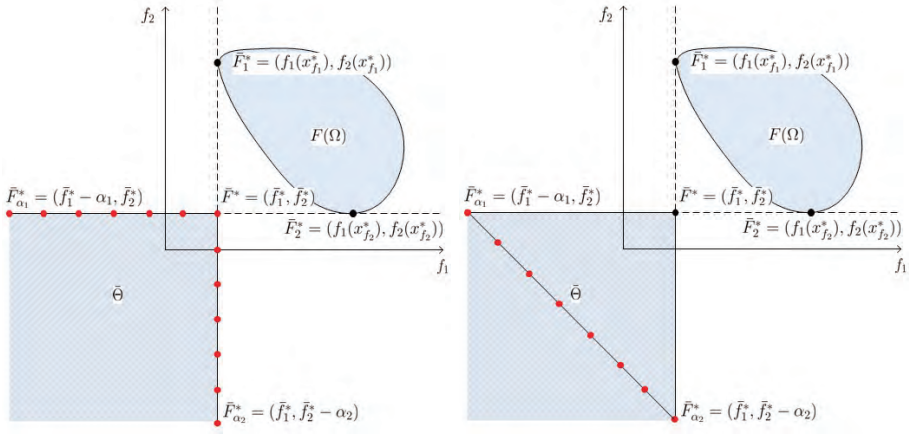
and

$$\boldsymbol{x}^*_{f_2} = \arg\min_{\boldsymbol{x}\in\Omega} f_2(\boldsymbol{x}),\ \ \bar{f}^*_2 = f_2(\boldsymbol{x}^*_{f_2}),$$

so $\bar{\Theta} = (\bar{f}^*_1, \bar{f}^*_2) - \mathbb{R}^2_{++}$. In Figure 4(a), referential points are uniformly generated on segment union $[(\bar{f}^*_1 - \alpha_1, \bar{f}^*_2), (\bar{f}^*_1, \bar{f}^*_2)] \cup [(\bar{f}^*_1, \bar{f}^*_2 - \alpha_2), (\bar{f}^*_1, \bar{f}^*_2)]$. In Figure 4(b), referential points are uniformly generated on segment $[(\bar{f}^*_1 - \alpha_1, \bar{f}^*_2), (\bar{f}^*_1, \bar{f}^*_2 - \alpha_2)]$. Here $\alpha_1, \alpha_2 > 0$ are proper positive numbers.

In Algorithm SNSM, θ is generated in $\bar{\Theta}'$, Figure 4(a) depicts the strategy of generating referential points on the upper boundary of $\bar{\Theta}'$, while Figure 4(b) depicts the referential points generated on a random line segment in $\bar{\Theta}'$.

When $m > 2$, the line segment becomes simplex, we can use the systematic method introduced in the previous subsection to generate referential points.



(a) θ on upper boundary (b) θ on hyperplane

Figure 3: Generating referential points θ for Algorithm NSM.

5.3 Global optimization solver

The global optimization solver plays an important role in the proposed algorithms. If Problem (2.1) is convex, which leads Problems (3.1) and (4.2) convex, then they can be efficiently solved using local or global optimization solvers; otherwise, we have to use nonconvex solvers, such as the global quasisecant method [27] and hybrid global optimization method [24], to tackle them. In our implementation, we simply use functions in the MATLAB optimization toolbox, such as *fmincon*, *fminsearch* and *ga*, to solve Problems (3.1) and (4.2).

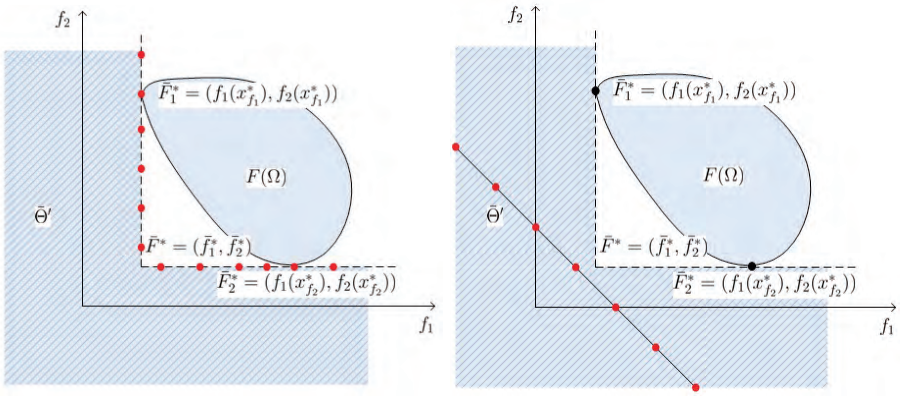
6 Numerical Experiments

In this section, we first present some illustrative examples to demonstrate the numerical performance of the proposed algorithms, then compare the proposed algorithms with two typical heuristic multi-objective optimization solvers: NSGAII [5] and MOEA/D [38]. All the numerical experiments are implemented in an environment of MATLAB(2010a) installed on an ACER ASPIRE 4730Z laptop with a 2G RAM and a 2.16GB CPU.

6.1 Illustrative examples

Problem SCH [5] in Table 1 is a one dimensional convex multi-objective problem. Its efficient solution set is $[0, 2]$, Figure 5(a) shows its image set and Pareto frontier. Solving Problem SCH using EWSM, we can obtain results showing in Figure 6. Among them, $\lambda \in \bar{\Lambda}$ for Figure 6(a) is uniformly chosen on the line segment $\lambda_1 + \lambda_2 = 1$ ($\lambda_1, \lambda_2 \geq 0$); while $\lambda \in \bar{\Lambda}$ for Figure 6(b) is randomly chosen. Note that λ and Pareto points are actually in different spaces, but in Figure 6 (so as the following figures), we draw them together to demonstrate the relationship between λ and Pareto frontier. From Figure 6, we can observe that Problem SCH is perfectly solved by EWSM (the numerical performance of uniformly chosen $\bar{\Lambda}$ is better than the randomly chosen one), and each point in Pareto frontier corresponds to a $\lambda \in \bar{\Lambda}$.

Problem FON [5] (see Table 1 and Figure 5(b) is a three dimensional nonconvex problem,



(a) θ on upper boundary (b) θ on hyperplane

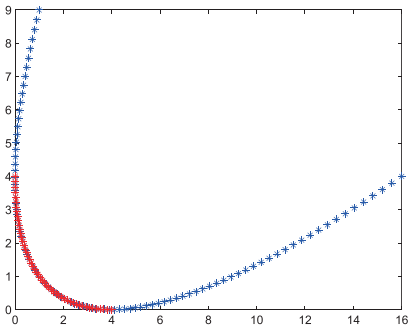
Figure 4: Generating referential points θ for Algorithm SNSM.

Table 1: Multi-objective test problems.

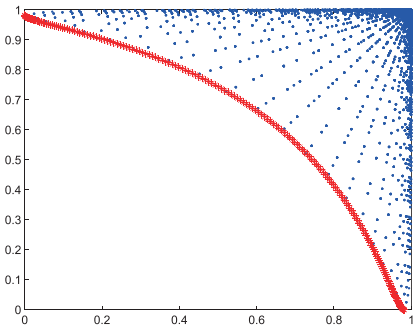
Pro.	n	Variable bounds	Objective functions	Optimal solutions	Convexity
SCH	1	$[-5, 10]$	$f_1(x) = x^2$ $f_2(x) = (x-2)^2$	$x \in [0, 2]$	convex
FON	3	$[-4, 4]$	$f_1(x) = 1 - \exp(-\sum_{i=1}^{3}(x_i - \frac{1}{\sqrt{3}})^2)$ $f_2(x) = 1 - \exp(-\sum_{i=1}^{3}(x_i + \frac{1}{\sqrt{3}})^2)$	$x_1 = x_2 = x_3$ $\in [-1/\sqrt{3}, 1/\sqrt{3}]$	nonconvex
KUR	3	$[-5, 5]$	$f_1(x) = \sum_{i=1}^{n-1}(-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2}))$ $f_2(x) = \sum_{i=1}^{n}(\lvert x_i\rvert^{0.8} + 5\sin^3(x_i))$	[5]	nonconvex

its Pareto solutions satisfy $x_1 = x_2 = x_3$, where $x_i \in [-1/\sqrt{3}, 1/\sqrt{3}]$, $i = 1, 2, 3$. Figure 7 demonstrates Problem FON solved by NSM. Among them, in Figure 7(a), referential points θ are uniformly generated on the upper boundary of $\bar{\Theta}$, i.e., $[(-1, 0), (0, 0)] \cup [(0, 0), (0, -1)]$; while in Figure 7(b), θ are uniformly generated on simplex $[(-1, 0), (0, -1)]$. Compare both figures, one can observe that both strategies can obtain perfect approximation of the Pareto frontier. Figure 8 demonstrates Problem FON solved by SNSM, two referential points generating strategies are applied as well. Form Figure 8, SNSM can still get good Pareto frontier approximation, but some non-efficient point appears at both ends of the approximate Pareto frontier. These non-efficient points can be identified and removed using Pareto sorting, which is then depicted in Figure 9.

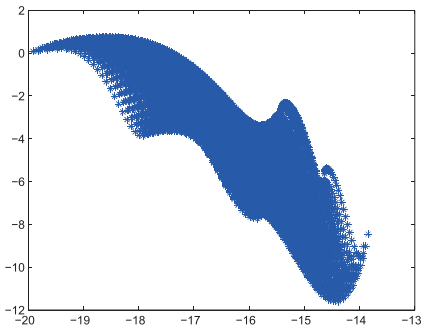
Problem KUR [5] (see Table 1 and Figure 5(c)) is a three dimensional nonconvex problem, its Pareto frontier is disconnected. Figure 10 demonstrates Problem KUR solved by NSM and SNSM. From Figure 10(a), when solving by NSM, there are only a few efficient solutions can be obtained, and different $\theta \in \bar{\Theta}$ may end up with same efficient point. However, when solving using SNSM, as illustrated in Figure 10(b), we can see that the disconnected Pareto frontier of Problem KUR is perfectly simulated and most of the $\theta \in \bar{\Theta}'$ are properly chosen. This reveals that SNSM, although theoretically defective, could numerically performs better than NSM.



(a) Problem SCH



(b) Problem FON



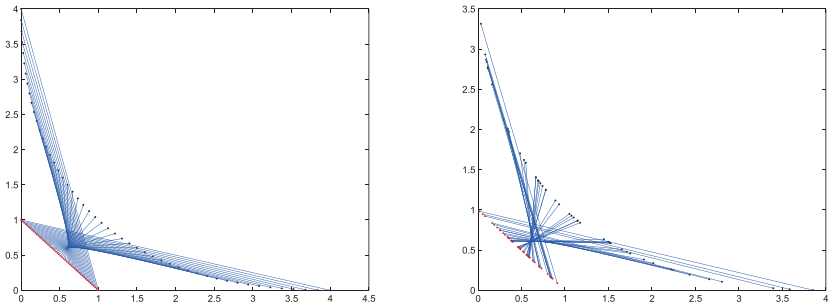
(c) Problem KUR

Figure 5: Objective function value set of Problems SCH, FON and KUR.

6.2 Comparison with MOEA/D and NSGAII

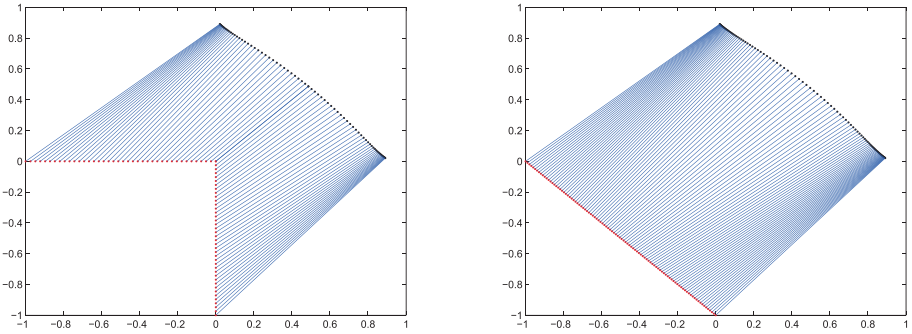
In this subsection, we first introduce MOGA/D [38] and NSGAII [5] as referential methods and then analyze the complexity and numerical performance of EWSM and SNSM by comparing them with MOEA/D and NSGAII on a set of test instances. The reason to choose MOEA/D and NSGAII as referential methods is that MOEA/D is one of the typical decomposition methods for MOPs and NSGAII is the most successful multi-objective genetic algorithm.

MOEA/D is a typical multi-objective optimization method based on evolutionary algorithm and decomposition. It decomposes an MOP into a number of scalar optimization subproblems and optimize them simultaneously. Paper [38] presented three strategies to decompose MOPs: the weighted sum approach, Tchebycheff approach and boundary intersection approach. Our proposed methods are similar with MOEA/D in decomposing MOP but different in treating the corresponding scalar (sigle-objective) optimization problems. NSGAII is without any doubt one of the most successful multi-objective genetic algorithm in the last decade. It introduced a nondominated sorting strategy, this strategy decreases the complexity of nondominated sorting from $O(MN^3)$ to $O(MN^2)$ and proposes a good approach to balance nonelitism and diversity of obtained solutions. In the last decade, NSGAII gains a large amount of citations and applications for its robustness and efficiency in



(a) $\bar{\Lambda}$ uniformly chosen. (b) $\bar{\Lambda}$ randomly chosen.

Figure 6: Solving Problem SCH using Algorithm 1.



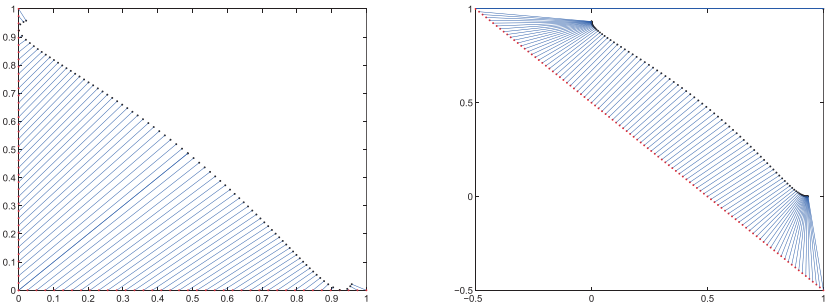
(a) λ on upper boundary (b) λ on simplex

Figure 7: Problem FON solved by NSM.

solving MOPs.

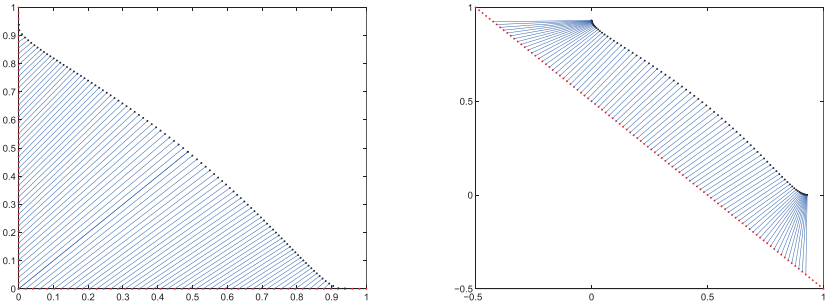
Test instances used in this subsection are SCH, FON, KUR from [5] and ZDT1∼4, ZDT6, DTLZ1, DTLZ2 from [38]. Codes for MOEA/D and NSGAII are taken from Yarpiz (`www.yarpiz.com`). As well as MOEA/D [38] and NSGAII [5], the population size N is set to be 100 for 2-objective test instances and 150 for 3-objective test instances. The maximal number of generations is set to be 50 for 2-objective problems and 100 for 3-objective ones. The comparison is respect to two factors: numerical performance and complexity. Numerical performance is observed through depicting Pareto frontiers obtained by different methods in one figure, and complexity is measured by number of function value evaluation and time consumption.

Figure 11 depicts obtained approximate Pareto frontiers of 2-objectives Problems SCH, FON, KUR, ZDT1∼4 and ZDT6 solved by SNSM, MOEA/D and NSGAII, respectively. For Problem SCH (see Figure 11(a)), all the algorithms reach the real Pareto frontier, the diversity of solutions obtained by NSGAII and SNSM is better than that of MOEA/D. For Problem FON (see Figure 11(b)), Pareto frontier obtained by SNSM is much better than that obtained by MOGA/D and NSGAII not only in elitism but also in diversity. For Problem KUR (see Figure 11(c)), solutions obtained by SNSM perfectly simulated the disconnected



(a) λ on upper boundary (b) λ on simplex

Figure 8: Problem FON solved by SNSM.

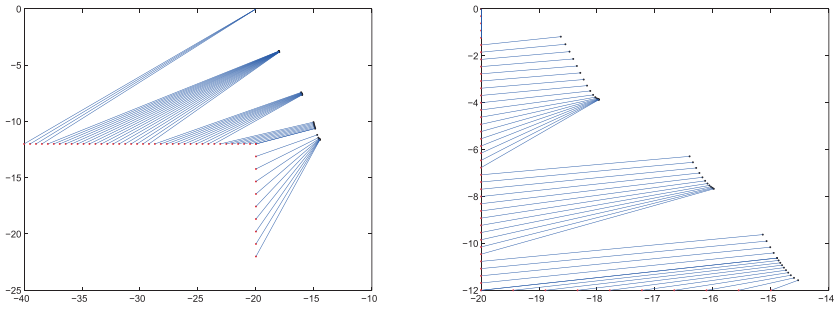


(a) λ on upper boundary (b) λ on simplex

Figure 9: Problem FON solved by SNSM with Pareto sorting.

Pareto frontier, while MOEA/D is not good at elitism and NSGAII only concentrate its solutions in the middle section. For Problem ZDT1 (see Figure 11(d)), all three algorithms performs evenly in elitism, but SNSM and NSGAII are better than MOEA/D in diversity. For Problem ZDT2 (see Figure 11(e)), MOEA/D and SNSM are neck and neck both in diversity and elitism, but both perform better than NSGAII. For Problem ZDT3 (see Figure 11(f)), although MOEA/D and SNSM perform better than NSGAII in elitism, solutions obtained by NSGAII simulate the Pareto frontier more comprehensively, and solutions obtained by SNSM are extremely dense in some area. For Problem ZDT4 (see Figure 11(g)), SNSM obtains solutions with better elitism, but MOEA/D and NSGAII obtain solutions with better diversity and distributed more comprehensively. For Problem ZDT6 (see Figure 11(h)), NSGAII performs better than both SNSM and MOEA/D not only in elitism but also in diversity, and again solutions obtained by SNSM concentrate in some points.

Figure 12 demonstrates numerical results of Problem DTLZ1 and DTLZ2 solved by MOEA/D, NSGAII and SNSM, respectively. From Figure 12(a), 12(b) and 12(c), MOEA/D and SNSM obtain solutions which can normally simulate the real Pareto frontier, while NSGAII almost fail to solve this problem. Comparing with MOEA/D and SNSM, one can observe that solutions obtained by SNSM are better in both diversity and elitism than



(a) NSM (b) SNSM

Figure 10: Problem KUR solved by NSM and SNSM.

MOEA/D. From Figure 12(d),12(e) and 12(f), it is obviously to observe that SNSM performs better than MOEA/D and NSGAII both in elitism and diversity.

Histogram 13 illustrates the complex comparison of SNSM, NSGAII and MOEAD for solving these problems. For 2-objective problems, the number of function evaluations of SNSM for most problems is larger than NSGAII and MOEAD, but it consumes less time than the other two for all problems. This reveals that SNSM is more efficient than the other two respect to number of function evaluations. For 3-objective problems, the number of function evaluations of SNSM is larger than the other two algorithm, its time consumption is only slightly larger. Actually, the reason for the number of function evaluations being large is that we used genetic algorithm to solve the scalar problems for Problem ZDT1∼ZDT4, ZDT6, DTLZ1 and DTLZ2, if one can substitute genetic algorithm by other more efficient optimization solvers, the number of function evaluations could decrease dramatically.

To be summarized from Figure 11, 12 and Histogram 13, Algorithm SNSM (or EWSM for convex problems) could performs not worse and even better than MOEA/D and NSGAII. One reason for this advantage of SNSM is that its subproblem (Problem (3.1) or (4.2)) is solved using deterministic methods which is normally more accurate and faster than metaheuristic methods, and for regular problems, uniformly generated parameters (λ for EWSM or θ for SNSM) usually yield diversely distributed Pareto frontier. One may think that globally solving the subproblem is already a difficult task, let alone there are many subproblems need to be solved in SNSM. It is true that the problem is not suitable to be solved by SNSM if its subproblems are difficult to be globally solved. Another doubt about SNSM is that it should be inefficient because of these time consuming subproblems. This is not the case, because subproblem of SNSM are solve using deterministic methods which are quite efficient, so it will not take to much time for every subproblems. In fact, according to our numerical tests, the time consumed by solving a single subproblem of SNSM is generally less than that consumed by a generation of MOEA/D and NSGAII.

6.3 Numerical test and comparisons using CEC'09

In this subsection, we compare the numerical performance of SNSM with the methods proposed in the special session on performance assessment of unconstrained/bound constrained multi-objective optimization algorithms at CEC'09. There are 13 algorithms submitted to the special session:

(1) MOEAD [39];

(2) GDE3 [15];

(3) MOEADGM [3];

(4) MTS [35];

(5) LiuLiAlgorithm [18];

(6) DMOEADD [19];

(7) NSGAIILS [33];

(8) OWMOSaDE [12]

(9) ClusteringMOEA [36]

(10) AMGA [34]

(11) MOEP [30]

(12) DECMOSA-SQP [37];

(13) OMOEAII [9].

Test problems applied in this subsection are quoted from [40], they are used as benchmarks in CEC'09. Figure 14 and 15 illustrate objective function value sets and real Pareto frontiers of these test problems (the figure of Problem 2 is ignored here since it is similar to Problem 1). Among these test problems, Problems 1-7 have two objective functions, whereas Problems 8-10 have three objective functions. The Pareto solutions of Problem 5, 6 and 9 are disconnected, while the others are connected.

In order to evaluate the numerical performance, we use the performance metric IGD proposed in [40]. Suppose that P^* is a set of uniformly distributed points along the Pareto frontier. Let A be a set of solutions obtained by a certain solver. Then, the average distance from P^* to A is defined as

$$IGD(A, P^*) = \frac{\sum_{v \in P^*} d(v, A)}{|P^*|},$$

where $d(v, A)$ is the minimum Euclidean distance between v and the points in A, i.e.,

$$d(v, A) = \min_{y \in A} \parallel v - y \parallel .$$

In fact, P^* represents a sample set of the real Pareto frontier, if $|P^*|$ is large enough to approximate the Pareto frontier very well, $IGD(A, P^*)$ could measure both the diversity and convergence of A. A smaller $IGD(A, P^*)$ means the set A is closer to the real Pareto frontier and has better diversity.

In order to keep consistent with the final report of CEC'09 [41], in the implementation of SNSM, we compute 100 efficient solutions for problems with two objectives and 150 for problems with three objectives, the number of function evaluations is less than $300,000$. The numerical performance evaluated by IGD are illustrated in Table 2.

From Table 2, for Problem 1, 2 and 3, IGD evaluations rank at first, fifth and first, respectively. This means that SNSM performs better than other algorithms in solving Problem 1 and 3. When solving Problem 2, although the IGD evaluation of SNSM ranks at

Table 2: The numerical performance evaluated by IGD.

rank	UF1		UF2		UF3	
1	**SNSM**	**0.00381**	MTS	0.00615	**SNSM**	**0.00380**
2	MOEAD	0.00435	MOEADGM	0.0064	MOEAD	0.00742
3	GDE3	0.00534	DMOEADD	0.00679	LiuliAlgorithm	0.01497
4	MOEADGM	0.0062	MOEAD	0.00679	DMOEADD	0.03337
5	MTS	0.00646	**SNSM**	**0.0072**	MOEADGM	0.049
6	LiuLiAlgorithm	0.00785	OWMOSaDE	0.0081	MTS	0.0531
7	DMOEADD	0.01038	GDE3	0.01195	ClusteringMOEA	0.0549
8	NSGAIILS	0.01153	LiuLiAlgorithm	0.0123	AMGA	0.06998
9	OWMOSaDE	0.0122	NSGAIILS	0.01237	DECMOSA-SQP	0.0935
10	ClusteringMOEA	0.0299	AMGA	0.01623	MOEP	0.099
11	AMGA	0.03588	MOEP	0.0189	OWMOSaDE	0.103
12	MOEP	0.0596	ClusteringMOEA	0.0228	NSGAIILS	0.10603
13	DECMOSA-SQP	0.07702	DECMOSA-SQP	0.02834	GDE3	0.10639
14	OMOEAII	0.08564	OMOEAII	0.03057	OMOEAII	0.27141
rank	UF4		UF5		UF6	
1	**SNSM**	**0.01833**	MTS	0.01489	MOEAD	0.00587
2	MTS	0.02356	GDE3	0.03928	**SNSM**	**0.00976**
3	GDE3	0.0265	AMGA	0.09405	MTS	0.05917
4	DECMOSA-SQP	0.03392	LiuLiAlgorithm	0.16186	DMOEADD	0.06673
5	AMGA	0.04062	DECMOSA-SQP	0.16713	OMOEAII	0.07338
6	DMOEADD	0.04268	OMOEAII	0.1692	ClusteringMOEA	0.0871
7	MOEP	0.0427	MOEAD	0.18071	MOEP	0.1031
8	LiuLiAlgorithm	0.0435	MOEP	0.2245	DECMOSA-SQP	0.12604
9	OMOEAII	0.04624	ClusteringMOEA	0.2473	AMGA	0.12942
10	MOEADGM	0.0476	DMOEADD	0.31454	LiuLiAlgorithm	0.17555
11	OWMOSaDE	0.0513	OWMOSaDE	0.4303	OWMOSaDE	0.1918
12	NSGAIILS	0.0584	NSGAIILS	0.5657	GDE3	0.25091
13	ClusteringMOEA	0.0585	**SNSM**	**0.7032**	NSGAIILS	0.31032
14	MOEAD	0.06385	MOEADGM	1.7919	MOEADGM	0.5563
rank	UF7		UF8		UF9	
1	MOEAD	0.00444	MOEAD	0.0584	**SNSM**	**0.0339**
2	LiuLiAlgorithm	0.0073	DMOEADD	0.06841	DMOEADD	0.04896
3	MOEADGM	0.0076	LiuLiAlgorithm	0.08235	NSGAIILS	0.0719
4	DMOEADD	0.01032	NSGAIILS	0.0863	MOEAD	0.07896
5	**SNSM**	**0.01063**	OWMOSaDE	0.0945	GDE3	0.08248
6	MOEP	0.0197	MTS	0.11251	LiuliAlgorithm	0.09391
7	NSGAIILS	0.02132	**SNSM**	**0.1707**	OWMOSaDE	0.0983
8	ClusteringMOEA	0.0223	AMGA	0.17125	MTS	0.11442
9	DECMOSA-SQP	0.02416	OMOEAII	0.192	DECMOSA-SQP	0.14111
10	GDE3	0.02522	DECMOSA-SQP	0.21583	MOEADGM	0.1878
11	OMOEAII	0.03354	ClusteringMOEA	0.2383	AMGA	0.18861
12	MTS	0.04079	MOEADGM	0.2446	OMOEAII	0.23179
13	AMGA	0.05707	GDE3	0.24855	ClusteringMOEA	0.2934
14	OWMOSaDE	0.0585	MOEP	0.423	MOEP	0.342
rank	UF10					
1	MTS	0.15306				
2	**SNSM**	**0.2382**				
3	DMOEADD	0.32211				
4	AMGA	0.32418				
5	MOEP	0.3621				
6	DECMOSA-SQP	0.36985				
7	ClusteringMOEA	0.4111				
8	GDE3	0.43326				
9	LiuLiAlgorithm	0.44691				
10	MOEAD	0.47415				
11	MOEADGM	0.5646				
12	OMOEAII	0.62754				
13	OWMOSaDE	0.743				
14	NSGAIILS	0.84468				

fifth, the difference with the first four algorithms are tiny, only in a precision of 10^{-3}. Additionally, the accuracy of IGD evaluations, which are 10^{-2}, reveals that Problem 1, 2 and 3 are prefect solved by SNSM. Figure 16 illustrates the obtained Pareto frontier of Problem 1 and 3, comparing with the real Pareto frontiers illustrated in Figure 15, we can conclude that these two problems are perfectly solved by SNSM.

For Problem 4, the IGD evaluation of SNSM ranks at the first, better than other referential algorithm. But its value is 0.01833 , only in a precision of 0.1, which is not perfect good. This point is also illustrated in Figure 17(a), which shows that the obtained efficient points are not extremely accurate and uniformly distributed.

SNSM is failed at solving Problem 5, one possible reason is that the Pareto frontier of Problem 5 consists of some isolated points, which is not suitable for SNSM.

The Pareto frontier of Problem 6 is disconnected, consist of two line segments and a point (See Figure 15). Figure 17(b) demonstrate the obtained Pareto frontier using SNSM, from the figure, we can observe that the real Pareto frontier is well simulated, points at the Pareto frontier have high accuracy and distribute evenly. In fact, from Table 2, IGD evaluation of SNSM for Problem 6 is 0.00976, accurate to 10^{-2}, ranks at the second.

For Problem 7 whose obtained Pareto frontier is illustrated in Figure 18(a), the IGD evaluation is 0.1063, ranks at the fifth. One interesting phenomenon showed in Figure 18(b) is that the lower part of the obtained Pareto frontier is very regular, but the upper part looks disorder. This may relate to the objective functions and option of θ.

For Problem 8, 9 and 10, IGD evaluation for SNSM are 0.1707, 0.03393 and 0.2382, rank at the seventh, first and second, respectively. The obtained Pareto frontier of Problem 9 is presented in Figure 18(b), Problem 8 and 10 are not presented since they are almost failed to simulate the real Pareto frontier. It is not uncommon that IGD evaluation for these three problems are not as small as others, because they all have three objective functions, which makes their Pareto frontiers surfaces. This not only increases the complexity of objective function of Problem (4.2), but also dramatically increases the amount of calculation since we have to work on much more points. In fact, even for the best MOEA, like MOEAD for Problem 8 (0.0584), DMOEADD for Problem 9 (0.04896) and MTS for Problem 10 (0.15306), the IGD evaluation is far from good enough.

7 Conclusion

This paper proposed population-based linear and nonlinear scalarization methods for MOPs. Scalarization is an important type of strategy to handle MOPs. The previous research mainly focuses on scalar techniques, while this paper contributes to generalizing the scalarization methods to population-based ones. We first extended the weighted sum method to a population-based case which has good theoretical properties and numerical performances for convex MOPs, but fails to solve nonconvex MOPs. In order to handle nonconvex MOPs, we designed a nonlinear scalar technique which transforms an MOP to a nonlinear scalar optimization problem. It can be proved that, in some conditions, the global optimal solution of the nonlinear optimization problem must be an efficient solution of the original multi-objective problem. Based on this property, a nonlinear scalarization method and a slack variation of it were proposed. A wide range of numerical tests were presented. First of all, numerical performance of the proposed methods were illustrated by some academic multi-objective optimization benchmarks; then numerical comparisons among the proposed methods and two typical multi-objective optimization methods MOEA/D and NSGAII were made; finally the proposed method were applied to solve CEC'09 test instances and the results were compared with 13 referential algorithms proposed in CEC'09. Numerical tests

show that methods proposed in this paper are able to solve MOPs with promising elitism and diversity.

There are two critical points that need to be tackled in our future work on this subject. First, a global optimization method plays a very important role in the proposed methods with a fast deterministic global optimization method dramatically increasing efficiency. In this paper, subproblems are solved directly using a global optimization method, but for MOPs whose objective functions are extremely complicated, this strategy may not work. Therefore, some metaheuristic strategies should be introduced to handle subproblems. Second, the distribution of predetermined scalarization parameters corresponds to the distribution of obtained solutions. For regular problems, uniformly generated scalarization parameters usually yields a diverse distribution of solutions, but for irregular problems, many scalarization parameters may correspond with one solution which destroys the diversity of the obtained solutions. In this case, a self-adaptive strategy for generating scalarization parameters should be developed.

References

[1] R. Burachik and K. Yalcin, A new scalarization technique and new algorithms to generate Pareto fronts, *SIAM J. Optim.* 27 (2017) 1010–1034.

[2] C.E. Gontijo, S. Gisele, C. Edgard and T. Ricardo, Subpermutation-based evolutionary multiobjective algorithm for load restoration in power distribution networks, *IEEE Trans. Evol. Comput.* 20 (2016) 546–562.

[3] C.M. Chen, Y.P. Chen and Q.F. Zhang, Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization, *Evolutionary Computation, 2009. CEC'09. IEEE Congress on* (2016), 209–216.

[4] D. Indraneel and J. Dennis, Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM J. Optim.* 8 (1998) 631–657.

[5] K. Deb, A. Pratap, S. Agarwal and T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. Evol. Comput.* 6 (2002) 182–197.

[6] E. Gabriele, An adaptive scalarization method in multiobjective optimization, *SIAM J. Optim.* 19 (2009) 1694–1718.

[7] E. Gabriele. *Multicriteria Optimization*, Springer Science & Business Media, 2006.

[8] H. Fazlollahtabar and M. Saidi-Mehrabad, Optimizing multi-objective decision making having qualitative evaluation, *J. Ind. Manag. Optim.* 11 (2015), 747–762.

[9] S. Gao, B. Xiao, L. Zhang, Y.L. Shi, X. Tian, Y. Yang, H.Q. Long, X.Q. Yang and D.P. Yu, Z. Yan, An orthogonal multi-objective evolutionary algorithm with lower-dimensional crossover, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 1959–1964.

[10] A. Geoffrion, Proper efficiency and the theory of vector maximization, *J. Math. Anal. Appl.* 22 (1968) 618–630.

[11] R. Hartley, On cone-efficiency, cone-convexity and cone-compactness, *SIAM J. Appl. Math.* 34 (1978) 211–222.

[12] V.L. Huang, S.G Zhao, R. Mallipeddi and P. Suganthan, Multi-objective optimization using self-adaptive differential evolution algorithm, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 190–194.

[13] S. Karlin, Mathematical methods and theory in games, programming and economics, *Physics Today* 13 (1960) 190–194.

[14] T. Koopmans, *Activity Analysis of Production and Allocation*, Wiley New York, 1960.

[15] S. Kukkonen and J. Lampinen, Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp.1943–1950.

[16] M. Laumanns, L. Thiele and E. Zitzler, An efficient adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, *European J. Oper. Res.* 169 (2006), 932–942.

[17] M. Laumanns, L. Thiele and E. Zitzler, An efficient adaptive parameter variation scheme for metaheuristics based on the epsilon-constraint method, *European J. Oper. Res.* 169 (2006) 932–942.

[18] H.B. Liu and X.Q. Li. The multiobjective evolutionary algorithm based on determined weight and sub-regional search, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 1928–1934.

[19] M.Z. Liu, X.F. Zou, Y. Chen and Z.J. Wu, Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 2913-2918.

[20] C.Y. Liu, Z.H. Gong, K.L. Teo and E. Feng, Multi-objective optimization of nonlinear switched time-delay systems in fed-batch process, *Appl. Math. Model.* 40 (2016) 10533–10548.

[21] C.Y. Liu, Z.H. Gong, K.L. Teo, R. Loxton and E. Feng, Bi-objective dynamic optimization of a nonlinear time-delay system in microbial batch process, *Optim. Lett.* 12 (2018) 1249–1264.

[22] C.Y. Liu, Z.H. Gong, K.L. Teo, J. Sun and L. Caccetta, Robust parameter estimation for constrained time-delay systems with inexact measurements, *Nonlinear Anal. Hybrid Syst.* 25 (2017) 1–20.

[23] C.Y. Liu, M.J. Han, Z.H. Gong and K.L. Teo, Robust multi-objective optimal switching control arising in 1,3-propanediol microbial fed-batch process, *J. Ind. Manag. Optim.* (2019), DOI:10.3934/jimo.2019113, 1-20.

[24] Q. Long and C.Z. Wu, A hybrid method combining genetic algorithm and Hooke-Jeeves method for constrained global optimization, *J. Ind. Manag. Optim.* 10 (2014) 1279–1296.

[25] Q. Long, A constraint handling technique for constrained multi-objective genetic algorithm, *Swarm and Evolutionary Computation* 15 (2014) 66–79.

[26] Q. Long, C.Z. Wu, T.W. Huang and X.Y. Wang, A genetic algorithm for unconstrained multi-objective optimization, *Swarm and Evolutionary Computation* 22 (2015) 1–14.

[27] Q. Long, C.Z. Wu, X.Y. Wang and Z.Y. Wu, A modified quasisecant method for global optimization, *Appl. Math. Model.* 51 (2017) 21–37.

[28] K. Miettinen, *Nonlinear Multiobjective Optimization* International Series in Operations Research and Management Science, vol.12, Kluwer Academic Publishers, Dordrecht, 1999.

[29] M.M. Robat, P. Tehrani and M. Bennis, Energy-efficient power allocation in OFDMA D2D communication by multiobjective optimization, *IEEE Wirel. Commu. Lett.* 5 (2016) 668–671.

[30] B.Y. Qu and P. Suganthan, Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 2934–2939.

[31] R.L. Álvaro, V.R. Miguel and D. González-Álvarez, A hybrid multiobjective memetic metaheuristic for multiple sequence alignment, *IEEE Trans. Evol. Comput.* 20 (2016) 499–514.

[32] B. Rudloff, F. Ulus and R. Vanderbei, A parametric simplex algorithm for linear vector optimization problems, *Math. Program.* 163 (2017) 213–242.

[33] K. Sindhya, A. Sinha, K. Deb and K. Miettinen, Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 2919–2926.

[34] S. Tiwari, G. Fadel, P. Koch and K. Deb, Performance assessment of the hybrid archive-based micro genetic algorithm (AMGA) on the CEC09 test problems, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 1935–1942.

[35] L.Y. Tseng and C. Chen, Multiple trajectory search for unconstrained/constrained multi-objective optimization, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 1951–1958.

[36] Y.P Wang, C.Y Dang, H.C. Li, L.X. Han and J.X. Wei, A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 2927–2933.

[37] A. Zamuda, J. Brest, B. Boskovic and V. Zumer, Differential evolution with self-adaptation and local search for constrained multiobjective optimization, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 195–202.

[38] Q.F. Zhang and H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, *IEEE Trans. on Evol. Comput.* 11 (2007) 712–731.

[39] Q.F. Zhang, W.D. Liu and H. Li, The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, 2009, pp. 203–208.

[40] Q.F. Zhang, A. Zhou, S.Z. Zhao, P. Suganthan, W.D. Liu and S. Tiwari, in: Multiobjective optimization test instances for the CEC 2009 special session and competition, *University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report*, 2008.

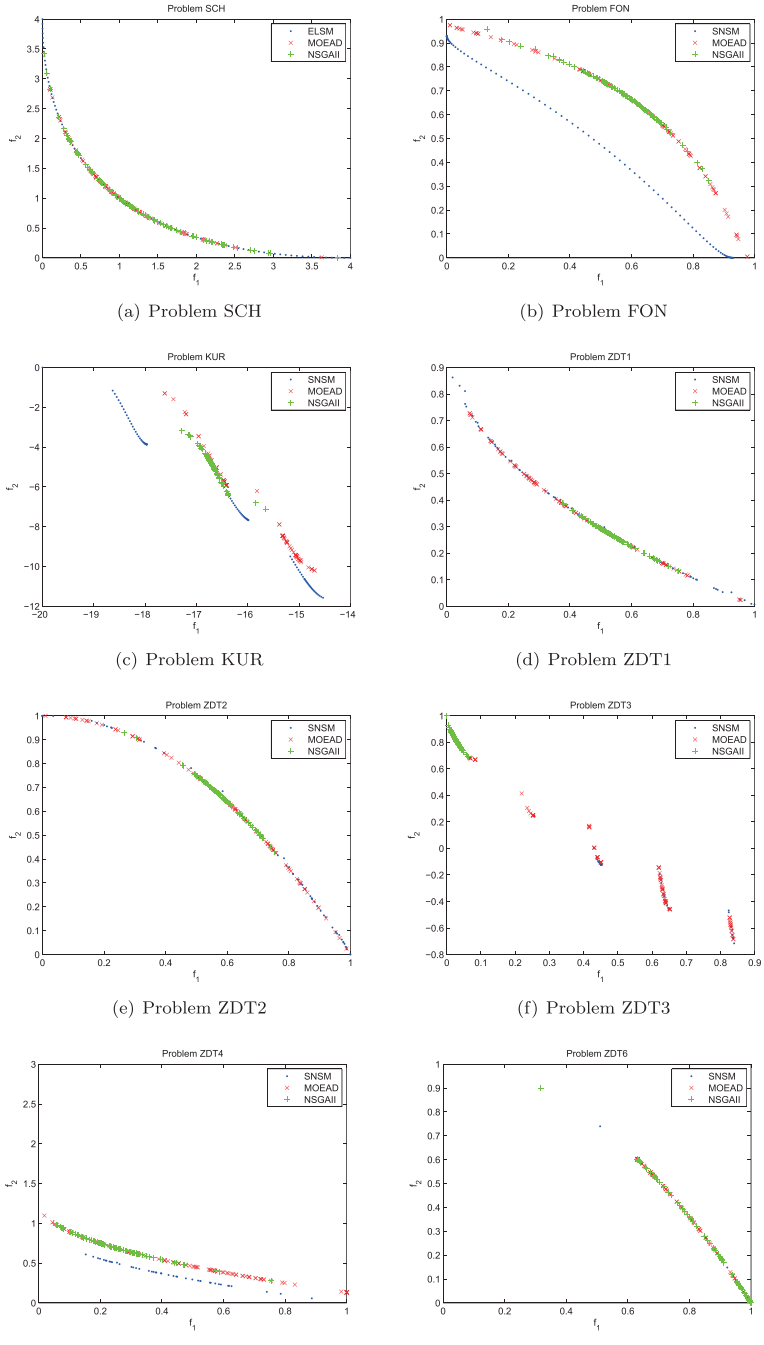
[41] Q.F. Zhang and P. Suganthan, Final report on CEC'09 MOEA competition, in: *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, (2009).

Manuscript received 10 June 2018
revised 22 December 2018
accepted for publication 22 May 2019

Qiang Long
School of Science, Southwest University of Science and Technology
Mianyang 621010, China
E-mail address: longqiang@swust.edu.cn

Lin Jiang
School of mathematics and statistics, Curtin University
Perth 6145 WA, Australia
E-mail address: lydiajianglin@gmail.com

Guoquan Li
School of Mathematical Sciences, Chongqing Normal University
Chongqing 401131, China
E-mail address: ligq@cqnu.edu.cn



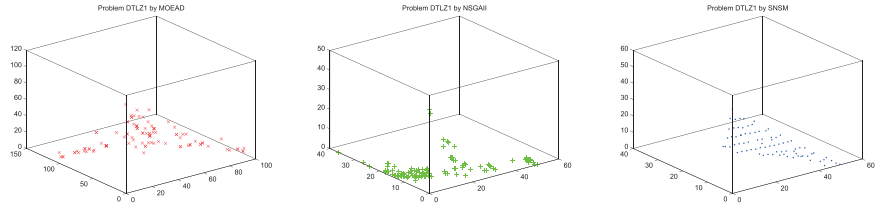
(a) Problem SCH (b) Problem FON

(c) Problem KUR (d) Problem ZDT1

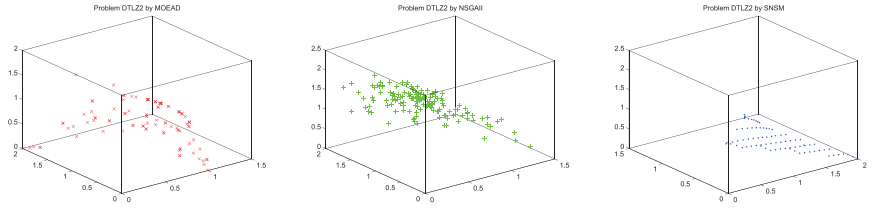
(e) Problem ZDT2 (f) Problem ZDT3

(g) Problem ZDT4 (h) Problem ZDT6

Figure 11: Numerical performance for 2-objective problems.

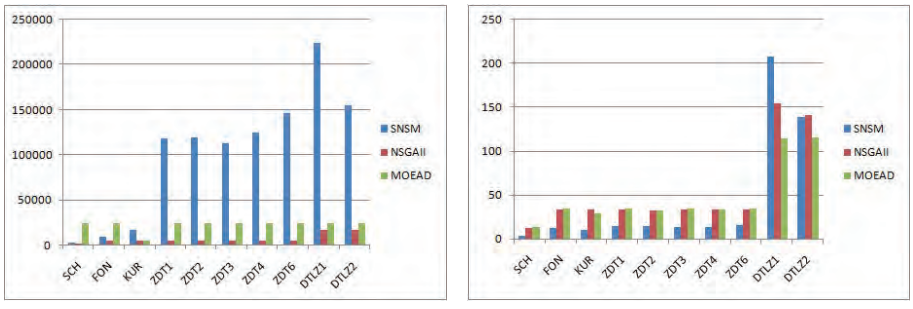


(a) Problem DTLZ1 by MOEAD (b) Problem DTLZ1 by NSGAII (c) Problem DTLZ1 by SNSM



(d) Problem DTLZ2 by MOEAD (e) Problem DTLZ2 by NSGAII (f) Problem DTLZ2 by SNSM

Figure 12: Numerical performance for 3-objective problems.



(a) Number of function evaluation (b) Time consumption

Figure 13: Comparison respect to time consumption and number of function value evaluation.

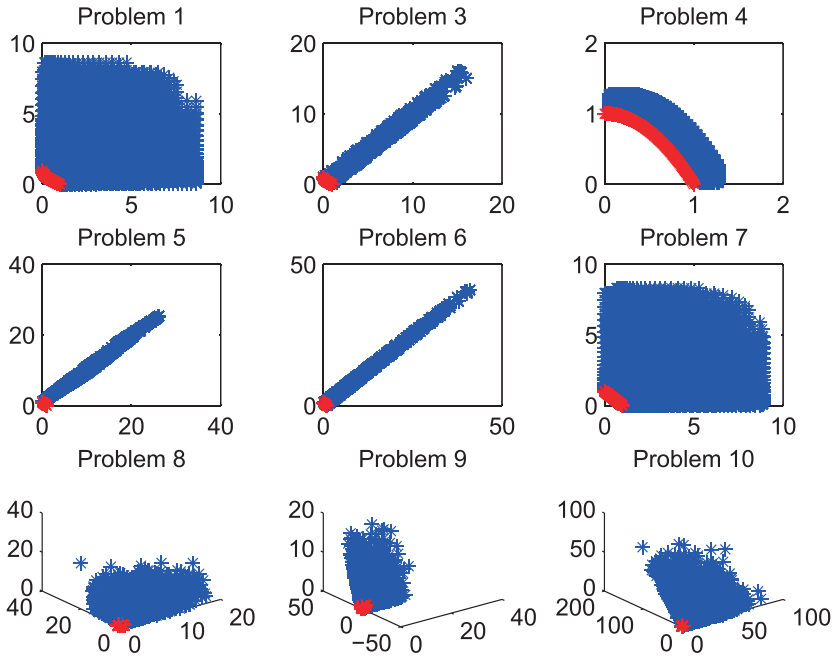


Figure 14: Objective function value space for test problems.

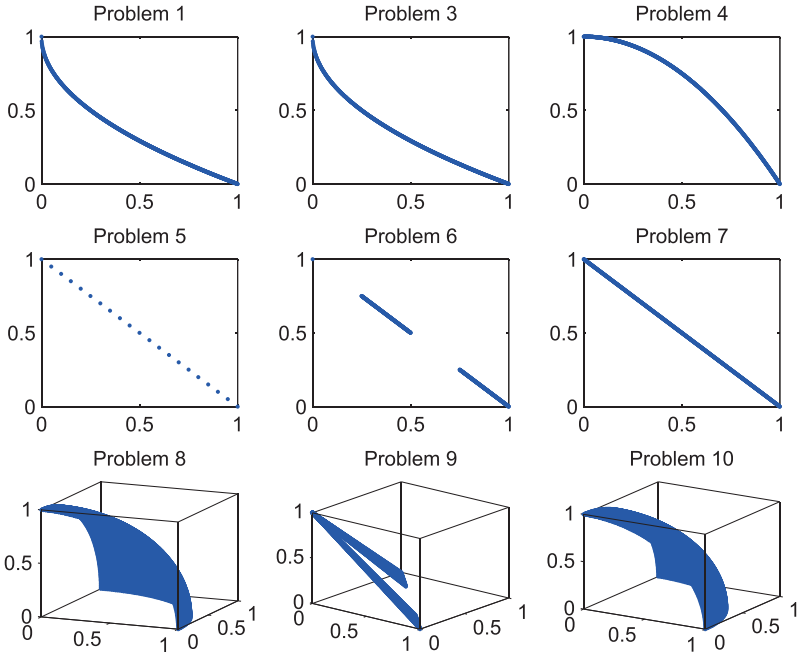
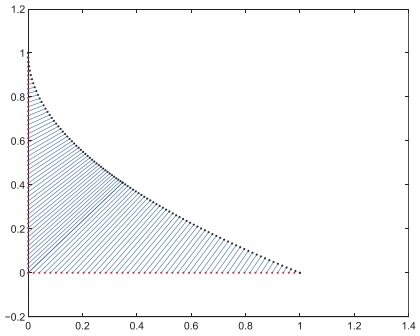
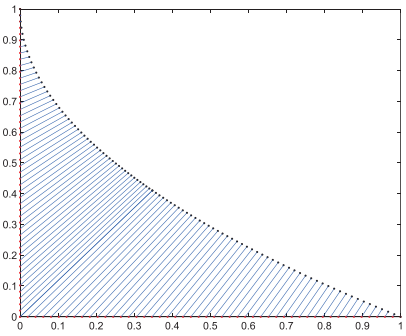


Figure 15: Real Pareto frontiers for test problems.

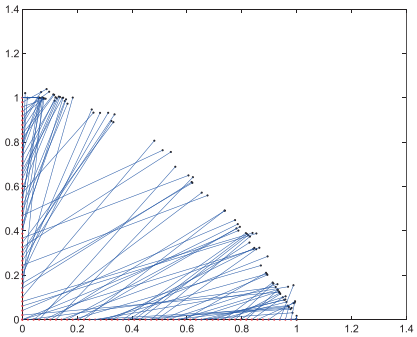


(a) Problem 1

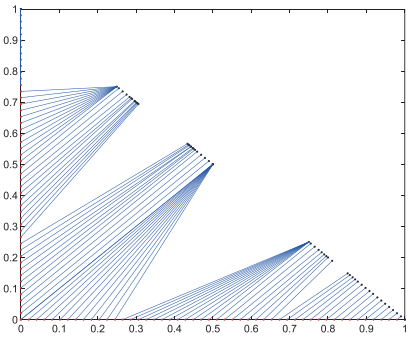


(b) Problem 3

Figure 16: Solving Problem 1 and 3 using SNSM.

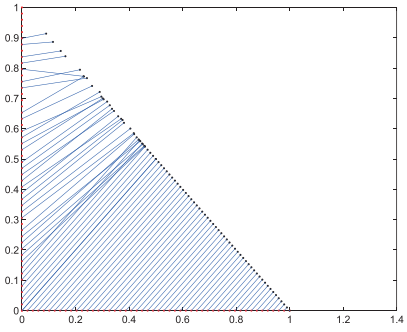


(a) Problem 4

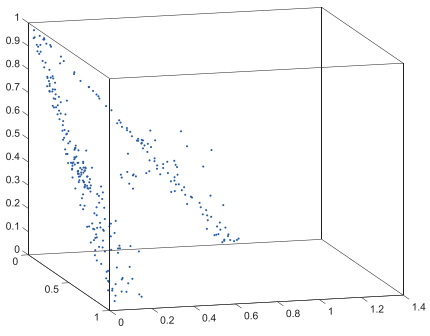


(b) Problem 6

Figure 17: Solving Problem 4 and 6 using SNSM.



(a) Problem 7



(b) Problem 9

Figure 18: Solving Problem 7 and 9 using SNSM.