Filomat 33:7 (2019), 2149–2179
https://doi.org/10.2298/FIL1907149D

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

# Efficient Algorithms for Solving Nonlinear Fractional Programming Problems

**Azam Dolatnezhadsomarin[a], Esmaile Khorram[a], Latif Pourkarimi[b]**

[a]*Department of Mathematics and Computer Sciences, Amirkabir University of Technology, Tehran, Iran*
[b]*Department of Mathematics, Faculty of Sciences, Razi University, Kermanshah, Iran*

**Abstract.** In this paper, an efficient algorithm based on the Pascoletti-Serafini scalarization (PS) approach is proposed to obtain almost uniform approximations of the entire Pareto front of bi-objective optimization problems. Five test problems with convex, non-convex, connected, and disconnected Pareto fronts are applied to evaluate the quality of approximations obtained by the proposed algorithm. Results are compared with results of some algorithms including the normal constraint (NC), weighted constraint (WC), Benson type, differential evolution (DE) with binomial crossover, non-dominated sorting genetic algorithm-II (NSGA-II), and S metric selection evolutionary multiobjective algorithm (SMS-EMOA). The results confirm the effectiveness of the presented bi-objective algorithm in terms of the quality of approximations of the Pareto front and CPU time. In addition, two algorithms are presented for approximately solving fractional programming (FP) problems. The first algorithm is based on an objective space cut and bound method for solving convex FP problems and the second algorithm is based on the proposed bi-objective algorithm for solving nonlinear FP problems. In addition, several examples are provided to demonstrate the performance of these suggested fractional algorithms.

## 1. Introduction

A nonlinear fractional programming (FP) problem minimizes or maximizes a ratio of two functions subject to given constraints. This problem is a powerful tool to formulate various applications of nonlinear programming such as production planning, location analysis, financial and corporate planning, portfolio selection, health care and hospital planning, etc, see [1, 2]. Therefore, this problem has attracted considerable researches because of its varied applications. The nonlinear FP problem is formulated as follows:

$$\begin{aligned} &min \quad \frac{f(x)}{g(x)} \\ &s.t. \quad x \in X = \{x \in \mathbb{R}^n \mid h(x) = (h_1(x), h_2(x), \ldots, h_m(x)) \leqq 0\}, \end{aligned} \tag{1}$$

where functions $f(x)$ and $g(x)$ are positive for each $x \in X$. According to the kind of the functions $f(x)$ and $g(x)$, the problem (1) is classified as follows:

2010 *Mathematics Subject Classification*. Primary 90B50
*Keywords*. Fractional programming problem, Bi-objective optimization problem, Pareto optimal solution, Pascoletti-Serafini scalarization
Received: 01 May 2017; Revised: 30 August 2018; Accepted: 06 February 2019
Communicated by Predrag Stanimirović
*Email addresses:* adolatnejad@gmail.com (Azam Dolatnezhadsomarin), eskhor@aut.ac.ir (Esmaile Khorram), lp_karimi@yahoo.com (Latif Pourkarimi)

1. If $f(x)$ and $g(x)$ are affine, the problem (1) is called a linear FP (LFP) problem.
2. If $f(x)$ and $g(x)$ are quadratic functions, the problem (1) is called a quadratic FP (QFP) problem.
3. If $f(x)$ is a concave function and $g(x)$ is a convex function, the problem (1) is called a concave FP problem.
4. If $f(x)$ is a convex function and $g(x)$ is a concave function, the problem (1) is called a convex FP problem.

In general, the convex FP problem is not a convex problem, but its objective function is strictly quasi-convex. In this problem, if $f(x)$ and $g(x)$ are positive and differentiable over a convex feasible set, the ratio is a pseudoconvex function and as a result, each local minimum of this type of functions is a global minimum [3–5]. Charnes and Cooper [6] showed if the objective function's denominator of the LFP problem has a unique sign in the feasible set, this problem is converted into a linear problem by using a nonlinear variable transformation. Borza et al. [7] used the Charnes and Cooper's technique and solved the LFP problem with interval coefficients in the objective function. Martos and Whinston [8] solved the LFP problem by an adjacent vertex method based on the simplex method. They indicated that linear programming methods could be extended to solve the LFP problems by considering the pseudo-linearity property of linear ratios. In addition, Dorn [9], Swarup [10], Wagner and Yuan [11], Sharma et al. [12], and Pandey and Punnen [13] applied some methods based on the simplex method for solving the LFP problem. Wolf [14], Isbell and Marlow [15], Bitran and Novae [16], and Hasan and Acharjee [17] used another idea to solve the LFP problem. They solved a sequence of auxiliary problems such that solutions of these auxiliary problems converge to a solution of the LFP problem. Tantawy [18, 19] solved the LFP problem by proposing methods based on the conjugate gradient projection for solving nonlinear programming problems with linear constraints. Odior [20] solved the LFP problem by an algebraic approach, which depends on the duality concept and partial fractions.

Swarup [21] examined the QFP problem with linear constraints and extended the Charnes and Cooper's results. He showed that an optimal solution (if it exists) could be obtained from solutions of two associated quadratic programming problems with linear constraints and one quadratic constraint. Beck and Teboulle [22] considered the QFP problem over a possibly degenerate ellipsoid. They introduced an exact convex semidefinite formulation by considering an assumption on the problem's data and presented a simple iterative procedure with converge super linearly to a global solution of this problem. Khurana and Arora [23] studied the QFP problem in which some of its constraints are homogeneous. They presented a new formulation with less number of constraints and a greater number of variables. Zhang and Hayashi [24] considered minimizing a ratio of two indefinite quadratic functions subject to two specific convex quadratic constraints. They used relation between the FP problem and parametric optimization problems and transformed this problem into a univariate equation. To evaluate the function in the equation, a problem of minimizing a nonconvex quadratic function subject to two quadratic constraints is solved by an iterative algorithm. Suleiman and Nawkhass [25] considered the QFP problem with linear constraints and presented an algorithm based on Wolfe's method [26] and a new modified simplex approach. In addition, they [27] considered this problem with a linear denominator and solved it by a new modified simplex method. Sharma and Singh [28] proposed an approach based on simplex techniques to solve the QFP problem with linear constraints such that the objective function's numerator can be decomposed into two linear functions and its denominator is linear. Moreover, Jayalakshmi [29] used the objective fractional separable method based on the simplex method to solve this kind of problem. Nguyen et al. [30] examined minimizing a ratio of quadratic functions subject to a two-sided quadratic constraint. They replaced the objective function by a parametric family of quadratic functions and proposed a semidefinite program approach to solve it.

Mangasarian [31] used Frank and Wolfe's algorithm [32] to solve the convex FP problem with linear constraints. Dinkelbach [33] considered the concave FP problem over a convex feasible set and solved it by considering a sequence of non-linear convex programming problems. Schaible [34] converted the concave FP problem into a parameter-free convex program by applying a generalization of the Charnes and Cooper's variable transformation. Benson [35] studied maximizing of a ratio of a convex function to a convex function, where at least one of the convex functions is a quadratic form on a compact convex

feasible set. He presented a branch and bound algorithm to solve this problem such that it requires solving a sequence of convex optimization problems. Yamamoto and Konno [36] proposed an algorithm for the QFP problem with convex quadratic functions and linear constraints. They solved this problem by combining the classical Dinkelbach method, integer programming approach for solving non-convex quadratic problems, and a standard nonlinear programming solver.

In this paper, a new idea is described to solve the problem (1) and it is shown that there is a relation between this problem and its corresponding bi-objective optimization problem, i.e., a global optimal solution of the problem (1) is a Pareto optimal solution of a bi-objective optimization problem. By using this relation, two algorithms are suggested for approximately solving the problem (1). The first fractional algorithm is proposed to solve convex FP problems and is based on an objective space cut and bound method developed by Shao and Ehrgott [37]. Then, before describing the second fractional algorithm, an algorithm based on the Pascoletti-Serafini scalarization (PS) approach is proposed to generate approximations of the Pareto front of bi-objective optimization problems. In this algorithm, points obtained are distributed almost uniformly on the entire Pareto front. It should be noted that according to the available literature of the FP problems, most of fractional algorithms were applied on linear, quadratic, convex, or concave FP problems or special cases of them. Therefore, the second fractional algorithm based on the proposed bi-objective algorithm is presented for approximately solving convex and non-convex FP problems. In addition, some examples are used to validate each of the proposed algorithms.

The paper is organized in six sections as follows: In Section 2, some preliminaries about multi-objective optimization problems (MOPs) are stated and the PS approach and some of its properties are reviewed. Then, the relation between the problem (1) and its related bi-objective problem is shown. In Section 3, the first fractional algorithm inspirited from the objective space cut and bound method is presented for solving convex FP problems and three examples are carried out to test its performance. An algorithm based on the PS approach is presented for bi-objective problems in Section 4 and its efficiency is demonstrated through five test problems. The second fractional algorithm to solve nonlinear FP problems is proposed in Section 5 and six examples are considered to examine the efficiency of the algorithm. Finally, Section 6 provides some conclusions.

## 2. Preliminaries and basic definitions

An MOP optimizes two or more (conflicting) objective functions under certain constraints at a same time. Formally, the general MOP can be formulated as follows:

$$\begin{aligned} min \quad & f(x) = (f_1(x), f_2(x), \ldots, f_p(x))^T \\ s.t. \quad & x \in X, \end{aligned} \tag{2}$$

where a nonempty set $X \subset \mathbb{R}^n$ denotes the feasible set in the decision space $\mathbb{R}^n$ and a feasible objective space set $Y = f(X) \subset \mathbb{R}^p$ denotes the image of the feasible set in the objective space $\mathbb{R}^p$. In addition, $p$ objective functions $f_i$, $i = 1, \ldots, p$ ($p \geq 2$), are real-valued objective functions over $X$ and therefore, $f : X \to \mathbb{R}^p$ is a vector-valued function. The following partial order relations are used to order the objective space such that for each $y$ and $\hat{y} \in \mathbb{R}^p$:

1. $y < \hat{y}$ means that $y_i < \hat{y}_i$ for all $i = 1, \ldots, p$,
2. $y \leqq \hat{y}$ means that $y_i \leq \hat{y}_i$ for all $i = 1, \ldots, p$,
3. $y \leq \hat{y}$ means that $y \leqq \hat{y}$ but $y \neq \hat{y}$. In this case, it is said that $y$ dominates $\hat{y}$.

**Definition 2.1.** *A feasible solution $\hat{x} \in X$ is a Pareto optimal solution of the problem (2) if there is no other feasible solution $x \in X$ such that $f(x) \leq f(\hat{x})$. If $\hat{x} \in X$ is a Pareto optimal solution, $f(\hat{x})$ is called a non-dominated point.*

The set of all Pareto optimal solutions of the problem (2) is denoted by $X_E$. In addition, the set of all non-dominated points of the problem (2) is called the Pareto front and is indicated by $Y_N$. Note that the goal of solving MOP is obtaining the Pareto front, which is an infinite set in general. Therefore, an approximation of the Pareto front with a finite size can be obtained in practice.

**Definition 2.2.** *A feasible solution $\hat{x} \in X$ is a weakly Pareto optimal solution of the problem (2) if there is no other feasible solution $x \in X$ such that $f(x) < f(\hat{x})$. If $\hat{x} \in X$ is a weakly Pareto optimal solution, $f(\hat{x})$ is called a weakly non-dominated point.*

The set of all weakly Pareto optimal solutions of the problem (2) is denoted by $X_{WE}$. In addition, the set of all weakly non-dominated points of the problem (2) is called the weakly Pareto front and is indicated by $Y_{WN}$.

**Definition 2.3.** *Let $x^i = argmin_{x \in X} f_i(x)$, for $i = 1, \ldots, p$. Then, the ideal point $y^I \in \mathbb{R}^p$ of the problem (2) is defined as $(f_1(x^1), f_2(x^2), \ldots, f_p(x^p))^T$.*

This point is often infeasible because of the presence of conflicting objective functions and is a lower bound of the Pareto front.

**Definition 2.4.** *The nadir point $y^N \in \mathbb{R}^p$ of the problem (2) is defined as $(f_1^N, f_2^N, \ldots, f_p^N)^T$ in which $f_i^N = max_{X_E} f_i(x)$ for $i = 1, \ldots, p$. Furthermore, the $i^{th}$ anchor point $f^i$, $i = 1, \ldots, p$, is expressed as $(f_1(x^i), \ldots, f_i(x^i), \ldots, f_p(x^i))^T$.*

The nadir point gives an upper bound on the Pareto front and only case that the nadir point can exactly be determined is $p = 2$, see [38]. In this paper, it is assumed that the ideal and nadir points exist and know.

**Definition 2.5.** *The problem (2) is called a convex MOP if all the objective functions $f_i$, $i = 1, \ldots, p$, and feasible set $X$ are convex.*

**Definition 2.6.** *Let $\varepsilon > 0$, and $x^* \in X$ be an optimal solution of the following single-objective optimization problem (SOP).*

$$\begin{aligned} &min \quad \varphi(x) \\ &s.t. \quad x \in X. \end{aligned}$$

*A feasible solution $\hat{x} \in X$ is called an $\varepsilon-$optimal solution if $\varphi(\hat{x}) - \varepsilon \leq \varphi(x^*)$.*

### *2.1. The PS approach*

The common approach to solve MOP is scalarization which combines the different objective functions and converts MOP into one parametric SOP. Pascoletti and Serafini [39] introduced a scalarization approach, which considers the following SOP denoted by SP(a,r) with parameters $a \in \mathbb{R}^p$ as a reference point and $r \in \mathbb{R}^p$ as a direction vector.

$$\begin{aligned} &min \quad t \\ &s.t. \quad f(x) \leqq a + tr, \qquad (SP(a, r)) \\ &\quad\quad x \in X, \\ &\quad\quad t \in \mathbb{R}. \end{aligned}$$

Note that an optimal solution of $SP(a, r)$ for specific parameters is a weakly Pareto optimal solution of the problem (2) and all Pareto optimal solutions of the problem (2) can be produced by varying parameters $a$ and $r$, see [40]. Since solving each SOP can be very costly, some SOPs are solved for a number of predetermined parameters and some non-dominated points are obtained as an approximation of the actual Pareto front.

Now, let $(x, t)$ be an optimal solution of $SP(a, r)$. Then, $f(x)$ is a weakly non-dominated point of MOP and there are two cases for this point as follows:

1. $f(x)$ is located along the vector $r$ and $f(x) = a + tr$, see Figure 1(a). This case always occurs in convex MOPs.
2. $f(x)$ is not located along the vector $r$ and $f(x) \leq a + tr$, see Figure 1(b).

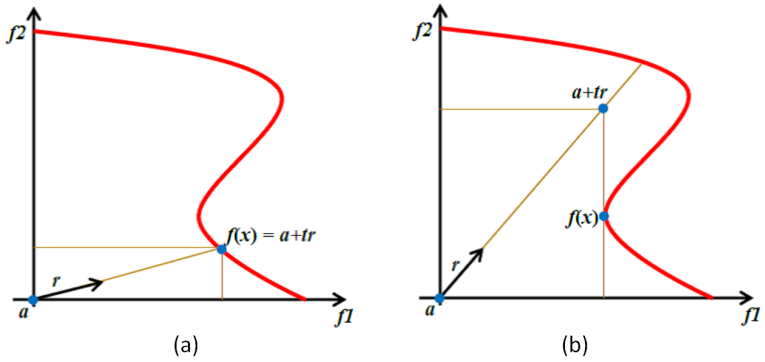


Figure 1: (a) The non-dominated point $f(x)$ is along $r$ and (b) non-dominated point $f(x)$ is not along $r$

*2.2. Relation between the problem (1) and its bi-objective optimization problem*

The following theorem is a main role to present algorithms for solving the problem (1).

**Theorem 2.7.** *A global optimal solution of the problem (1) is a Pareto optimal solution of the following bi-objective problem.*

$$\begin{aligned} &min \quad F(x) = (f(x), -g(x))^T \\ &s.t. \quad x \in X. \end{aligned} \tag{3}$$

*Proof.* *Let $x^* \in X$ be a global optimal solution of the problem (1), but it is not a Pareto optimal solution of the problem (3). Then, without loss of generality, there is $x \in X$ such that $f(x) < f(x^*)$ and $-g(x) \leq -g(x^*)$. Consequently, $\frac{f(x)}{g(x)} < \frac{f(x^*)}{g(x^*)}$ and this inequality is in contradiction to optimality of $x^*$.* □

By Theorem 2.7, two algorithms are proposed for solving the problem (1) in the objective space. Hence, the problem (1) is projected into the objective space and is reformulated as follows:

$$\begin{aligned} &min \quad Z(y) \\ &s.t. \quad y \in Y, \end{aligned} \tag{4}$$

where $Y = \{F(x) \mid x \in X\}$ is the image of the feasible set in the objective space $\mathbb{R}^2$ and $Z(y) = \frac{y_1}{-y_2}$ for each $y = (y_1, y_2) \in Y$. Now, consider the following problem:

$$\begin{aligned} &min \quad Z(y) \\ &s.t. \quad y \in Y', \end{aligned} \tag{5}$$

where $Y' = \{y \in \mathbb{R}^2 \mid F(x) \leqq y \leqq b, \ for\ some\ x \in X\}$ such that $y^N \leqq b$. The problems (4) and (5) have the same objective function and according to Theorem 2.7, an optimal solution of the problem (1) is a non-dominated point of its corresponding bi-objective problem. On the other hand, non-dominated points of the sets Y and $Y'$ are the same, i.e., $Y_N = Y'_N$, see [38]. Thus, optimal solutions of the problems (4) and (5) are the same, and the problem (5) can be used instead of the problem (4).

**Theorem 2.8.** *If the ideal point of the problem (3) is feasible, its corresponding solution in the decision space is an optimal solution of the problem (1).*

*Proof.* *Let $(F_1(x^1), F_2(x^2))$ be the ideal point of the problem (3) in which $x^1 = argmin_{x\in X} f(x)$ and $x^2 = argmin_{x\in X} - g(x) = argmax_{x\in X} g(x)$. If this point is feasible, there is a feasible solution $\hat{x} \in X$, which minimizes $f(x)$ and maximizes $g(x)$ on the set X. It is obvious that $\hat{x}$ is an optimal solution of the problem (1).* □

## 3. Algorithm 1 for solving convex FP problems

Shao and Ehrgott [37] suggested the objective space cut and bound approximation algorithm to solve convex multiplicative programming problems. We use this algorithm and propose an algorithm called Algorithm 1 for solving convex FP problems and obtain an $\varepsilon-$optimal solution of the problem (1). Consider a convex FP problem such that $f(x)$, $-g(x)$ and the constraint functions $h_i(x), i = 1, \ldots, m$, are convex and continuously differentiable on the compact feasible set $X$ (or $Y$ is $\mathbb{R}^2$-bounded from below and $\mathbb{R}^2$-closed). At first, an exact approach presented in [38] is used to calculate the ideal point, nadir point and anchor points of the related bi-objective optimization problem (3). Set $Z^* = min\{Z(y^1), Z(y^2)\}$ in which $y^1$ and $y^2$ are the anchor points. In addition, $x'$ and $y'$ are solutions corresponding to $Z^*$ in the decision space and objective space, respectively. Then, it is checked that $f(x)$ and $-g(x)$ are conflicting or not. If these functions are not conflict, then the ideal point is feasible and is an optimal solution of the problem (5). Otherwise, do the following iterative algorithm.

Algorithm 1 starts with a polyhedron $S^0 = \{y \in \mathbb{R}^2 \mid y^I \leqq y \leqq b\}$ such that $y^N \leqq b$ and $Y' \subseteq S^0$. Set $k := 0$, consider an approximation error $\varepsilon > 0$, and obtain $LB_0 := Z(y^I)$ and $UB_0 := Z(b)$ as an initial lower bound value and an initial upper bound value, respectively. At iteration $k$, a vertex $s^k$ of the polyhedron $S^k$ is chosen with a minimum value of the function $Z$ among all the vertices such that $Z(s^k) < UB_k$. Then, the boundary point $y^k := F(x^k) \in Y'$ is calculated by solving $SP(s^k, r^k)$ with $r^k := b - s^k$. As previously mentioned, $y^k$ is a non-dominated point and since the Pareto front is convex, $y^k$ is the intersection of the objective space and the direction vector $r^k$. If $Z(y^k) < UB_k$, then $UB_k$ is updated with $Z(y^k)$. Next, if the absolute difference between the upper bound and lower bound values is less than $\varepsilon$, then Algorithm 1 terminates by considering that whether $Z^* < UB_k$ or not. If $Z^* < UB_k$, then set $UB_k := Z^*$. Otherwise, a supporting hyperplane of $Y'$ at the point $y^k$ is constructed as $\{z \in \mathbb{R}^2 \mid z^T\lambda^k = F(x^k)^T\lambda^k\}$ such that $(\lambda^k, u^k)$ is an optimal solution of the following linear SOP:

$$
\begin{aligned}
max \quad & b_{x^k}^T u - (y^k - f_{x^k})^T \lambda \\
s.t. \quad & A_{x^k}^T u - C_{x^k}^T \lambda = 0, \qquad (ID(x^k, y^k)) \\
& e^T \lambda = 1, \\
& u, \lambda \in \mathbb{R}^2_{\leqq},
\end{aligned}
$$

where $\nabla F(x^k)$ and $\nabla h(x^k)$ are the gradients of $F(x^k)$ and $h(x^k)$, respectively. In addition, $f_{x^k} := F(x^k) - \nabla F(x^k)^T x^k$, $b_{x^k}^T := h(x^k) - \nabla h(x^k)^T x^k$, $A_{x^k}^T := -\nabla h(x^k)^T$, $C_{x^k}^T := \nabla F(x^k)^T$ and $e := (1, 1)$. This hyperplane separates $s^k$ from the set $S^k$ and generates a new polyhedron $S^{k+1}$ such that $S^{k+1} \subseteq S^k$. In more detail, $S^{k+1}$ is the intersection of $S^k$ and $\{z \in \mathbb{R}^2 \mid z^T\lambda^k \geq F(x^k)^T\lambda^k\}$. It is clear that a hyperplane can be considered when the Pareto front of the related bi-objective problem is convex. Then, all vertices of $S^{k+1}$, symbolized by $vert(S^{k+1})$, are obtained by a method presented by Chen et al. [41] and $Z(s)$ is calculated for each vertex $s \in vert(S^{k+1})$ such that $Z(s) < UB_k$. A minimum value of $Z(s)$ among all these vertices is obtained and is set $LB$. If $LB$ is greater than the lower bound value $LB_k$, then the lower bound value is updated to $LB$. Furthermore, if the absolute difference between the upper bound value and the lower bound value is less than $\varepsilon$, then Algorithm 1 terminates by considering that whether $Z^* < UB_k$ or not, and if $Z^* < UB_k$, then set $UB_k := Z^*$. Otherwise, set $k := k + 1$ and the algorithm is repeated.

Therefore, Algorithm 1 constructs a sequence of polyhedrons that each of them is an approximation of $Y'$ from the outside. Steps of Algorithm 1 are summarized as follows:

**Algorithm 1: an algorithm for solving convex FP problems**

**Initialisation:**

- Choose $\varepsilon > 0$ and calculate the ideal point, nadir point, anchor points, and $Z^*$. Select $b$ such that $y^N \leqq b$. Construct a polyhedron $S^0$, set $vert(S^0) := \{y^I\}$, $LB_0 := Z(y^I)$, $UB_0 := Z(b)$, and $k := 0$.

**Main iteration repeat:**

1. Choose a vertex $s^k \in vert(S^k)$ with $Z(s^k) < UB_k$ and obtain a minimum value of the objective function $Z$ among all vertices of $S^k$.
2. Obtain an optimal solution $(x^k, t^k)$ by solving $SP(s^k, r^k)$ with $r^k := b - s^k$ and set $y^k := F(x^k)$.
3. If $Z(y^k) < UB_k$, then set $UB_k := Z(y^k)$, $y^c := y^k$, and $x^c := x^k$.
4. If $UB_k - LB_k \leq \varepsilon$ and $Z^* < UB_k$, then set $UB_k := Z^*$, $y^c := y'$, $x^c := x'$, and stop. If $UB_k - LB_k \leq \varepsilon$ and $UB_k \leq Z^*$, then stop. Otherwise go to Step 5.
5. Solve the problem $ID(x^k, y^k)$, obtain an optimal solution $(u^k, \lambda^k)$ and generate a polyhedron $S^{k+1} := \{z \in \mathbb{R}^2 \mid z^T\lambda^k \geq F(x^k)^T\lambda^k\} \bigcap S^k$.
6. Determine vertices of $S^{k+1}$ and set $LB := min\{Z(s) \mid s \in vert(S^{k+1})\}$. If $LB > LB_k$, then set $LB_k := LB$.
7. If $UB_k - LB_k \leq \varepsilon$ and $Z^* < UB_k$, then set $UB_k := Z^*$, $y^c := y'$, and stop. If $UB_k - LB_k \leq \varepsilon$ and $UB_k \leq Z^*$, then stop. Otherwise set $k := k + 1$, $UB_k := UB_{k-1}$, $LB_k := LB_{k-1}$, and go to Step 1.

In the following theorems, it is proved that by considering a pre-specified approximation error $\varepsilon > 0$, Algorithm 1 finds $x^c$ as an $\varepsilon-$optimal solution of the problem (1) and $y^c$ as an $\varepsilon-$optimal solution of the problems (4) or (5) in a finite number of iterations.

**Theorem 3.1.** *Let $f(x)$, $-g(x)$ and $h_i(x)$, $i = 1, \ldots, m$ are convex and continuously differentiable on the compact feasible set $X$. If Algorithm 1 is infinite, then it generates a sequence of feasible solutions of the problem (1) such that every accumulation point is a global optimal solution of the problem (1) and*

$$\lim_{k\to\infty} UB_k = \lim_{k\to\infty} LB_k = inf\{Z(y) \mid y \in Y'\}.$$

*Proof. It is similar to Theorem 4.2 in [37].* □

**Theorem 3.2.** *If $\varepsilon > 0$, then Algorithm 1 is terminated after a finite number of iterations at an $\varepsilon-$optimal solution of the problem (1).*

*Proof. It is similar to Theorem 4.3 in [37].* □

*3.1. Numerical results of Algorithm 1*

In this subsection, Algorithm 1 is applied to three numerical examples. It should be noted that all algorithms in this paper were coded in Matlab 2016 and all experiments were implemented on a Laptop with Pentium 4 at 2.3GHZ and 4GB RAM running Windows 7 Home Basic Operating system. Besides, the fmincon and fgoalattain solvers in Matlab are applied to solve every SOP and PS problem, respectively, in all algorithms.

**Example 3.3.** *Consider the following convex FP problem:*

$$\begin{aligned} min \quad & \frac{1 + x^2}{-(x-2)^2 + 18} \\ s.t. \quad & 0 \leq x \leq 2. \end{aligned}$$

*The following problem is a bi-objective problem corresponding to Example 3.3:*

$$\begin{aligned} min \quad & (f(x) = 1 + x^2, -g(x) = (x-2)^2 - 18) \\ s.t. \quad & 0 \leq x \leq 2. \end{aligned}$$

*In this problem, $y^I = (1, -18)$, $b = y^N = (5, -14)$, and $Z^* = 0.071429$. In addition, the initial lower bound value and initial upper bound value are 0.0556 and 0.3571, respectively, and $\varepsilon = 0.0025$. Figure 2 shows the ideal point, nadir point, $Y'$, and polyhedron $S^0$. Furthermore, Figure 3 illustrates points $y^k$, and polyhedrons $S^k$ obtained by Algorithm 1 at four iterations. Table 1 reports the vertex $s^k$, boundary point $y^k$, supporting hyperplane, upper bound value and lower bound value at four iterations for Example 3.3. After four iterations, $UB_4 = 0.0701$, $LB_4 = 0.0686$, and $UB_4 - LB_4 = 0.0015$ are obtained that this difference is less than $\varepsilon$ and $UB_k < Z^*$. Thus, Algorithm 1 terminates and obtains $y^c = (1.02, -14.56)$ as an $\varepsilon-$optimal solution of the problem (5), $x^c = 0.1458$ as an $\varepsilon-$ optimal solution of the problem (1) at 1.22 seconds.*

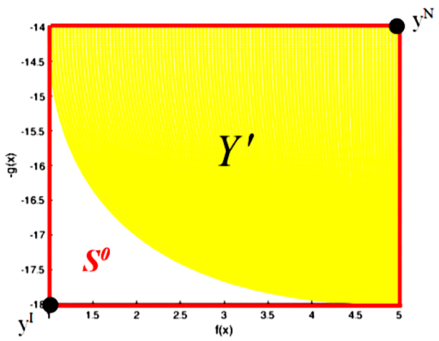


Figure 2: Illustration of $y^I$, $y^N$, the set $Y'$, and the polyhedron $S^0$ for Example 3.3

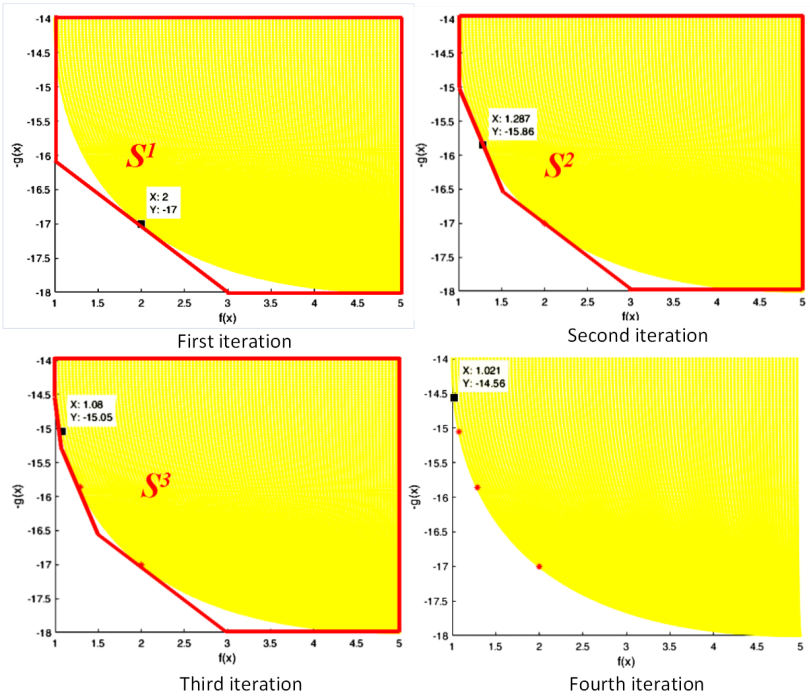


Figure 3: The points $y^k$ and polyhedrons $S^k$ at four iterations of Algorithm 1 for Example 3.3

Table 1: The results obtained by Algorithm 1 at four iterations for Example 3.3

| Iteration | $s^k$ | $y^k$ | Supporting hyperplane | $UB_k$ | $LB_k$ |
|---|---|---|---|---|---|
| $k = 0$ | $(1, -18)$ | $(2, -17)$ | $0.5y_1 + 0.5y_2 = -7.50$ | 0.1176 | 0.0625 |
| $k = 1$ | $(1, -16)$ | $(1.29, -15.86)$ | $0.73y_1 + 0.27y_2 = -3.31$ | 0.0812 | 0.0663 |
| $k = 2$ | $(1, -15.07)$ | $(1.08, -15.05)$ | $0.86y_1 + 0.14y_2 = -1.20$ | 0.0717 | 0.0687 |
| $k = 3$ | $(1, -14.5651)$ | $(1.02, -14.56)$ | $0.93y_1 + 0.07y_2 = -0.12$ | 0.0701 | 0.0686 |

**Example 3.4.** *The second convex FP problem is stated as follows:*

$$\begin{aligned} min \quad & \frac{f(x)}{g(x)} = \frac{(2x_1 + x_2)^2 + 1}{x_1 + 2x_2 + 1} \\ s.t. \quad & 2x_1 + x_2 \geq 6, \\ & x_1 + 3x_2 \geq 8, \\ & 0 \leq x_1, x_2 \leq 5, \end{aligned}$$

*where $f(x)$ is a convex function, $g(x)$ is an affine function and X is a convex set. (37, -16) and (226, -11.5) are the ideal and nadir points of a bi-objective problem corresponding to Example 3.4. In Algorithm 1, $x' = (0.5, 5)$, $Z^* = 3.217391$, $LB_0 = 2.3125$, and $UB_0 = 19.6522$ are calculated, and $\varepsilon = 0.001$ and $b = (226, -5)$ are considered. Algorithm 1 obtains $UB_4 = 3.217395$ in last iteration and since $Z^* < UB_4$, Algorithm 1 sets $UB_4 = Z^*$, $y^c = (37, -11.50)$, and $x^c = (0.5, 5)$. The CPU time of Algorithm 1 is 3.23 seconds. Figure 4 shows the feasible objective set of the bi-objective problem corresponding to Example 3.4 and points $y^k$ obtained by Algorithm 1 in the objective space at four iterations.*

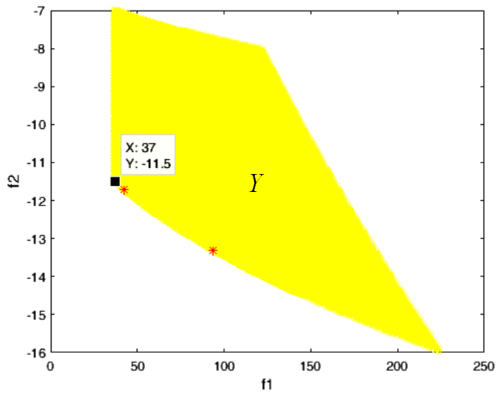


Figure 4: The points $y^k$ obtained by Algorithm 1 for Example 3.4 at four iterations

According to the description of Algorithm 1, it is able to obtain an $\varepsilon-$optimal solution of the problem (1) whose the corresponding bi-objective problem (3) has the convex Pareto front. Example 3.5 emphasizes this fact.

**Example 3.5.** *Consider the following FP problem:*

$$\begin{aligned} min \quad & \frac{f(x)}{g(x)} = \frac{3x_1^2 + 2x_2^2 - 4x_1 - 8x_2 + 9}{x_1^2 + x_2^2 - 6x_2 + 8} \\ s.t. \quad & x_1 + x_2 \geq 2, \\ & x_1, x_2 \geq 0, \end{aligned}$$

*where $f(x)$ and $g(x)$ are convex functions, and the feasible set is convex. The ideal and nadir points of a bi-objective problem corresponding to Example 3.5 are $(0.2, -12)$ and $(13, -1.12)$, respectively. Algorithm 1 considers $b = y^N$, $\varepsilon = 0.001$, $Z^* = 0.178571$, $LB_0 = 0.016667$, and $UB_0 = 11.607142$. It obtains $y^c = (0.2178, -1.349)$, $x^c = (0.46, 1.54)$, and 0.162305 as an objective value of Example 3.5 at six iterations and 2.64 seconds. Figure 5 demonstrates the feasible objective set of the bi-objective problem and points $y^k$ obtained by Algorithm 1 in the objective space. Table 2 reports vertices $s^k$, boundary points $y^k$, supporting hyperplanes, upper bound values and lower bound values at six iterations for Example 3.5.*

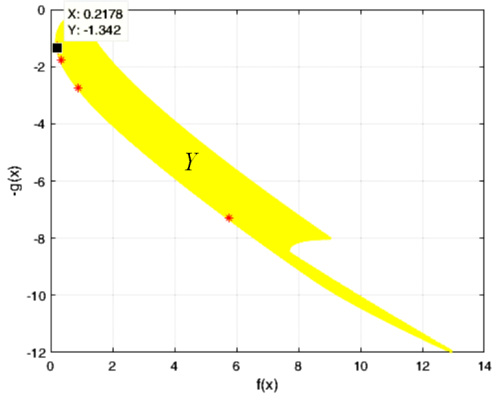


Figure 5: The points $y^k$ obtained by Algorithm 1 for Example 3.5 at four iterations

Table 2: The results obtained by Algorithm 1 at six iterations for Example 3.5

| Iteration | $s^k$ | $y^k$ | Supporting hyperplane | $UB_k$ | $LB_k$ |
|---|---|---|---|---|---|
| $k = 0$ | $(0.20, -12)$ | $(5.74, -7.29)$ | $0.45y_1 + 0.55y_2 = -1.48$ | 0.7878 | 0.0706 |
| $k = 1$ | $(0.20, -2.83)$ | $(0.89, -2.74)$ | $0.58y_1 + 0.42y_2 = -0.64$ | 0.3262 | 0.1117 |
| $k = 2$ | $(0.20, -1.79)$ | $(0.34, -1.78)$ | $0.72y_1 + 0.28y_2 = -0.26$ | 0.1917 | 0.1405 |
| $k = 3$ | $(0.20, -1.42)$ | $(0.23, -1.42)$ | $0.83y_1 + 0.17y_2 = -0.05$ | 0.1633 | 0.1581 |
| $k = 4$ | $(0.20, -1.26)$ | $(0.21, -1.26)$ | $0.91y_1 + 0.09y_2 = 0.07$ | 0.1633 | 0.1608 |
| $k = 5$ | $(0.20, -1.34)$ | $(0.22, -1.34)$ | $0.87y_1 + 0.13y_2 = 0.01$ | 0.1623 | 0.1625 |

## 4. An MOP algorithm based on the PS approach

In this section, an algorithm based on the PS approach named Algorithm 2 is proposed to generate almost uniform approximations of the entire Pareto front of bi-objective optimization problems. In addition, it obtains non-dominated points in non-convex parts of the Pareto front and disregards dominated points. In algorithms based on the PS approach, an approximation of the Pareto front is obtained by selecting a set of parameters $a$ and $r$, and solving their corresponding PS problems. Most of these algorithms consider a set of these parameters such that either $a$ is fixed and $r$ is changed, see [42] and Figure 6(a), or $r$ is fixed and $a$ is changed, see [40, 43, 44] and Figure 6(b). In iterative Algorithm 2, an almost uniform approximation of the Pareto front is constructed by changing both the reference point and direction vector.

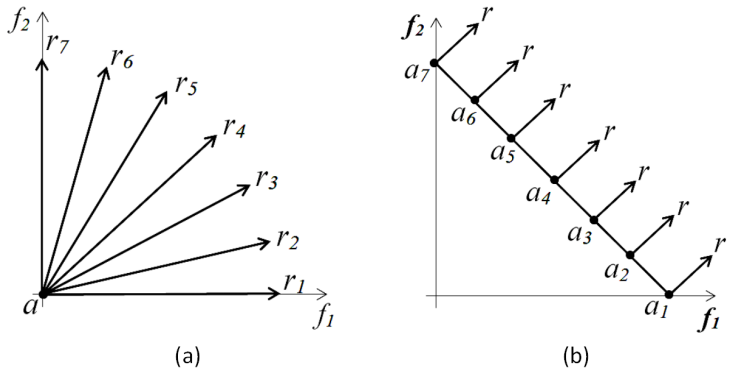


Figure 6: (a) An approach with a fixed reference point and different direction vectors and (b) an approach with a fixed direction vector and different reference points

Assume that $a = (a_1, a_2)$ is a reference point and $b = (b_1, b_2)$ is a point corresponding to the reference point located on a rectangle where $a$ and $b$ are its opposite vertices. Let $(x, t)$ be an optimal solution of

$SP(a,r)$, in which $r = \frac{b-a}{\|b-a\|_2}$. Then, $f(x)$ is a (weakly) non-dominated point, see Figure 7(a). Then, by having points $y = a + tr$ and $f(x)$, a set $Y_0 = \{\hat{y} \in \mathbb{R}^2 \mid a + \mathbb{R}^2_{\leqq} \leqq \hat{y} \leqq b - \mathbb{R}^2_{\leqq}\}$ is partitioned into four regions 1, 2, 3 and 4, see Figure 7(b). There is no non-dominated point in the regions 1 and 3, because $f(x)$ dominates each feasible solution in the region 1 and there are no feasible solutions in the region 3. Hence, the regions 2 and 4 are only examined. For this purpose, the point $y = (y_1, y_2)$ is imaged on lines $f_1 = a_1$ and $f_2 = a_2$, and the points $a^1 = (y_1, a_2)$ and $a^2 = (a_1, y_2)$ are obtained as the new reference points. In addition, the point $f(x) = (f_1(x), f_2(x))$ is imaged on lines $f_1 = b_1$ and $f_2 = b_2$, and the points $b^1 = (b_1, f_2(x))$ and $b^2 = (f_1(x), b_2)$ are obtained as the points corresponding to $a^1$ and $a^2$, respectively. It is noted that $b^1$ is a vertex in front of $a^1$ in the rectangle 4 and $b^2$ is a vertex in front of $a^2$ in the rectangle 2.

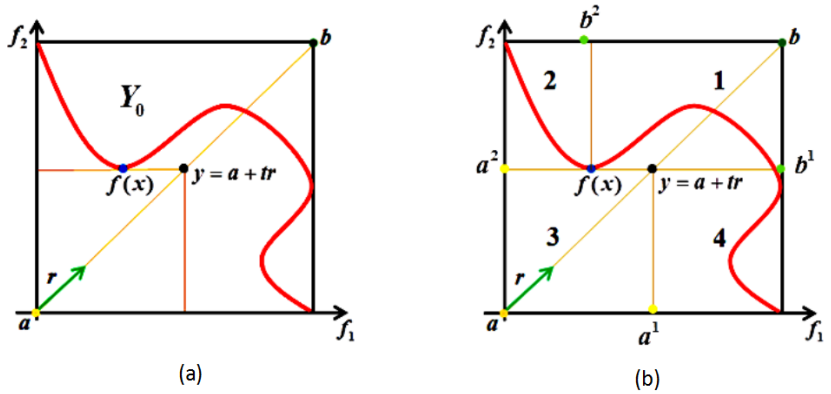


Figure 7: (a) Solving $SP(a,r)$ with $r = \frac{b-a}{\|b-a\|_2}$ and obtaining $f(x)$ and $y$ and (b) partitioning $Y_0$ into four regions for a bi-objective problem

Let $(x^1, t^1)$ and $(x^2, t^2)$ be optimal solutions of problems $SP(a^1, r^1)$ and $SP(a^2, r^2)$ such that $r^1 = \frac{b^1-a^1}{\|b^1-a^1\|_2}$ and $r^2 = \frac{b^2-a^2}{\|b^2-a^2\|_2}$. In this example, $f(x^1)$ and $f(x^2)$ are non-dominated points, and two points $y^2 = a^2 + t^2 r^2$ and $f(x^2)$ are coincident, see Figure 8(a).

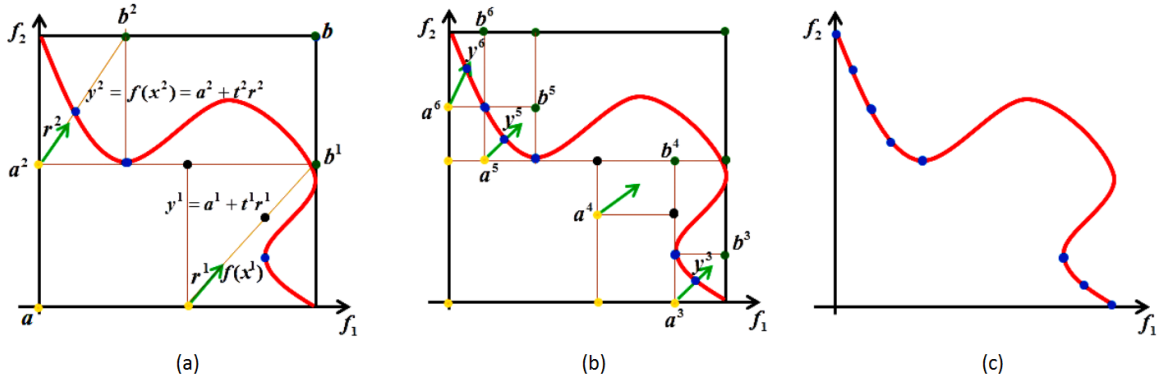


Figure 8: (a)-(b) Some iterations of Algorithm 2, and (c) the approximation of the Pareto front of a non-convex bi-objective problem obtained by Algorithm 2

The next steps are done in the same way by having each pair of the points $y^1$ and $f(x^1)$, and the points $y^2$ and $f(x^2)$. Therefore, $a^3, a^4, a^5$, and $a^6$ are obtained as new reference points and $b^3, b^4, b^5$, and $b^6$ are obtained as points corresponding to these reference points, respectively. Then, non-dominated points $y^3$, $y^5$, and $y^6$ are obtained by solving problems $SP(a^3, r^3)$, $SP(a^5, r^5)$, and $SP(a^6, r^6)$, respectively, in which $r^i = \frac{b^i-a^i}{\|b^i-a^i\|_2}$ for $i = 3, 5, 6$, see Figure 8(b). Note that if $SP(a^4, r^4)$ is solved with $r^4 = \frac{b^4-a^4}{\|b^4-a^4\|_2}$, the point $f(x^1)$ or $y$ is obtained

again. In order to avoid considering this case, let $(x, t)$ be an optimal solution of $SP(z, r)$ with $r = \frac{Z-z}{\|Z-z\|_2}$. If $z < f(x) < Z$, new reference points and their corresponding points are considered. Figure 8(c) illustrates an approximation of the Pareto front of a non-convex bi-objective problem obtained by Algorithm 2. It should be noted that two anchor points are considered in the approximation.

One of the two following conditions is used for considering the reference point $z$ and their corresponding point $Z$. Consider a rectangle where $z$ and $Z$ are its opposite vertices. The first condition (C1) is that the area of the rectangle is greater than a predetermined value $\varepsilon$ and the second condition (C2) is that maximum of the length and width of this rectangle is greater than $\varepsilon$.

Now, Algorithm 2 is explained for solving a bi-objective optimization problem in detail. At first, the ideal point, nadir point and two anchor points of the bi-objective problem are calculated. At the end of the algorithm, a set $PF$ is an approximation of the Pareto front obtained by Algorithm 2. At first, this set only contains the anchor points and then, obtained points are added to it at each iteration. In addition, two sets $A$ and $B$ are considered such that $A$ consists of reference points and initially includes the ideal point, and $B$ consists of points corresponding to the reference points and initially includes the nadir point. Besides $\varepsilon$ is a predetermined value and one of the conditions C1 and C2 is considered. Then, the following steps are done until $A$ is equal to empty:

**Step 1:** $z := (z_1, z_2) \in A$ and its corresponding point $Z := (Z_1, Z_2) \in B$ are chosen. A normalized direction vector $r := \frac{Z-z}{\|Z-z\|_2}$ is considered and set $A := A\backslash\{z\}$ and $B := B\backslash\{Z\}$. An optimal solution $(x, t)$ is gained by solving $SP(z, r)$.

**Step 2:** By considering a weakly non-dominated point $f(x) = (f_1(x), f_2(x))$ and $y := z + tr$, two points $a^1 := (y_1, z_2)$ and $a^2 := (z_1, y_2)$ are considered as candidate reference points that their corresponding points are $b^1 := (Z_1, f_2(x))$ and $b^2 := (f_1(x), Z_2)$, respectively.

**Step 3:** If $z < f(x) < Z$, then do the following steps:

- The point $f(x)$ is added to $PF$.
- If a rectangle generated by $a^1$ and $b^1$ such that $a^1 \neq z$ and $b^1 \neq Z$ has the considered condition, $a^1$ is added to $A$ and $b^1$ is added to $B$.
- If a rectangle generated by $a^2$ and $b^2$ such that $a^2 \neq z$ and $b^2 \neq Z$ has the considered condition, $a^2$ is added to $A$ and $b^2$ is added to $B$.

Finally, dominated points are removed from $PF$.

### *4.1. Evaluating the quality of approximations of the Pareto front*

Generating non-dominated points that distribute uniformly on the entire Pareto front is an important feature in MOP algorithms. There are several metrics in the literature to evaluate and compare the quality of approximations of the Pareto front obtained by different algorithms. In following, some of these metrics including set coverage, purity, coverage, and spacing are reviewed.

#### *4.1.1. Set coverage and Purity*

The set coverage metric $C(S_1, S_2)$ [45] presents the percentage of points in the set $S_2$ dominated by at least one point from the set $S_2$ and is defined as follows:

$$C(S_1, S_2) = \frac{|\{y_2 \in S_2 \mid \exists y_1 \in S_1 \text{ such that } y_1 \leq y_2\}|}{|S_2|}.$$

If $C(S_1, S_2) = 0$, then none of the points of $S_2$ is dominated by at least one point of $S_1$ and $C(S_1, S_2) = 1$ means all points of $S_2$ are dominated by at least one point of $S_1$. In addition, the approximation $S_1$ is better than $S_2$, if $C(S_1, S_2)$ is larger and $C(S_2, S_1)$ is smaller.

Now consider $N$ algorithms to obtain approximations of the Pareto front of an MOP. Let $S_i$ denotes an approximation obtained by $i^{th}$ algorithm and $n_i = |S_i|$ for all $i = 1, \ldots, N$. A set $S$ is the union of all

the approximations such that all dominated points are removed from it. The purity metric [46] for $S_i$ ($i = 1, \ldots, N$) is defined as follows:

$$P = \frac{|S_i \cap S|}{n_i}.$$

Note that the purity metric is a real number between [0, 1] and an algorithm with a larger value for this metric has a better performance in terms of the percentage of non-dominated points.

*4.1.2. Coverage*

Meng et. al. [47] proposed Extension as a metric of the coverage that examines whether all regions of the Pareto front are represented. It is an average distance of an approximation $S$ from $k$ points $\{v^1, v^2, \ldots, v^k\}$ as a set of reference points. At first, the distance between each reference point and $S$ is calculated as follows:

$$d(v^i, S) = min\{d(v^i, y) \mid y \in S\}, \qquad \forall i \in \{1, \ldots, k\}.$$

Then, the Extension is obtained as follows:

$$EX(S) = \frac{\sqrt{\sum_{i=1}^{k} (d(v^i, S))^2}}{k}.$$

An approximation with a large value of Extension shows that the obtained points mainly locate in the center of the actual Pareto front without considering the outskirts. Therefore, a small value of $EX$ is preferred to a larger value. Note that anchor points are considered as the reference points in this paper.

*4.1.3. Spacing*

Messac and Mattson [48] presented Evenness as a spacing metric to verify how points are distributed on the Pareto front. Evenness of the approximation S is calculated as follows:

$$EV(S) = \frac{\sigma(D)}{\mu(D)}, \qquad D = \{d_l^1, d_u^1, d_l^2, d_u^2, \ldots, d_l^m, d_u^m\}, |S| = m,$$

where $\sigma(D)$ and $\mu(D)$ are the standard deviation and arithmetic mean of D, respectively. In addition, $d_l^i$ ($i \in \{1, \ldots, m\}$) is a minimum distance of each point $y^i \in S$ to any other point in $S$ and $d_u^i$ ($i \in \{1, \ldots, m\}$) is the maximum radius of a (hyper) sphere that can be formed between $y^i \in S$ and any other point in $S$ such that no other points are within the (hyper) spheres. If all points of the approximation are equidistant, $d_l$ and $d_u$ will all be equal, then $\sigma(D) = 0$ and $EV(S) = 0$.

Note that having equidistant points on the Pareto front does not guarantee a good coverage, therefore, a spacing metric should be used with a coverage metric. In this paper, an approximation with the smallest Extension and Evenness are desirable.

*4.2. Numerical implementations*

In this section, the performance of Algorithm 2 is checked by five test problems with convex, non-convex, connected, and disconnected Pareto fronts. Results of this algorithm are compared to results of evolutionary algorithms including the non-dominated sorting genetic algorithm-II (NSGA-II) [49], differential evolution (DE) with binomial crossover [50], and S metric selection evolutionary multiobjective algorithm (SMS-EMOA) [51]. Besides, some algorithms based on the scalarization approach such as the normal constraint (NC) [52] with $\delta = \frac{1}{100}$, weighted constraint (WC) [53] with $N = 100$, and Benson type algorithm [54] are considered for better evaluation of Algorithm 2. In these evolutionary algorithms, the population size and number of the function evaluations are considered 100 and 20000, respectively. In all tables of this subsection, columns 2-11 show the number of solutions obtained ($NS$), number of SOPs solved ($NSOP$), Evenness ($EV$), Extension ($EX$), purity ($P$), $C(Alg2, S)$, $C(S, Alg2)$, number of the function evaluations per each obtained solution ($FE/NS$), and CPU time in second to solve each problem ($CPU$), respectively. Note that $S$ is an approximation obtained by one of the mentioned algorithms, which will be compared to Algorithm 2 and $Alg2$ is an approximation obtained by Algorithm 2.

*4.2.1. Test problem 1*

The 10-variable DTLZ2 problem is the first example and has a non-convex and connected Pareto front. This problem is formulated as follows:

$$\begin{aligned} min \quad & f_1(x) = cos(\frac{\pi}{2}x_1)(1 + \sum_{i=2}^{10}(x_i - 0.5)^2) \\ min \quad & f_2(x) = sin(\frac{\pi}{2}x_1)(1 + \sum_{i=2}^{10}(x_i - 0.5)^2) \\ s.t. \quad & x_i \in [0,1],\ i = 1, \ldots, 10. \end{aligned}$$

By considering the second condition with $\varepsilon = 0.05$ and $\varepsilon = 0.025$, Algorithm 2 is capable of obtaining points, which cover all the regions of the Pareto front almost uniformly, see Figure 9. Algorithm 2 obtains 41 points on the Pareto front at 1.29 seconds, 2401 function evaluations, $EV = 0.1641$, and $EX = 0$ for $\varepsilon = 0.05$.

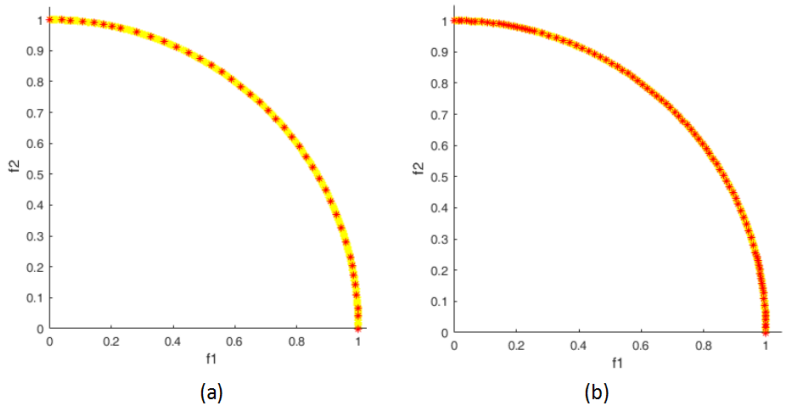


Figure 9: The approximations of the Pareto front of Test problem 1 obtained by Algorithm 2 for (a) $\varepsilon = 0.05$ and (b) $\varepsilon = 0.025$

Figure 10 illustrates approximations of the Pareto front obtained by the Benson type algorithm with $\epsilon = 0.01$, $\hat{p} = (100, 100)$, and $d = (1, 1)$ and the other mentioned algorithms for Test problem 1. This figure and Figure 9 indicate that all regions of the Pareto front are only covered by the points obtained by Algorithm 2 and WC, while the other algorithms cannot cover the entire Pareto front. In addition, the results of Algorithm 2 with $\varepsilon = 0.025$ and the other algorithms are given in Table 3. These results emphasize that Algorithm 2 can obtain an approximation with high quality for all indexes in Test problem 1. None of the points obtained by Algorithm 2 is dominated by the points obtained by the other algorithms, i.e., $P = 1$ and $C(S, Alg2) = 0$ for each approximation $S$ obtained by the other algorithms. Moreover, values of $EV$ and $EX$ of Algorithm 2 are less than the other algorithms and this algorithm can obtain these points in the acceptable time with the least number of the function evaluations per each obtained solution. It should be noted that the CPU time of the DE algorithm is less than Algorithm 2, but the quality of Algorithm 2 is better than the DE algorithm in the other indexes. Therefore, Algorithm 2 is able to generate an almost uniform approximation of the entire Pareto front with the best quality among the other six algorithms.

Table 3: The numerical results of the seven algorithms for Test problem 1

| Method | *NS* | *NSOP* | *EV* | *EX* | *P* | $C(Alg2, S)$ | $C(S, Alg2)$ | *FE/NS* | *CPU* |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm 2 | 85 | 85 | 0.1948 | 0 | 1 | 0 | 0 | 62.01 | 2.50 |
| NC | 96 | 102 | 0.4186 | 7.92e-9 | 0.9792 | 0.0316 | 0 | 275.34 | 7.42 |
| WC | 128 | 204 | 0.8315 | 5.49e-5 | 1 | 0.0229 | 0 | 99.98 | 5.80 |
| Benson type | 143 | 288 | 0.6004 | 3.97e-4 | 0.9930 | 0 | 0 | 228.16 | 16.68 |
| DE | 98 | - | 1.2468 | 0.3321 | 0.9898 | 0.0102 | 0 | 204.08 | 2.46 |
| SMS-MOEA | 100 | - | 0.6441 | 0 | 0.9900 | 0.0200 | 0 | 200.00 | 19.91 |
| NSGA-II | 100 | - | 0.4707 | 0.0002 | 0.6200 | 0.1000 | 0 | 200.00 | 203.08 |

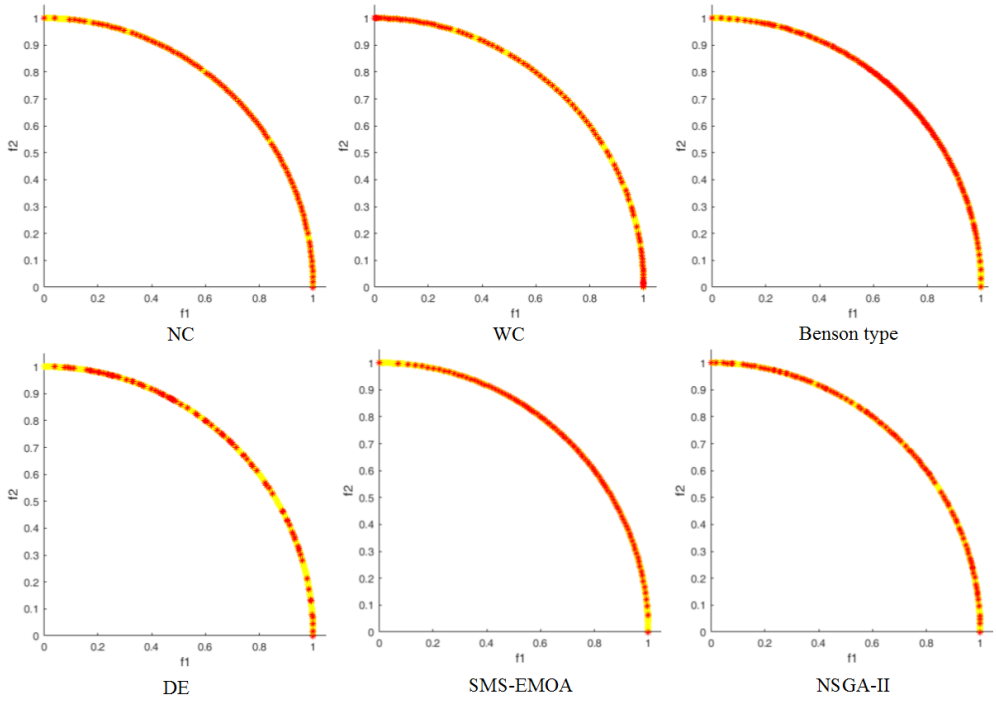


Figure 10: The approximations of the Pareto front of Test problem 1 obtained by the six algorithms

*4.2.2. Test problem 2*

The following bi-objective problem with a convex and connected Pareto front is another selected problem to test the quality of Algorithm 2. This problem was examined in [55, 56].

$$\begin{aligned} &min \quad f_1(x) = \sqrt{1 + x_1^2} \\ &min \quad f_2(x) = x_1^2 - 4x_1 + x_2 + 5 \\ &s.t. \quad x_1, x_2 \geq 0. \end{aligned}$$

The ideal and nadir points are (1.0842, 1) and (2.2361, 3.5), respectively. Algorithm 2 considers the first condition and obtains 33 points on the Pareto front at 0.33 seconds, 566 function evaluations, $EV = 0.2394$, and $EX = 0$ for $\varepsilon = 0.005$. Figure 11 shows the points obtained by Algorithm 2 for $\varepsilon = 0.005$ and $\varepsilon = 0.001$.

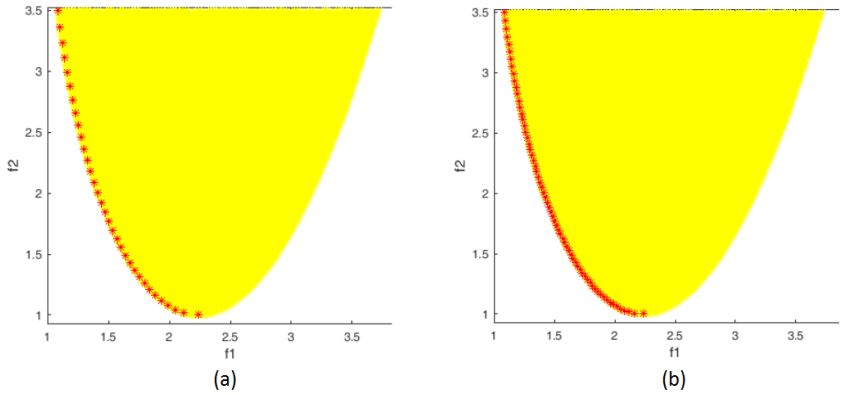


Figure 11: The approximations of the Pareto front of Test problem 2 obtained by Algorithm 2 for (a) $\varepsilon = 0.005$ and (b) $\varepsilon = 0.001$

Figure 12 shows approximations of the Pareto front obtained by the other mentioned algorithms for Test problem 2. In addition, the Benson type algorithm considers $\epsilon = 0.02$, $\hat{p} = (100, 100)$, and $d = (1, 1)$. As seen in this figure, the Benson type, NSGA-II, and DE algorithms are not able to generate the approximations that cover all regions of the actual Pareto front.

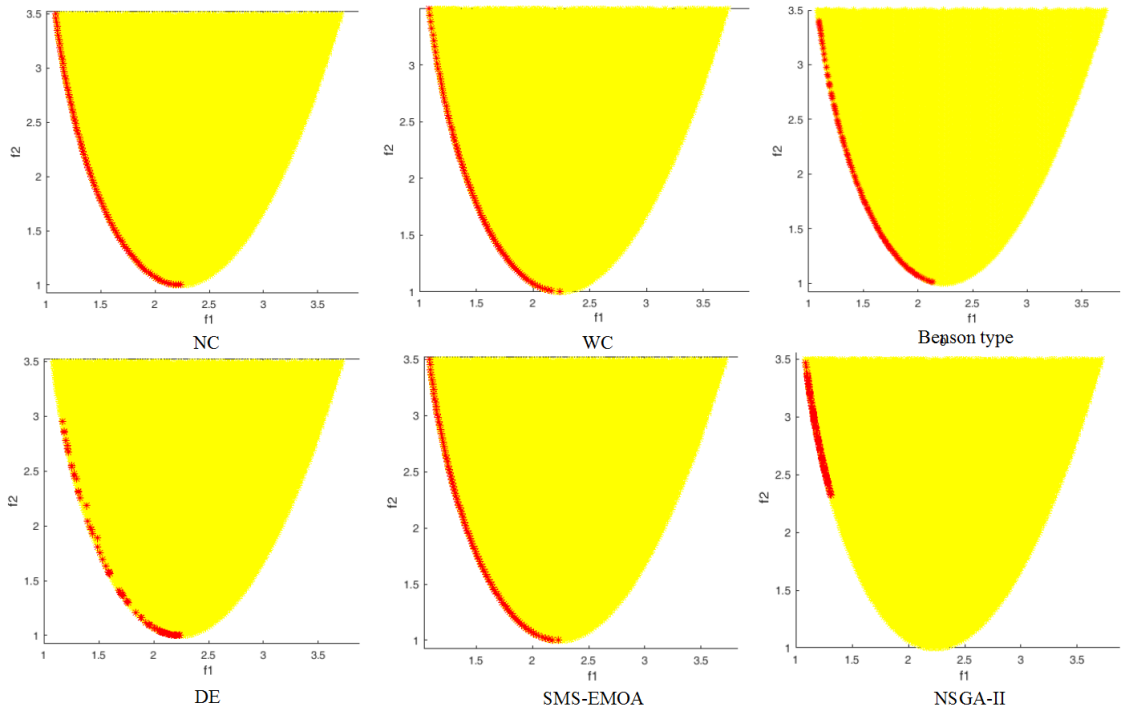


Figure 12: The approximations of the Pareto front of Test problem 2 obtained by the six algorithms

Table 4 shows the results of Algorithm 2 with $\varepsilon = 0.001$ and the other mentioned algorithms for Test problem 2. Since, the purity of Algorithm 2, Benson type, SMS-EAMO, and NSGA-II are equal to 1 and none of the points obtained by these algorithms is dominated by the points obtained by Algorithm 2, these algorithms have a good performance in terms of the percentage of non-dominated points. Besides, the proposed algorithm has the least CPU time and number of the function evaluations per each obtained solution. From a comparison of $EX$ and $EV$, it can be concluded that the quality of the distribution of the points obtained by Algorithm 2 is better than that of the other algorithms, because it has the smallest $EX$ and $EV$ among the other mentioned algorithms.

Table 4: The numerical results of the seven algorithms for Test problem 2

| Method | $NS$ | $NSOP$ | $EV$ | $EX$ | $P$ | $C(Alg2, S)$ | $C(S, Alg2)$ | $FE/NS$ | $CPU$ |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm 2 | 65 | 65 | 0.2457 | 0 | 1 | 0 | 0 | 22.03 | 0.80 |
| NC | 100 | 102 | 0.2016 | 1.50e-7 | 0.9900 | 0 | 0 | 90.06 | 5.45 |
| WC | 112 | 204 | 0.5442 | 2.12e-8 | 0.9911 | 0.0089 | 0 | 27.14 | 2.47 |
| Benson type | 115 | 232 | 0.5644 | 0.0724 | 1 | 0 | 0 | 54.56 | 4.09 |
| DE | 98 | - | 1.1542 | 0.3264 | 0.9900 | 0.0122 | 0 | 204.08 | 2.06 |
| SMS-MOEA | 100 | - | 0.2987 | 4.09e-4 | 1 | 0 | 0 | 200.00 | 18.35 |
| NSGA-II | 100 | - | 0.7012 | 0.8066 | 1 | 0 | 0 | 200.00 | 279.60 |

*4.2.3. Test problem 3*

This problem introduced by Tanaka et al. [57] has a non-convex and disconnected Pareto front and is given as follows:

$$\begin{aligned} min \quad & f(x) = (x_1, x_2) \\ s.t. \quad & x_1^2 + x_2^2 \geq 1 + 0.1cos(16arctan(\frac{x_1}{x_2})), \\ & (x_1 - 0.5)^2 + (x_2 - 0.5)^2 \leq 0.5, \\ & 0 \leq x_1, x_2 \leq \pi. \end{aligned}$$

The ideal and nadir points of Test problem 3 are (0.0417, 0.0417) and (1.0384, 1.0384), respectively. The second condition is considered in this test problem. Algorithm 2 with $\varepsilon = 0.05$ obtains 34 points on the Pareto front at 1.10 seconds, 1210 function evaluations, $EV = 0.2943$, and $EX = 0$. Figure 13 illustrates the feasible objective set of Test problem 3 and the points obtained by Algorithm 2 with $\varepsilon = 0.05$ and $\varepsilon = 0.02$. This figure also shows that the proposed algorithm is able to generate almost uniform approximations of the disconnected and non-convex Pareto front by selecting parameters of the PS approach suitably.

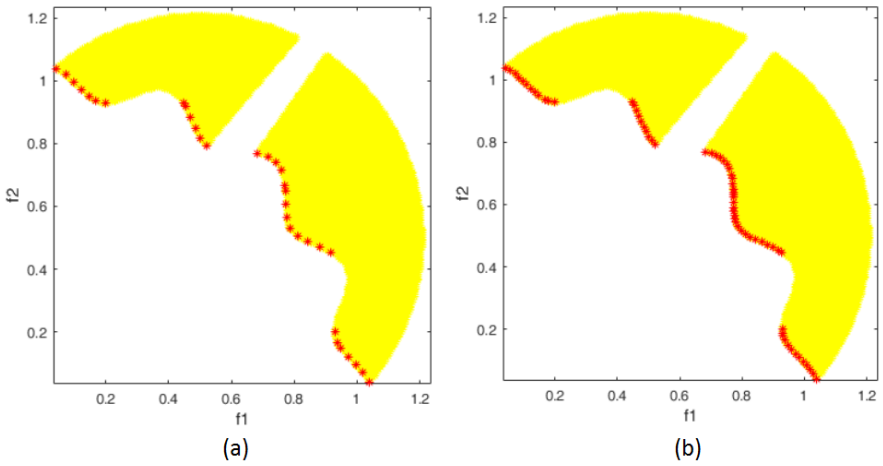


Figure 13: The approximations of the Pareto front of Test problem 3 obtained by Algorithm 2 for (a) $\varepsilon = 0.05$ and (b) $\varepsilon = 0.02$

Figure 14 indicates approximations of the Pareto front obtained by the Benson type algorithm with $\epsilon = 0.01$, $\hat{p} = (100, 100)$, and $d = (1, 1)$ and the other algorithms for Test problem 3. According to this figure, the WC and Benson type algorithms obtain some dominated points and the approximations obtained by the DE, SMS-EMOA, and NSGA-II algorithms cannot cover the entire Pareto front.

Table 5 presents the results of Algorithm 2 with $\varepsilon = 0.02$ and the other algorithms for Test problem 3. By attention to this table, Figure 13, and Figure 14, Algorithm 2 can obtain the high quality results for this problem. For example, the purity of Algorithm 2 is equal 0.9846 and none of the points obtained by Algorithm 2 is dominated by the points obtained by the other algorithms except the WC algorithm. In the WC algorithm, (0.68301469, 0.76924913) is only dominated by (0.68301468, 0.76924913), although these two points are much closer together. Furthermore, the quality of the distributed points obtained by Algorithm 2 is better than the other algorithms in terms of the percentage of non-dominated points and it has the least number of the function evaluations per each obtained solution, $EX$ and $EV$ among all the algorithms. Moreover, the CPU time of the DE algorithm is very close to Algorithm 2 while Algorithm 2 is better than the DE algorithm in the other indexes. It is concluded that the quality of the points obtained by Algorithm 2 and its performance is better than the other algorithms.

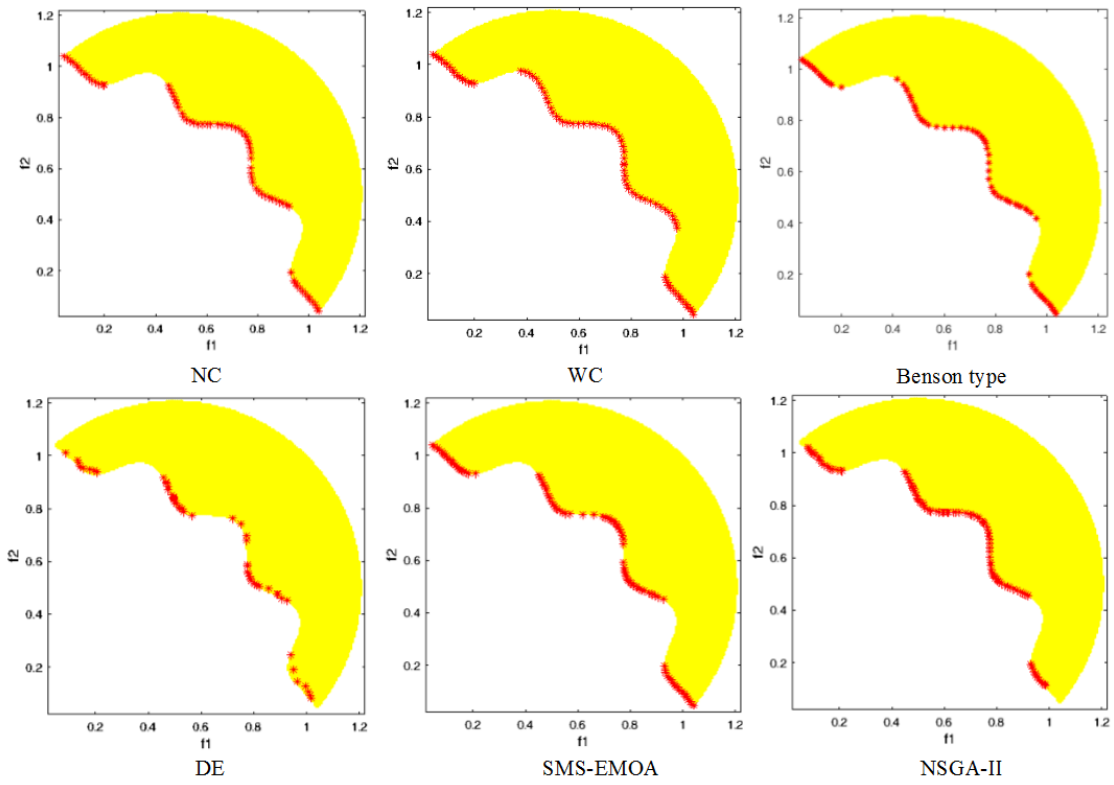


Figure 14: The approximations of the Pareto front of Test problem 3 obtained by the six algorithms

Table 5: The numerical results of the seven algorithms for Test problem 3

| Method | *NS* | *NSOP* | *EV* | *EX* | *P* | $C(Alg2, S)$ | $C(S, Alg2)$ | *FE/NS* | *CPU* |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm 2 | 65 | 68 | 0.2435 | 0 | 0.9846 | 0 | 0 | 21.46 | 1.53 |
| NC | 63 | 102 | 0.5418 | 4.62e-13 | 0.8710 | 0.1290 | 0 | 86.74 | 5.09 |
| WC | 106 | 204 | 0.6795 | 2.41e-13 | 0.8861 | 0.3113 | 0.01538 | 40.39 | 5.24 |
| Benson type | 112 | 226 | 0.9095 | 0.0036 | 0.8468 | 0.1607 | 0 | 31.78 | 5.63 |
| DE | 45 | - | 0.7665 | 0.0323 | 0.1333 | 0.6000 | 0 | 444.44 | 1.55 |
| SMS-MOEA | 100 | - | 0.7296 | 0.0023 | 0.6700 | 0.1200 | 0 | 200.00 | 14.52 |
| NSGA-II | 100 | - | 0.4218 | 0.0217 | 0.3200 | 0.2700 | 0 | 200.00 | 322.75 |

### 4.2.4. *Test problem 4*

This test problem with a non-convex and connected Pareto front is defined as follows:

$$\begin{aligned} &min \quad f_1(x) = x_1 \\ &min \quad f_2(x) = 1 + x_2^2 - x_1 - 0.1sin(3\pi x_1) \\ &s.t. \quad 0 \le x_1 \le 1, -2 \le x_2 \le 2. \end{aligned}$$

This bi-objective problem considered in [55] and its ideal and nadir points are $(0, 0)$ and $(1, 1)$, respectively. Figure 15 indicates points obtained of Algorithm2 by considering the first condition with $\varepsilon = 0.001$ and $\varepsilon = 0.0001$. Algorithm 2 can obtain 33 points on the Pareto front at 0.3797 seconds, 741 function evaluations, $EV = 0.2198$, and $EX = 0$ for $\varepsilon = 0.001$.

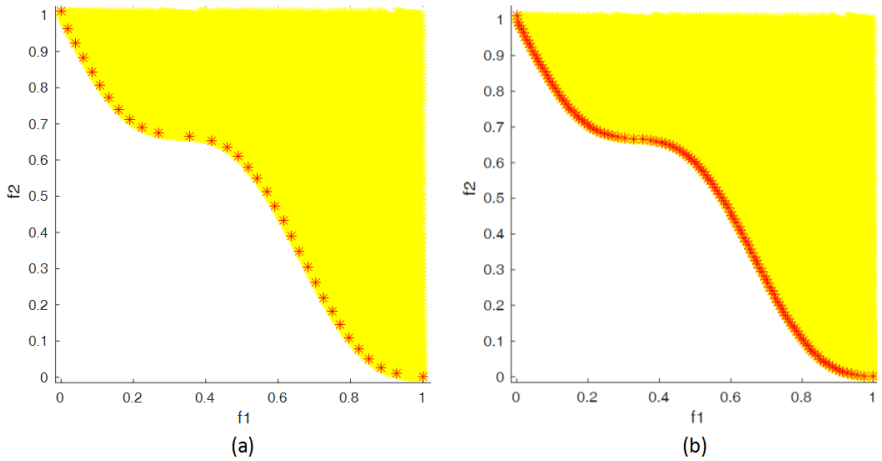


Figure 15: The approximations of the Pareto front of Test problem 4 obtained by Algorithm 2 for (a) $\varepsilon = 0.001$ and (b) $\varepsilon = 0.0001$

Figure 16 shows the distribution of points obtained by the other mentioned algorithms for Test problem 4, in which the Benson type algorithm considers $\epsilon = 0.01$, $\hat{p} = (100, 100)$, and $d = (1, 1)$. By comparing Figure 15 and Figure 16, it can be concluded that Algorithm 2 obtains a good distribution of the points on the Pareto front rather than the other algorithms.

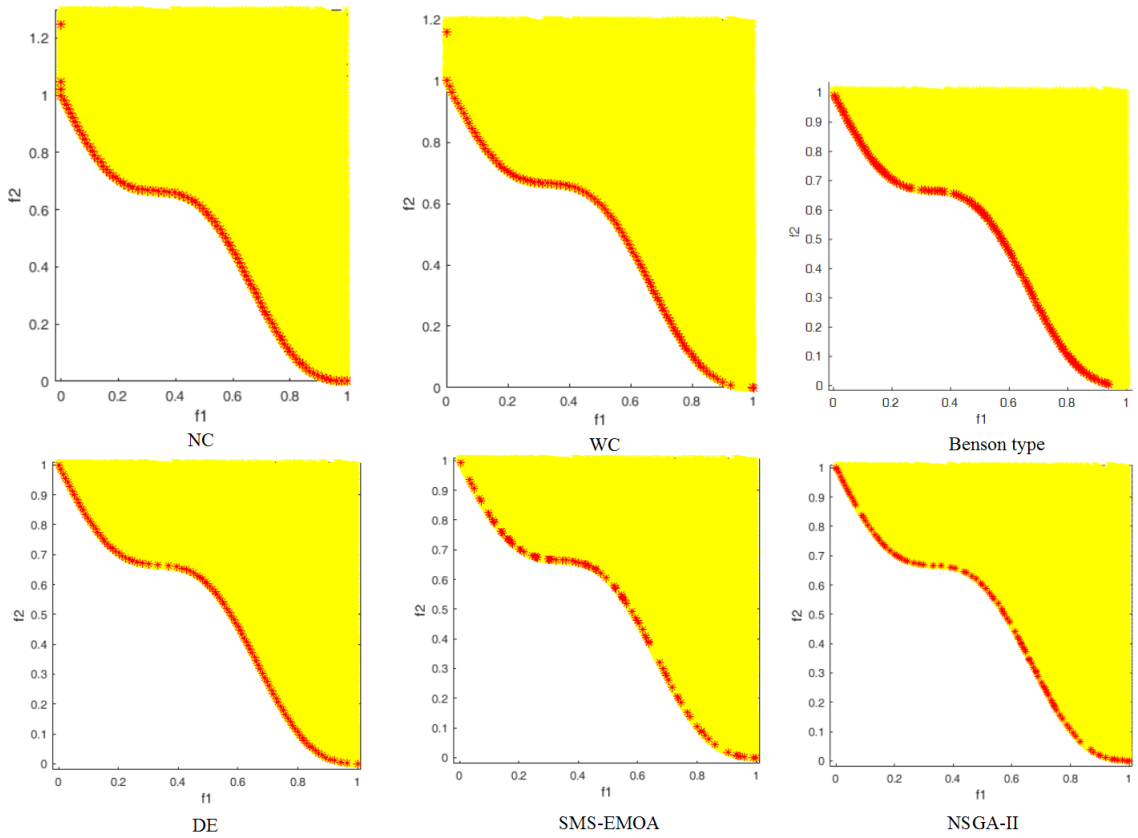


Figure 16: The approximations of the Pareto front of Test problem 4 obtained by the six algorithms

The results of Algorithm 2 with $\varepsilon = 0.0001$ and the other algorithms for Test problem 4 are given in Table 6. These results emphasize that Algorithm 2 can obtain the high quality results at all indexes. For example, Algorithm 2 runs in an acceptable time, none of the points obtained by Algorithm 2 is dominated by the points obtained by the other algorithms while the NC, WC, SMS-MOEA, and NSGA-II algorithms have some points dominated by some points obtained by Algorithm 2. Furthermore, values of $EV$ and $EX$ of the proposed algorithm are better than those of the other algorithms. Therefore, the performance of Algorithm 2 is acceptable among the other mentioned algorithms.

Table 6: The numerical results of the seven algorithms for Test problem 4

| Method | *NS* | *NSOP* | *EV* | *EX* | *P* | $C(Alg2, S)$ | $C(S, Alg2)$ | *FE/NS* | *CPU* |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm 2 | 129 | 129 | 0.2410 | 0 | 1 | 0 | 0 | 16.13 | 1.01 |
| NC | 97 | 102 | 1.1695 | 8.64e-4 | 0.9691 | 0.0300 | 0 | 2122 | 2.08 |
| WC | 103 | 204 | 1.1125 | 1.11e-16 | 0.9903 | 0.0096 | 0 | 4573 | 2.25 |
| Benson type | 158 | 318 | 0.4316 | 0.0303 | 1 | 0 | 0 | 57.06 | 8.83 |
| DE | 100 | - | 0.9588 | 0.0040 | 1 | 0 | 0 | 20000 | 1.13 |
| SMS-MOEA | 100 | - | 0.2653 | 5.88e-15 | 0.9800 | 0.0200 | 0 | 200.00 | 19.51 |
| NSGA-II | 100 | - | 0.5436 | 2.18e-8 | 0.9700 | 0.0155 | 0 | 200.00 | 189.67 |

*4.2.5. Test problem 5*

Test problem 5 named BNH has a convex and connected Pareto front and non-convex feasible set. This problem introduced by Binh and Korn [58] is defined as follows:

$$\begin{aligned} &min \quad f_1(x) = 4x_1^2 + 4x_2^2 \\ &min \quad f_2(x) = (x_1 - 5)^2 + (x_2 - 5)^2 \\ &s.t. \quad (x_1 - 5)^2 + x_2^2 \leq 25, \\ &\quad\quad (x_1 - 8)^2 + (x_2 + 3)^2 \geq 3, \\ &\quad\quad 0 \leq x_1 \leq 5, 0 \leq x_2 \leq 3. \end{aligned}$$

The ideal and nadir points are (0, 4) and (136, 50), respectively. Figure 17 shows points obtained by Algorithm 2 by considering the second condition with $\varepsilon = 2$ and $\varepsilon = 4$. Algorithm 2 obtains 55 points on the Pareto front at 1.14 seconds, 1134 function evaluations, $EV = 0.2081$, and $EX = 0$ for $\varepsilon = 4$.

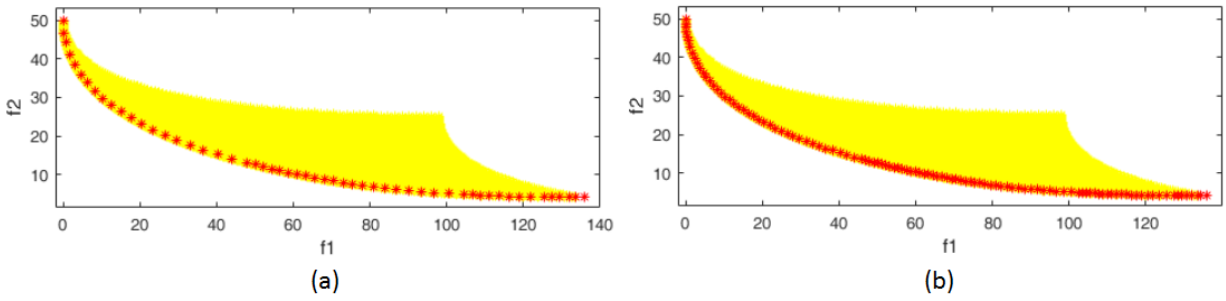


Figure 17: The approximations of the Pareto front of Test problem 5 obtained by Algorithm 2 for (a) $\varepsilon = 4$ and (b) $\varepsilon = 2$

Figure 18 demonstrates approximations obtained by the Benson type algorithm with $\epsilon = 1, \hat{p} = (500, 500)$, and $d = (1, 1)$ and the other mentioned algorithms. This figure shows that there are regions of the Pareto front such that the points obtained by the WC, Benson type, DE, SMS-EMOA, and NSGA-II algorithms are not evenly distributed on the regions.

The results of Algorithm 2 with $\varepsilon = 2$ and the other algorithms for Test problem 5 are given in Table 7. The results indicate that Algorithm 2 can obtain the best results with the least number of the function

evaluations per each obtained solution, shortest CPU time, and least values of $EX$ and $EV$ among all the algorithms.

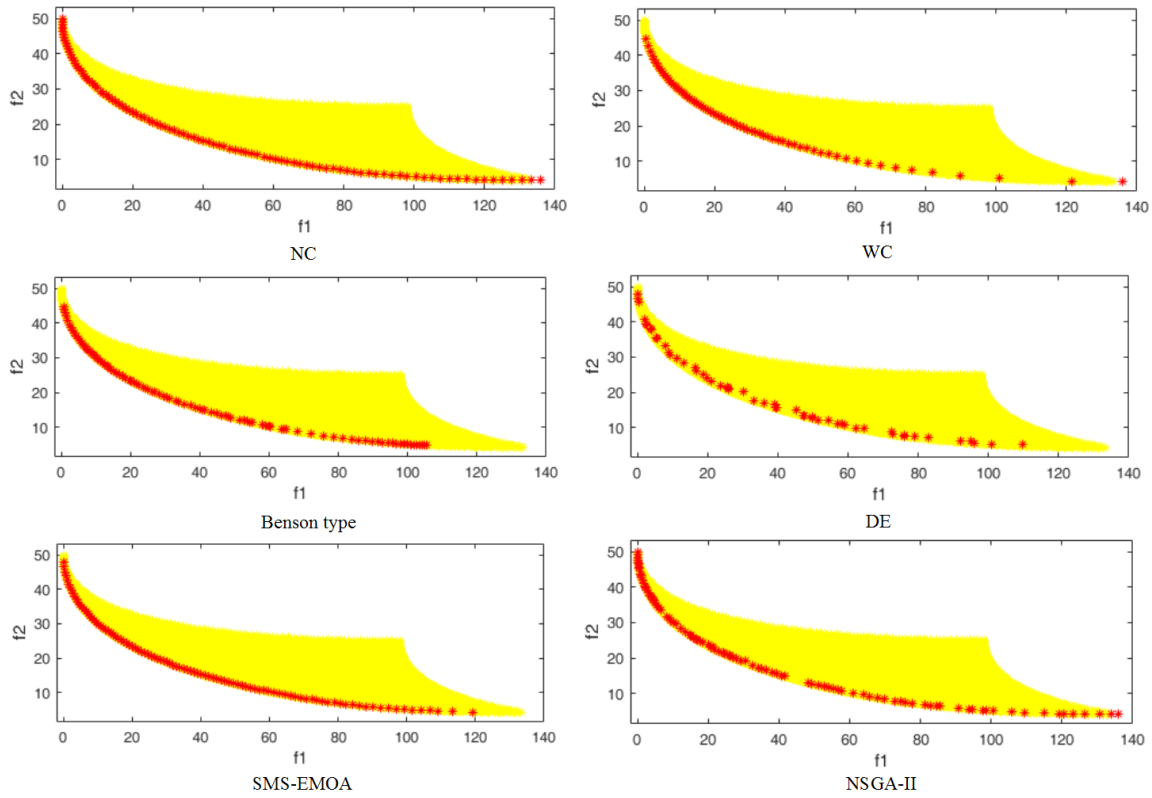


Figure 18: The approximations of the Pareto front of Test problem 5 obtained by the six algorithms

Table 7: The numerical results of the seven algorithms for Test problem 5

| Method | $NS$ | $NSOP$ | $EV$ | $EX$ | $P$ | $C(Alg2, S)$ | $C(S, Alg2)$ | $FE/NS$ | $CPU$ |
|---|---|---|---|---|---|---|---|---|---|
| Algorithm 2 | 110 | 110 | 0.2001 | 0 | 1 | 0 | 0 | 20.06 | 2.02 |
| NC | 100 | 102 | 0.3691 | 2.7908 | 0.9900 | 0 | 0 | 38.14 | 3.79 |
| WC | 188 | 204 | 2.2530 | 2.6075 | 1 | 0 | 0 | 82.77 | 6.85 |
| Benson type | 98 | 198 | 0.6055 | 15.4510 | 1 | 0 | 0 | 49.99 | 3.74 |
| DE | 47 | - | 1.2976 | 20.9472 | 0.2340 | 0.9787 | 0 | 425.53 | 0.65 |
| SMS-MOEA | 100 | - | 0.5539 | 8.4234 | 1 | 0 | 0 | 200.00 | 20.04 |
| NSGA-II | 100 | - | 0.8172 | 0 | 0.6200 | 0.4600 | 0 | 200.00 | 216.19 |

## 5. Algorithm 3 for solving nonlinear FP problems

In this section, the problem (1) is considered in the general case, i.e., the functions $f(x)$, $g(x)$, and $h_i(x)$, $i = 1, \ldots, m$, are nonlinear convex or non-convex functions. We describe how to adjust Algorithm 2 to solve the problem (1) and to obtain its $\varepsilon-$optimal solution. At first, calculate the ideal point, nadir point, and two anchor points of the bi-objective problem (3) corresponding to the problem (1) and consider a point $b$ such that $y^N \leqq b$. Two sets $A$ and $B$ are defined as same as Algorithm 2, i.e., $A$ consists of reference points and initially includes the ideal point and $B$ consists of points corresponding to the reference points and initially includes the point $b$. Algorithm 3 examines that the ideal point is feasible or not. If this point is feasible, it is a global optimal solution of the problem (5), otherwise the following steps are done.

Algorithm 3 begins with the ideal point and $b$, and consider $Y^0 = \{y' \in \mathbb{R}^2 \mid y^I + \mathbb{R}^2_{\leqq} \leqq y' \leqq b - \mathbb{R}^2_{\leqq}\}$ which $Y_N \subseteq Y_0$. Set $k := 0$ and consider $LB_0 := min\{Z(a) \mid a \in A\}$, and $UB_0 := Z(b)$ as an initial lower bound value and an initial upper bound value, respectively. In addition, the value of the function $Z$ of the problem (5) is calculated at the anchor points and a minimum of these two values is considered as $Z^*$. An approximation error $\varepsilon > 0$ is determined as the stop condition.

At iteration $k$, a reference point $a^k \in A$ and its corresponding point $b^k \in B$ are selected such that $Z(a^k) := min\{Z(a) \mid a \in A \text{ and } Z(a) < UB_k\}$. Set $A := A\backslash\{a^k\}$ and $B := B\backslash\{b^k\}$, and obtain $(x^k, t^k)$ as an optimal solution of $SP(a^k, r^k)$ in which $r^k := \frac{b^k - a^k}{\|b^k - a^k\|_2}$is the normalized direction vector. Then, a (weakly) non-dominated point $F(x^k) = (f(x^k), -g(x^k))$ and $y^k := a^k + t^k r^k$ are earned. If $Z(F(x^k)) < UB_k$, $UB_k$ will be updated to $Z(F(x^k))$. Besides, if $UB_k - LB_k \leq \varepsilon$ and $Z^* < UB_k$, then set $UB_k := Z^*$, an $\varepsilon-$optimal solution of the problem (5) is the anchor point corresponding to $Z^*$ and Algorithm 3 terminates. If $UB_k - LB_k \leq \varepsilon$ and $Z^* > UB_k$, then Algorithm 3 terminates. Otherwise, do the following steps:

- If $a^k < y^k$, then add new reference points $(y^k_1, a^k_2)$ and $(a^k_1, y^k_2)$ to $A$ and add their corresponding points $(b^k_1, F_2(x^k))$ and $(F_1(x^k), b^k_2)$, respectively, to $B$ and set $Y^k := Y^{k-1}\backslash R^k$ in which a rectangle $R^k$ is defined as $\{y' \in \mathbb{R}^2 \mid a^k + \mathbb{R}^2_{\leqq} \leqq y' \leqq y^k - \mathbb{R}^2_{\leqq}\}$. It should be noted that each reference point and its corresponding point are distinct from $a^k$ and $b^k$, respectively.

- Set $LB_k = min\{Z(a) \mid a \in A\}$. If $UB_k - LB_k \leq \varepsilon$ and $Z^* < UB_k$, then set $UB_k := Z^*$ and an $\varepsilon-$optimal solution of the problem (5) is the anchor point corresponding to $Z^*$ and Algorithm 3 terminates. If $UB_k - LB_k \leq \varepsilon$ and $Z^* > UB_k$, then Algorithm 3 terminates. Otherwise, set $k : k + 1$ and the algorithm will be repeated.

In Algorithm 3 summarized as follows, $x^c$ is an $\varepsilon-$optimal solution of the problem (1), $y^c$ is an $\varepsilon-$optimal solution of the problem (5), and $Z(y^c)$ is an $\varepsilon-$approximate of the optimal objective value of the problem (1).

**Algorithm 3: an algorithm for solving nonlinear FP problems**

**Initialisation:**

- Choose $\varepsilon > 0$, the ideal point, nadir point, and two anchor points are obtained and select $b$ such that $y^N \leqq b$. If the ideal point is feasible, then set $y^c := y^I$ and a global optimal solution of the problem (1) is a solution corresponding to $y^I$ in the decision space.

- Calculate $Z^*$ that $x'$ and $y'$ are solutions corresponding to $Z^*$ in the decision space and objective space, respectively. Set $LB_0 := Z(y^I)$, $UB_0 := Z(b)$, $A := \{y^I\}$, $B := \{b\}$, and $k := 0$.

**Main iteration repeat:**

1. Choose $a^k \in A$ and its corresponding point $b^k \in B$ such that $Z(a^k) := min\{Z(a) \mid a \in A \text{ and } Z(a) < UB_k\}$. Set $r^k := \frac{b^k - a^k}{\|b^k - a^k\|_2}$. Obtain the points $F(x^k) := (f(x^k), -g(x^k))$ and $y^k := a^k + t^k r^k$ such that $(x^k, t^k) := argSP(a^k, r^k)$. Set $A := A\backslash\{a^k\}$ and $B := B\backslash\{b^k\}$.
2. If $Z(F(x^k)) < UB_k$, then set $UB_k := Z(F(x^k))$, $y^c := F(x^k)$, and $x^c := x^k$.
3. If $UB_k - LB_k \leq \varepsilon$ and $Z^* < UB_k$, then set $UB_k := Z^*$, $y^c := y'$, $x^c := x'$ and stop. If $UB_k - LB_k \leq \varepsilon$ and $UB_k \leq Z^*$, then stop. Otherwise do Step 4.
4. If $a^k \leq y^k$, then reference points $(y^k_1, a^k_2)$ and $(a^k_1, y^k_2)$ are added to $A$ and their corresponding points $(b^k_1, F_2(x^k))$ and $(F_1(x^k), b^k_2)$, respectively, are added to $B$. Set $LB_k := min\{Z(a) \mid a \in A\}$. If $UB_k - LB_k \leq \varepsilon$, then check Step 3. Otherwise, set $k := k + 1$, $UB_k := UB_{k-1}$, $LB_k := LB_{k-1}$, and go to Step 1.

In Algorithm 3, $Z(a^k)$ is used to update the lower bound values and $Y_N \subseteq Y^k$ for each $k$. A minimum value of the function $Z$ on the set $Y^k$ is related to a reference point in $A$. Therefore, it is clear that $LB_k$ is always a lower bound for the objective function value of the problem (1) and $Y^k \subseteq Y^{k-1}$ for each iteration $k$. As a result, $LB_{k-1} \leq LB_k$. In addition, $Z(F(x^k))$ is used to update the upper bound value, and $UB_k$ is always an upper bound for the objective function value of the problem (1) and $UB_k \leq UB_{k-1}$ at iteration $k$.

**Theorem 5.1.** *Let the functions $f(x)$ and $-g(x)$ be continuous and bounded from below on the set $X$ and Algorithm 3 be infinite. Then, it generates a sequence $\{F(x^k)\}$ of the feasible solutions of the problem (5) such that there exists at least an accumulation point and every accumulation point is a global optimal solution of the problem (5) and*

$$\lim_{k\to\infty} UB_k = \lim_{k\to\infty} LB_k = inf\{Z(y) \mid y \in Y'\}.$$

*Proof. Since Algorithm 3 is infinite, it generates an infinite sequence $\{F(x^k)\}$ such that $F(x^k) \in Y'_N$. In addition, by considering $F(x^k) \in Y^0$, this infinite sequence is bounded and it has at least an accumulation point such as $F^*$. It will prove that $F^*$ is a global optimal solution of the problem (5).*

*For this purpose, it is shown that sequences $\{a^k\}$ of reference points, $\{r'^k\}$ of normalized direction vectors and $\{t'^k\}$ of real numbers are constructed such that $F(x^k) = a^k + t'^k r'^k$. At iteration $k$, if $F(x^k)$ is along $r^k$, then $F(x^k) = a^k + t^k r^k$ and set $r'^k = r^k$ and $t'^k = t^k$. If $F(x^k)$ is not along $r^k$, i.e., $F(x^k) \leq a^k + t^k r^k$, then there are a vector $r'^k$ and a real number $t'^k$ such that $t'^k r'^k \leq t^k r^k$ and $F(x^k) = a^k + t'^k r'^k$. On the other hand, $\{R^k\}$ is a bounded sequence of distinct sets in the rectangle $Y^0$. Consider $s_k$ as the area of the rectangle $R^k$. In this case, since $\sum_{k=1}^{\infty} s_k$ is smaller than the area of $Y_0$, $\sum_{k=1}^{\infty} s_k$ is convergent and hence, $\lim_{k\to\infty} s_k = 0$. Since $r'^k$ is the normalized vector, when $k \to \infty$, $t'^k r'^k \to 0_2$. By according to the convergence of the sequence $\{F(x^k)\}$, $\{a^k\}$ is convergent. Let $\{a^k\}$ converge into $a^*$, then $F^* = a^* \in Y'$. Furthermore, the sequence $\{Z(a^k)\}$ is non-decreasing and bounded and the sequence $\{Z(F(x^k))\}$ is non-increasing and bounded in Algorithm 3. Therefore, these sequences are convergent and $\lim_{k\to\infty} UB_k = \lim_{k\to\infty} LB_k = Z(F^*)$.*

*On the other hand, if $\hat{y}$ be a global optimal solution of the problem (5), then $LB_k \leq Z(\hat{y}) \leq UB_k$. Therefore, $Z(\hat{y}) = Z(F^*)$, $Z(F^*) = inf\{Z(y) \mid y \in Y'\}$, and the proof is completed.* □

**Theorem 5.2.** *If $\varepsilon > 0$, then Algorithm 3 is terminated after a finite number of iterations at an $\varepsilon-$optimal solution of the problem (5).*

*Proof. Assume $\varepsilon > 0$ and this algorithm is infinite. In this case, according to Theorem 5.1:*

$$\lim_{k\to\infty} UB_k = \lim_{k\to\infty} LB_k = inf\{Z(y) \mid y \in Y'\}.$$

*Since $\varepsilon > 0$, according to steps 3 and 4 of Algorithm 3, there is an integer number $\bar{k}$ sufficiently large such that $UB_{\bar{k}} - LB_{\bar{k}} < \varepsilon$. In this case, Algorithm 3 terminates and it is a contradiction to the assumption that the algorithm is infinite. Since the proposed algorithm is finite, assume that it stops at the $k^{th}$ iteration. According to the steps 3 and 4 from the termination condition: $UB_k - LB_k < \varepsilon$. Therefore, $Z(F(x^k)) < \varepsilon + LB_k$.*

*Let $y^*$ be an optimal solution of the problem (5), then $LB_k \leq Z(y^*) \leq Z(F(x^k)) = UB_k$ and as a result, $Z(y^*) \leq Z(F(x^k)) < \varepsilon + LB_k \leq \varepsilon + Z(y^*)$. Therefore,*

$$Z(y^*) \leq Z(F(x^k)) \leq \varepsilon + Z(y^*),$$

*and it means $F(x^k)$ is an $\varepsilon-$optimal solution of the problem (5) and $x^k$ is an $\varepsilon-$optimal solution of the problem (1).* □

It is worth noting that Algorithm 3 has some advantages to Algorithm 1 such as:

1. Algorithm 3 does not need to assume that the functions $f(x)$, $-g(x)$, and $h_i(x)$, $i = 1, \ldots, m$, are convex and continuously differentiable on $X$.
2. Algorithm 3 does not need to specify vertices of polyhedrons and it leads to improve in the CPU times.
3. Algorithm 1 solves two SOPs at each iteration, while Algorithm 3 considers only an SOP at each iteration.
4. Algorithm 3 can be applied to non-convex FP problems.

### *5.1. Numerical results of Algorithm 3*

In this section, the efficiency of Algorithm 3 is examined by six numerical examples. At first, Example 3.3 is considered. Algorithm 2 obtains 45 points of a bi-objective problem corresponding to Example 3.3 at 0.31 seconds and 702 function evaluations by considering the second condition with $\varepsilon = 0.2$. Figure 19 indicates that the points obtained by Algorithm 2 covering the entire Pareto front.

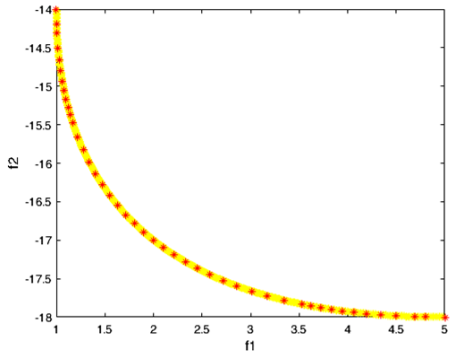


Figure 19: The approximation of the Pareto front of the bi-objective problem corresponding to Example 3.3 obtained by Algorithm 2

Algorithm 3 obtains an $\varepsilon-$optimal solution of this problem at five iterations by considering $\varepsilon = 0.0025$, $b = y^N$, $LB_0 = 0.0556$, $UB_0 = 0.3571$, and $Z^* = 0.071429$. Figure 20 displays the set $A$, set $B$, and point $F(x^k)$ of Algorithm 3 at four iterations. After five iterations, the upper bound value is 0.0701161, which is less than $Z^*$, the lower bound value is 0.0684148 and the difference between these two bounds is less than 0.0025. Therefore, $y^c = (1.0168, -14.5018)$ and $x^c = 0.1296$ are $\varepsilon-$ optimal solutions of this example in the decision space and the objective space, respectively. The CPU time is 0.0611 seconds.

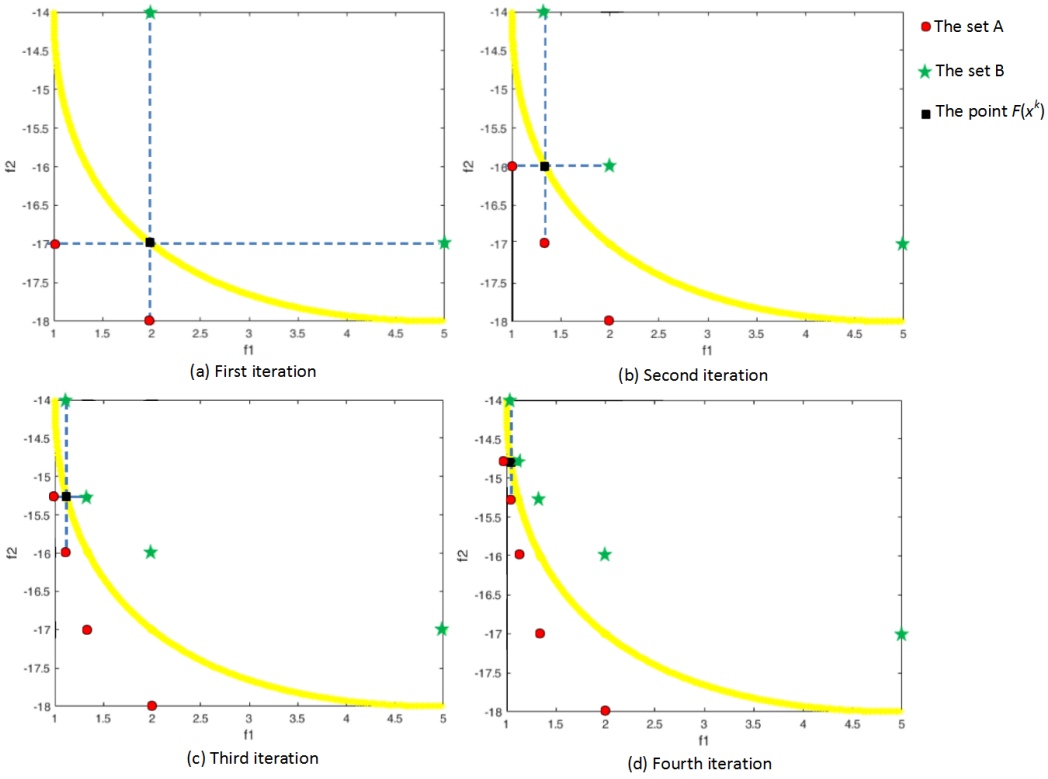


Figure 20: Illustration of four iterations of Algorithm 3 for Example 3.3

Table 8 reports the reference point $a^k$, direction vector $r^k$, point $F(x^k)$, $UB_k$, and $LB_k$ at five iterations. As seen in this table, the upper and lower bound values are very close to each other at each iteration of the algorithm.

Table 8: The results obtained by Algorithm 3 at five iterations for Example 3.3

| Iteration | $y^k$ | $a^k$ | $F(x^k)$ | $UB_k$ | $LB_k$ |
|---|---|---|---|---|---|
| k=0 | (0, -18) | (0.7071, 0.7071) | (2, -17) | 0.1176 | 0.0588 |
| k=1 | (1, -17) | (0.3162, 0.9487) | (1.3377, -15.9868) | 0.0837 | 0.0626 |
| k=2 | (1, -15.9868) | (0.1676, 0.9859) | (1.1214, -15.2724) | 0.0734 | 0.0655 |
| k=3 | (1, -15.2724) | (0.0950, 0.9955) | (1.0449, -14.8024) | 0.0706 | 0.0676 |
| k=4 | (1, -14.8024) | (0.0558, 0.9984) | (1.0168, -14.5018) | 0.0701 | 0.0684 |

Table 9 shows the results of Algorithm 1 and Algorithm 3 for Example 3.3. According to this table, Algorithm 3 is able to find a better objective value of Example 3.3 than Algorithm 1 and the number of SOPs and CPU time of Algorithm 3 are less than Algorithm 1.

Table 9: Comparing the results obtained by Algorithm 1 and Algorithm 3 for Example 3.3

| Algorithm | $Z(y^c)$ | NS | NSOP | CPU |
|---|---|---|---|---|
| Algorithm 1 | 0.070132 | 4 | 12 | 1.22 |
| Algorithm 3 | 0.070116 | 5 | 9 | 0.06 |

Now consider Example 3.4. Algorithm 2 obtains 33 points of a bi-objective problem corresponding to Example 3.4 at 0.23 seconds and 600 function evaluations by considering the first condition with $\varepsilon = 10$. Figure 21(a) demonstrates that Algorithm 2 generates an approximation of the Pareto front which its points cover the entire Pareto front almost uniformly. In addition, Algorithm 3 considers $\varepsilon = 0.001$, $b = (230, -10)$, and $Z^* = 3.217391$. An upper bound value in the ninth iteration is 3.217392, which is greater than $Z^*$. Therefore, $UB_9 = 3.217391$ is an $\varepsilon-$approximate of the optimal objective function value of Example 3.4 and $(0.5, 5)$ is an $\varepsilon-$optimal solution of this example. Furthermore, the CPU time is 0.1623 seconds. Figure 21(b) illustrates points $F(x^k)$ obtained by Algorithm 2 in the objective space.

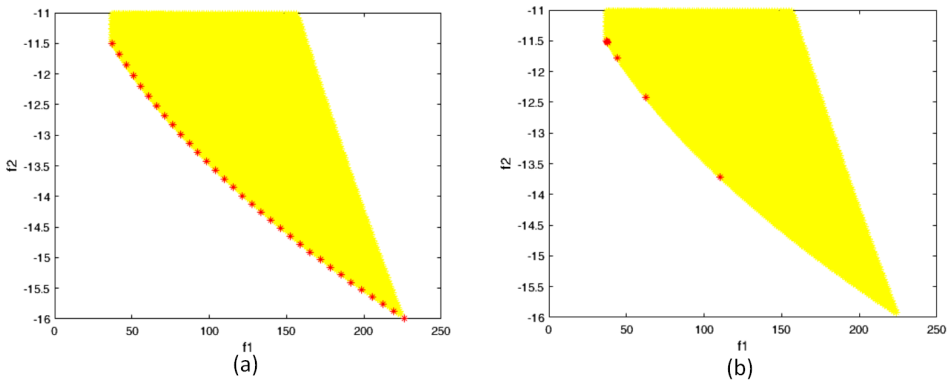


Figure 21: (a) The approximation of the Pareto front of the bi-objective problem corresponding to Example 3.4 obtained by Algorithm 2 and (b) points $F(x^k)$ derived from Algorithm 3 for Example 3.4

Table 10 reports the results of Algorithm 1 and Algorithm 3 for Example 3.4. Based on these results, they obtain the same solution for this example and the CPU time of Algorithm 3 is less than that of Algorithm 1.

Table 10: Comparing the results obtained by Algorithm 1 and Algorithm 3 for Example 3.4

| Algorithm | $Z(y^c)$ | NS | NSOP | CPU |
|---|---|---|---|---|
| Algorithm 1 | 3.217391 | 4 | 12 | 3.23 |
| Algorithm 3 | 3.217391 | 9 | 13 | 0.16 |

Figure 22(a) shows an almost uniform approximation of the Pareto front of a bi-objective problem corresponding to Example 3.5 obtained by Algorithm 2. It obtains 34 points at 0.23 seconds and 746 function evaluations by considering the second condition with $\varepsilon = 0.7$. In addition, Algorithm 3 considers

$b = y^N = (13, -1.12)$, $\varepsilon = 0.001$, and $Z^* = 0.178571$. It obtains 0.162279 as an $\varepsilon-$optimal value of the objective function corresponding to $y^c = (0.2195, -1.3501)$ and $x^c = (0.4617, 1.5382)$ at 0.35 seconds and 37 iterations. Figure 22(b) shows points $F(x^k)$ obtained by the proposed Algorithm 3 in the objective space.

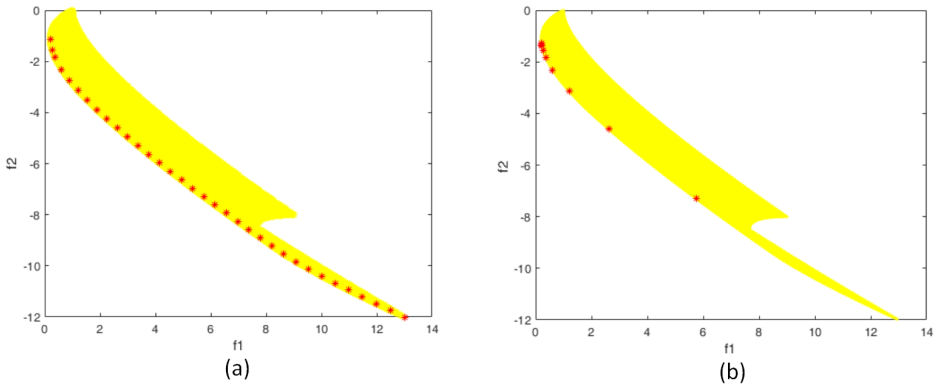


Figure 22: (a) The approximation of the Pareto front of the bi-objective problem corresponding to Example 3.5 obtained by Algorithm 2 and (b) points $F(x^k)$ derived from Algorithm 3 for Example 3.5

The results of Algorithm 1 and Algorithm 3 for Example 3.5 are shown in Table 11. As seen in this table, Algorithm 3 can obtain a better objective function value and less CPU time than Algorithm 1. Besides the number of SOPs of Algorithm 1 is less than that of Algorithm 3.

Table 11: Comparing the results obtained by Algorithm 1 and Algorithm 3 for Example 3.5

| Algorithm | $Z(y^c)$ | NS | NSOP | CPU |
|---|---|---|---|---|
| Algorithm 1 | 0.162305 | 6 | 10 | 2.64 |
| Algorithm 3 | 0.162278 | 37 | 41 | 0.35 |

**Example 5.3.** *Consider the following FP problem:*

$$\begin{aligned} min \quad & \frac{x_1^2 + 1}{x_2^2 + 1} \\ s.t. \quad & x_1^2 + 2x_3 \geq 1, \\ & 0 \leq x_1, x_2, x_3 \leq 2. \end{aligned}$$

*In a bi-objective problem related to this example, (0, 2, 0.5) is a solution corresponding to $y^I = (1, -5)$, in the decision space. Since this solution is a feasible point, it is an optimal solution of Example 5.3 which its objective function value is 0.2.*

**Example 5.4.** *Consider the following concave FP problem:*

$$\begin{aligned} min \quad & \frac{f(x)}{g(x)} = \frac{2x_2}{x_2^2 + 1} \\ s.t. \quad & 0 \leq 2x_1 + x_2^2 \leq 2, \\ & x_1, x_2 \geq 0.01, \end{aligned}$$

*where $f(x)$ is an affine function and $g(x)$ is a convex function. Algorithm 2 obtains 19 points of a bi-objective problem corresponding to Example 5.4 at 0.13 seconds and 353 function evaluations by considering $y^I = (0.0200, -2.9800)$ and $y^N = (2.8142, -1.0000)$, and the second condition with $\varepsilon = 0.25$, see Figure 23(a). In addition, Algorithm 3*

*considers* $LB_0 = 0.0067$, $UB_0 = 2.8140$, $\varepsilon = 0.0025$, $b = (2.8142, -1)$, *and* $Z^* = 0.019998$. *An upper bound value at the fourth iteration is 0.020003, which is greater than* $Z^*$. *Hence, Algorithm 3 earns* $Z^*$ *as an* $\varepsilon-$*approximate of the optimal objective function value of this example corresponding to* $y^c = (0.0200, -1.0001)$ *and* $x^c = (0.9999, 0.0100)$ *at 0.0457 seconds and four iterations. Figure 23(b) shows points* $F(x^k)$ *obtained by Algorithm 3 in the objective space.*

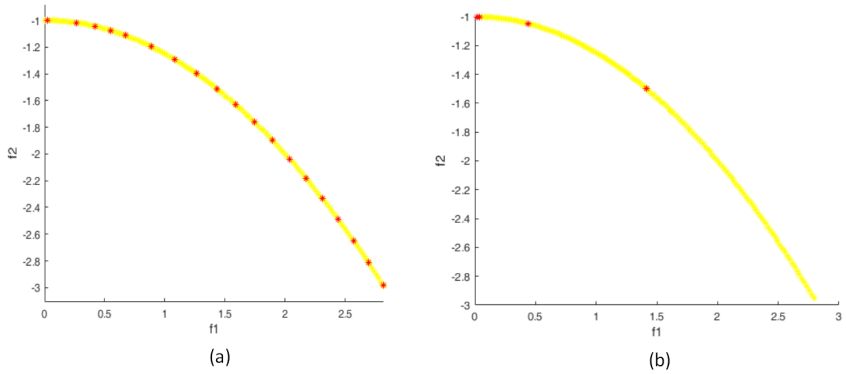


Figure 23: (a) The approximation of the Pareto front of the bi-objective problem corresponding to Example 5.4 obtained by Algorithm 2 and (b) points derived from Algorithm 3 for Example 5.4

**Example 5.5.** *Consider the following nonlinear FP problem:*

$$\begin{aligned} min \quad & \frac{f(x)}{g(x)} = \frac{x_1 + x_2 + 1}{6x_1^2 + e^{x_2}} \\ s.t. \quad & x_2^2 + x_2 \geq x_1^2, \\ & x_1, x_2 \geq 1, \end{aligned}$$

*where* $f(x)$ *is an affine function,* $g(x)$ *is a convex function and the constraint function* $h(x) = x_1^2 - x_2^2 - x_2$ *is non-convex. The points* $(1, -7.8553)$ *and* $(2.6180, -1)$ *are the ideal and nadir points of a bi-objective problem corresponding to Example 5.5, respectively. Algorithm 2 obtains 33 points at 0.21 seconds and 664 function evaluations by considering the second condition with* $\varepsilon = 0.025$. *Figure 24(a) illustrates the points obtained by Algorithm 2 which cover the entire Pareto front almost uniformly. Algorithm 3 considers* $LB_0 = 0.1273$, $UB_0 = 2.6180$, $\varepsilon = 0.05$, $b = y^N$, *and* $Z^* = 1$. *In addition, it obtains an* $\varepsilon-$*optimal objective function value equal to 0.286264,* $y^c = (1.9984, -6.9810)$, *and* $x^c = (0.9984, 0)$. *Besides, this algorithm finds this solution at 0.0220 seconds and three iterations. Figure 24(b) shows points* $F(x^k)$ *obtained by Algorithm 3 for Example 5.5 in the objective space.*

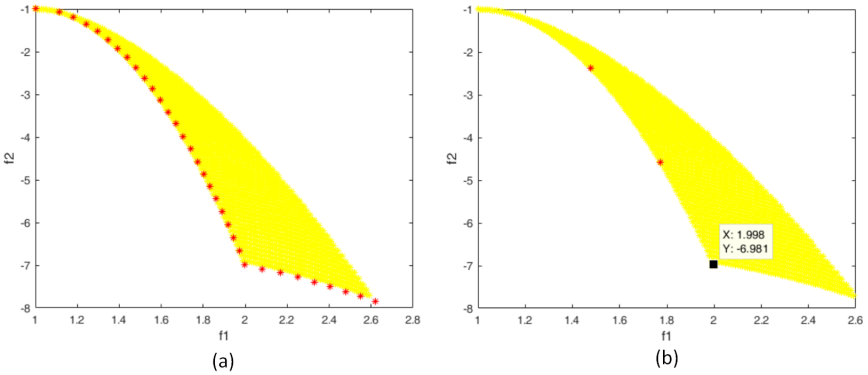


Figure 24: (a) The approximation of the Pareto front of the bi-objective problem corresponding to Example 5.5 obtained by Algorithm 2 and (b) points $F(x^k)$ derived from Algorithm 3 for Example 5.5

*Figure 25 displays the set A, set B, and point $F(x^k)$ of Algorithm 3 obtained at three iterations.*

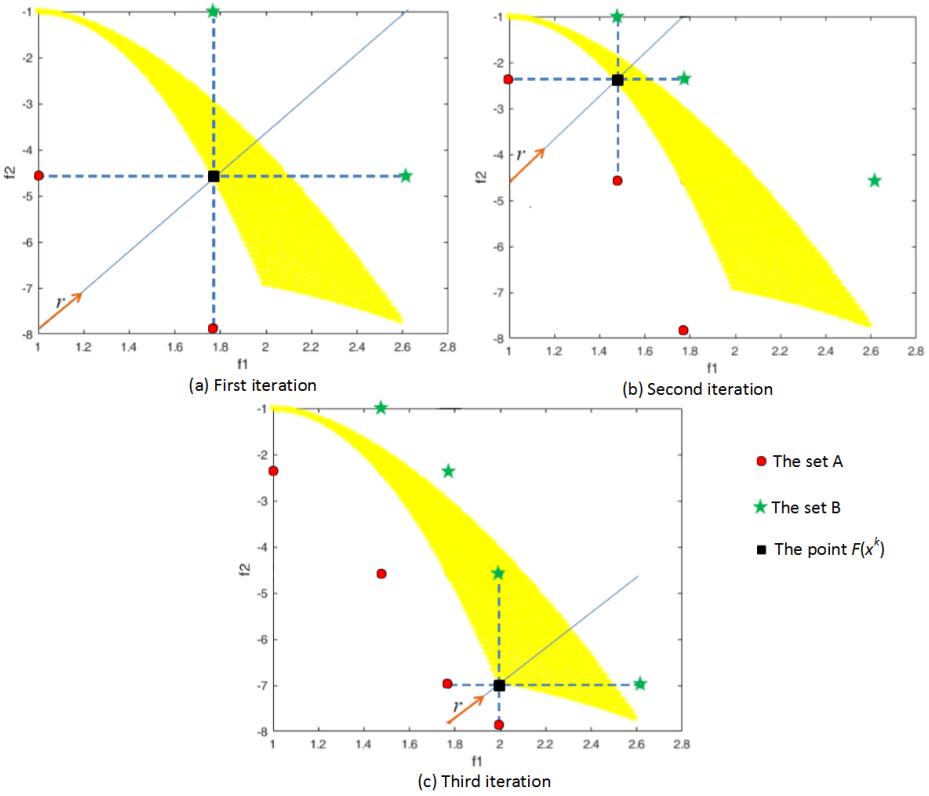


Figure 25: Illustration of three iterations of Algorithm 3 for Example 5.5

**Remark 5.6.** *Algorithm 3 may obtain a weakly non-dominate point $y = (y_1, y_2)$ which it is not a non-dominate point and is an $\varepsilon-$optimal solution of the problem (5), see Theorem 5.2. Besides, it is obvious that there is a non-dominate point such as $y'$ such that $y' \leq y$ and $Z(y') < Z(y)$. Consider the following bi-objective optimization problem:*

$$\begin{aligned} min \quad & f(x) = (f_1(x), f_2(x))^T \\ s.t. \quad & x \in X. \end{aligned}$$

*For obtaining the point $y'$, it is enough two following problems $P_1$ and $P_2$ are solved:*

$$\begin{aligned} min \quad & f_1(x) \\ s.t. \quad & f_1(x) \leq y_1, \qquad (P_1) \\ & f_2(x) \leq y_2, \\ & x \in X. \end{aligned}$$

$$\begin{aligned} min \quad & f_2(x) \\ s.t. \quad & f_1(x) \leq y_1, \qquad (P_2) \\ & f_2(x) \leq y_2, \\ & x \in X. \end{aligned}$$

*By solving the problems $P_1$ and $P_2$, two optimal solutions $x^1$ and $x^2$ are obtained, respectively. Then, between the points $y^1 = f(x^1)$ and $y^2 = f(x^2)$, a point which do not dominate by another point is the point $y'$, see Figure 26.*

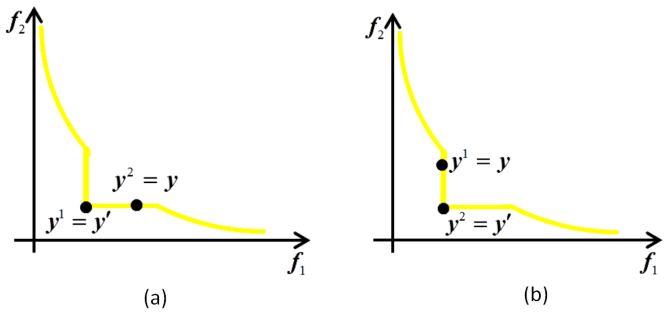


Figure 26: Solving the problems $P_1$ and $P_2$ and obtaining the points $y^1$ and $y^2$ for a bi-objective problem

## 6. Conclusions

In this paper, it was shown that an optimal solution of a FP problem is a Pareto optimal solution of its corresponding bi-objective problem. Then, Algorithm 1 and Algorithm 3 were presented to solve FP problems and proved that these algorithms guarantee to find an $\varepsilon-$optimal solution with considering a specified approximation error $\varepsilon$. Algorithm 1 is based on the cut and bound method for solving convex FP problems and assumed that the objective functions and constraint functions are convex and continuously differentiable on $X$. Then, Algorithm 2 based on the PS approach was proposed to obtain approximations of the Pareto front of bi-objective optimization problems with convex, non-convex, connected, and disconnected Pareto fronts. In addition, it was shown that Algorithm 2 is able to find a set of non-dominated points that covers almost uniformly the entire Pareto front. The results of Algorithm 2 on five test problems indicated that approximation obtained by this algorithm has a better quality than the NC, WC, Benson type, DE, NSGA-II, and SMS-EMOA algorithms. Finally, Algorithm 3 based on Algorithm 2 was presented to solve convex and non-convex FP problems for the first time.

## References

[1] J.B.G. Frenk, and S. Schaible, Fractional programming in: N. Hadjisavvas, S. Komlosi, and S. Schaible, Handbook of Generalized Convexity and Generalized Monotonicity, Springer, Boston (2005) 335–386.
[2] I.M. Stancu-Minasian, Fractional programming: theory, methods, and applications, Kluwer Academic Publishers, Dordrecht, 1997.
[3] S. Schaible, Minimization of ratios, Journal of Optimization Theory and Applications 19.2 (1976) 347–352.
[4] M. Avriel, W. E. Diewert, S. Schaible, and I. Zang, Generalized Concavity, New York, Plenum, 1988.
[5] O.L. Mangasarian, Nonlinear Programming, New York, McGraw-Hill, 1969.
[6] A. Charnes, and W. W. Cooper, Programming with linear fractional functionals, Naval Research Logistics Quarterly 9.3-4 (1962) 181–186.
[7] M. Borza, A. S. Rambely, and M. Saraj, Solving linear fractional programming problems with interval coefficients in the objective function. A new approach, Applied Mathematical Sciences 6 (2012) 3443–3459.
[8] B. Martos, and V. Whinston, Hyperbolic programming, Naval Research Logistics Quarterly 11.2 (1964) 135–155.
[9] W. S. Dorn, Linear fractional programming, International Business Machines Corporation, Thomas J. Watson Research Center, 1962.
[10] K. Swarup, Linear fractional functional programming, Operations Research 13 (1965) 1029–1036.
[11] H. M. Wagner, and J. S. Yuan, Algorithmic equivalence in linear fractional programming, Management Science 14.5 (1968) 301–306.
[12] J. K. Sharma, A. K. Gupta, and M. P. Gupta, Extension of simplex technique for solving fractional programming problems, Indian Journal of Pure and Applied Mathematics 11 (1980) 961–968.
[13] P. Pandey, and A. P. Punnen, A simplex algorithm for piecewise-linear fractional programming problems, European Journal of Operational Research 178.2 (2007) 343–358.
[14] H. Wolf, A parametric method for solving the linear fractional programming problem, Operations Research 33.4 (1985) 835–841.
[15] J. Isbell, and W. Marlow, Attrition games, Naval Research Logistics Quarterly 3.1-2 (1956) 71–94.
[16] G. R. Bitran, and A. G. Novaes, Linear programming with a fractional objective function, Operations Research 21.1 (1973) 22–29.
[17] M. B. Hasan, and S. Acharjee, Solving LFP by converting it into a single LP, International Journal of Operations Research 8.3 (2011) 1–14.

[18] S. F. Tantawy, Using feasible directions to solve linear fractional programming problems, Australian Journal of Basic and Applied Sciences 1.2 (2007) 109–114.
[19] S. F. Tantawy, A new procedure for solving linear fractional programming problems, Mathematical and Computer Modelling 48.5-6 (2008) 969–973.
[20] A. O. Odior, An approach for solving Iinear fractional programming problems, International Journal of Engineering and Technology 1.4 (2012) 298–304.
[21] K. Swarup, Programming with quadratic fractional functionals, Opsearch 2 (1965) 23–30.
[22] A. Beck, and M. Teboulle, A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid, Mathematical programming 118.1 (2009) 13–35.
[23] A. Khurana, and S. R. Arora, An algorithm for solving quadratic fractional program with linear homogeneous constraints, Vietnam Journal of Mathematics 39.4 (2011) 391–404.
[24] A. Zhang, and S. Hayashi, Celis-Dennis-Tapia based approach to quadratic fractional programming problems with two quadratic constraints, Numer. Algebra Control Optim 1.1 (2011) 83–98.
[25] N. A. Suleiman, and M. A. Nawkhass, Solving quadratic fractional programming problem, International Journal of Applied Mathematics Research 2.2 (2013) 303–309.
[26] P. Wolfe, The simplex method for quadratic programming, Econometrica: Journal of the Econometric Society (1959) 382–398.
[27] N. A. Suleiman, and M. A. Nawkhass, A new modified simplex method to solve quadratic fractional programming problem and compared it to a traditional simplex method by using pseudoaffinity of quadratic fractional functions, Applied Mathematical Sciences 7.76 (2013) 3749–3764.
[28] K. C. Sharma, and J Singh, Solution methods for linear factorized quadratic optimization and quadratic fractional optimization problem, IOSR Journal of Mathematics (IOSR-JM) 8.3 (2013) 81–86.
[29] M. Jayalakshmi, On solving linear factorized quadratic fractional programming problems, International Journal of Applied Research 9 (2015) 1037–1040.
[30] V. B. Nguyen, R. L. Sheu, and Y. Xia, An SDP Approach For Solving Quadratic Fractional Programming Problems, arXiv preprint arXiv:1402.4198 (2014).
[31] O.L. Mangasarian, Nonlinear fractional programming, Journal of the Operations Research Society of Japan 12 (1969) 1–10.
[32] M. Frank, and P Wolfe, An algorithm for quadratic programming, Naval Research Logistics Quarterly 3.1-2 (1956) 95–110.
[33] W. Dinkelbach, On nonlinear fractional programming, Management science 13.7 (1967) 492–498.
[34] S. Schaible, Parameter-free convex equivalent and dual programs of fractional programming problems, Zeitschrift für Operations Research 18.5 (1974) 187–196.
[35] H. P. Benson, Fractional programming with convex quadratic forms and functions, European Journal of Operational Research 173.2 (2006) 351–369.
[36] R. Yamamoto, and H. Konno, An efficient algorithm for solving convexconvex quadratic fractional programs, Journal of Optimization Theory and Applications 133.2 (2007) 241–255.
[37] L. Shao, and M. Ehrgott, An objective space cut and bound algorithm for convex multiplicative programmes, Journal of Global Optimization 58.4 (2014) 711–728.
[38] M. Ehrgott, Multicriteria optimization, Springer Science & Business Media, 2005.
[39] A. Pascoletti, and P. Serafini, Scalarizing vector optimization problems, Journal of Optimization Theory and Applications 42.4 (1984) 499–524.
[40] G. Eichfelder, An adaptive scalarization method in multiobjective optimization, SIAM Journal on Optimization 19.4 (2009) 1694–1718.
[41] P. C. Chen, P. Hansen, and B. Jaumard, On-line and off-line vertex enumeration by adjacency lists, Operations Research Letters 10.7 (1991) 403–409.
[42] E. Khorram, K. Khaledian, and M. Khaledyan, A numerical method for constructing the Pareto front of multi-objective optimization problems, Journal of Computational and Applied Mathematics 261 (2014) 158–171.
[43] G. Eichfelder, Scalarizations for adaptively solving multi-objective optimization problems, Computational Optimization and Applications 44.2 (2009) 249–273.
[44] D. Mueller-Gritschneder, H. Graeb, and U. Schlichtmann, A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems, SIAM Journal on Optimization 20.2 (2009) 915–934.
[45] E. Zitzler, and L. Thiele, Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach, IEEE transactions on Evolutionary Computation 3.4 (1999) 257–271.
[46] S. Bandyopadhyay, S. K. Pal, and B. Aruna, Multiobjective GAs, quantitative indices, and pattern classification, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 34.5 (2004) 2088–2099.
[47] H. Y. Meng, X. H. Zhang, and S. Y. Liu, New quality measures for multiobjective programming, Lecture Notes in Computer Science 3611 (2005) 1044–1048.
[48] A. Messac, and C. A. Mattson, Normal constraint method with guarantee of even representation of complete Pareto frontier, AIAA journal 42.10 (2004) 2101–2111.
[49] K. Deb, A. Pratap, S. Agarwal, and T. A. M. T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Transactions on Evolutionary Computation 6.2 (2002) 182–197.
[50] S. Das, and P. N. Suganthan, Differential Evolution: A Survey of the State-of-the-Art, IEEE Transactions on Evolutionary Computation 15 (2011) 4–31.
[51] N. Beume, B. Naujoks, and M. Emmerich, SMS-EMOA: Multiobjective selection based on dominated hypervolume, European Journal of Operational Research 181.3 (2007) 1653–1669.
[52] A. Messac, A. Ismail-Yahaya, and C. A. Mattson, The normalized normal constraint method for generating the Pareto frontier, Structural and Multidisciplinary Optimization 25.2 (2003) 86–98.

[53] R. S. Burachik, C. Y. Kaya, and M. Rizvi, A new scalarization technique to approximate Pareto fronts of problems with disconnected feasible sets, Journal of Optimization Theory and Applications 162.2 (2014) 428–446.
[54] S. Nobakhtian, and N. Shafiei, A Benson type algorithm for nonconvex multiobjective programming problems, TOP: An Official Journal of the Spanish Society of Statistics and Operations Research 25.2 (2017) 271–287.
[55] A. Ghane-Kanafi, and E. Khorram, A new scalarization method for finding the efficient frontier in non-convex multi-objective problems, Applied Mathematical Modelling 39.23-24 (2015) 7483–7498.
[56] G. Leitmann, and A. Marzollo, Multicriteria Decision Making, Springer-Verlag, New York, 1975.
[57] M. Tanaka, H. Watanabe, Y. Furukawa, T. Tanino, GA-based decision support system for multicriteria optimization, IEEE International Conference on Systems, Man and Cybernetics, Intelligent Systems for the 21St Century 2 (1995) 1556–1561.
[58] T.T. Binh, and U. Korn, MOBES: a multiobjective evolution strategy for constrained optimization problems, The Third International Conference on Genetic Algorithms (1997) 176–182.