Iranian Journal of Operations Research
Vol. 10, No. 2, 2019, pp. 49-61

A New Algorithm for Constructing the Pareto Front of Bi-objective Optimization Problems

M. Ghaznavi[1,*], M. Abkhizi[2]

Here, scalarization techniques for multi-objective optimization problems are addressed. A new scalarization approach, called unified Pascoletti-Serafini approach, is utilized and a new algorithm to construct the Pareto front of a given bi-objective optimization problem is formulated. It is shown that we can restrict the parameters of the scalarized problem. The computed efficient points provide a nearly equidistant approximation of the whole Pareto front. The performance of the proposed algorithm is illustrated by various test problems and its effectiveness with respect to some existing methods is shown.

***Keywords**: Bi-objective optimization, Pareto front, Scalarization, Unified Pascoletti-Serafini method, Proper efficiency.*

Manuscript was received on 17/07/2019, revised on 01/10/2019 and accepted for publication on 16/12/2019.

1. Introduction

In many real-life applications, to make a decision it is necessary to maximize or minimize several conflicting objective functions. In general, it is not possible to attain a solution that optimizes all the conflicting functions, simultaneously. Therefore, one needs to trade off between different objectives to arrive at a compromise solution. Multi-objective programming is concerned with finding the best trade-off, in some sense, between the existing objectives. A solutions set is obtained, while a trade-off is made between the competing objective functions. Such a compromise solution is named a Pareto optimal (an efficient) solution, and the set containing the images of these efficient solutions in the objective space, is referred to as the Pareto front (efficient curve).

A widespread approach for solving a given multi-objective optimization problem (MOP) consists in reformulating the MOP as a scalarized one, i.e., a real-valued optimization problem, involving possibly a set of additional parameters and/or some constraints. The solution of the single-objective optimization problem can be obtained by standard optimization algorithms. Some scalarization methods applied in the literature are the weighted sum [8, 21], the modified weighted Tchebycheff [16], the (improved) ε-constrained [8], the Benson [8] and the Pascoletti-Serafini and its generalization [1, 9, 12, 24, 26]. Further scalarization methods are discussed in [2, 8, 9, 13, 22, 25, 26].

In general, the Pareto set is infinite. Theoretically, the Pareto front can be constructed by solving some scalarization problems. However, in order to construct a reasonable approximation of the

[1] Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran, Email: ghaznavi@shahroodut.ac.ir.
[2] Faculty of Mathematical Sciences, Shahrood University of Technology, Shahrood, Iran, Email: mahbobe.abkhizi@gmail.com.

efficient curve, it is mandatory to solve the utilized scalarization problem many times, which turns to be costly. Furthermore, the feasible set and/or the Pareto front may be non-convex or disconnected, which causes the computational difficulties. Therefore, in order to get rid of these problems, researchers focused on providing algorithms to find a discrete approximation of the Pareto front by generating a reasonable number of Pareto solutions. After finding a discrete approximation, the obtained Pareto front is given to the decision maker and she chooses the solution which best fits her preferences. Hence, it is essential that the proposed algorithm be capable to produce an even approximation of the whole Pareto front.

In recent years, researchers have developed methods for constructing even Pareto fronts. For example, Das and Dennis [4] proposed the normal boundary intersection (NBI) approach. Messac et al. [21] presented the normal constraint (NC) method which is more flexible as compared to the NBI method, by investigating a single-objective optimization problem with inequality constraints. Mueller-Gritschneder et al. [23] proposed the successive Pareto optimization (SPO) method. The weighted constraint method was used by Burachik et al. [2] and Rizvi [25] for generating the Pareto front of bi-objective optimization problems that may produce solutions that are not Pareto. Ghaznavi and Azizi [14] proposed an algorithm for generating the Pareto front of a convex MOP. The Pascoletti–Serafini (PS) scalarization approach was used by some authors for constructing the Pareto front. Eichfelder [9, 10] proposed an algorithm for solving nonlinear MOPs based on this approach. She adaptively controlled the scalarization parameters using sensitivity results. Thereafter, Khorram et al. [19] suggested a new algorithm based on the PS technique and provided new results to restrict the parameter set of this technique. More recently, Khaledian et al. [17, 18] and Dolatnezhadsomarin and Khorram [7] also utilized the PS method to obtain new algorithms for constructing the Pareto front.

However, some of these algorithms may generate non-Pareto optimal solutions. Some other methods are unable to cover the whole Pareto front, and some methods are not able to solve MOPs with disconnected or non-convex Pareto fronts. Furthermore, proper efficient solutions can not be characterized using the PS scalarization approach. In fact, by the algorithms based on the PS approach, it is possible to find solutions with trade-off between their criteria being unbounded. In order to resolve this drawback of the PS approach, more recently, Ghaznavi et al. [12] suggested a new scalarization approach called the unified PS technique. They showed that, under some conditions, the optimal solutions of this new scalarization problem are properly efficient for the given MOP.

Here, we are to utilize a unified Pascoletti-Serafini approach to provide a new algorithm for constructing a discrete approximation of the Pareto front of a given bi-objective optimization problem. We control the parameters of the scalarized problem via sensitivity results. The proposed algorithm can cover the whole Pareto front. Moreover, we provide some conditions on the algorithm to have a uniform distribution of the Pareto points. Unlike some other approaches, the proposed algorithm does not generate non-Pareto solutions. The proposed algorithm is not only applicable to bi-objective problems with disconnected and non-convex efficient curves, but also to problems with disconnected and non-convex feasible sets. We denote the effectiveness of the algorithm with various test problems, specially with ones having disconnected and non-convex Pareto fronts.

The remainder of our work is proceed as follows. In Section 2, the basic concepts in multi-objective programming are recalled and the unified Pascoletti-Serafini scalarization approach is reviewed. The proposed algorithm is explained in Section 3 and some theoretical results are provided to control the parameters. In Section 4, efficiency of the proposed algorithm is shown by some test problems. Finally, concluding remarks are presented in Section 5.

2. Preliminaries

Let $X \subseteq \mathbb{R}^n$ and $f: X \to \mathbb{R}^p$. A multi-objective optimization problem is given by

$$\begin{aligned}(MOP):\ & \min f(x) = (f_1(x), \dots, f_p(x)) \\ & \text{s.t.} \quad x \in X.\end{aligned} \tag{1}$$

If $p = 2$, then (1) is called a bi-objective optimization problem.

The image of X under f is denoted by $Y = f(X) = \{y \in \mathbb{R}^m: y = f(x), \text{for some } x \in X\}$. The natural ordering cone is defined as

$$\mathbb{R}^p_{\gg} = \{x \in \mathbb{R}^p: x_i \geq 0, \qquad i = 1, 2, \dots, p\}$$

For any $y, \bar{y} \in \mathbb{R}^p$, we define $y < \bar{y}$ if $y_i < \bar{y}_i$, for $i = 1,2, \dots, p$. Moreover, $y \ll \bar{y}$ if $y_i \leq \bar{y}_i$, for $i = 1, 2, \dots, p$, and $y \leq \bar{y}$ means $y_i \leq \bar{y}_i$, for $i = 1, 2, \dots, p$, and $y \neq \bar{y}$.

Definition 2.1. A feasible solution $\hat{x} \in X$ is said to be

- efficient (Pareto optimal) for (1), if there is no other $x \in X$ such that $f(x) \leq f(\hat{x})$.
- weakly efficient (weakly Pareto optimal) for (1), if there is no other $x \in X$ such that $f(x) < f(\hat{x})$.

The image of an efficient (a weakly efficient) solution in the decision space is called a non-dominated (a weakly non-dominated) point. We denote the sets of all efficient solutions and non-dominated points by X_E and Y_N, respectively. Also, the sets of all weakly efficient solutions and weakly non-dominated points will be denoted by X_{WE} and Y_{WN}, respectively.

Based on Definition 2.1, an efficient solution is a feasible solution for which it is not possible to improve one criterion without worsening the others. These trade-offs between the objectives can be characterized by calculating the increase in objectivef_i, say, per unit decrease in objectivef_j. However, sometimes, these trade-offs between criteria may be unbounded. Geoffrion [11] defined properly efficient points as efficient points with bounded trade-offs.

Definition 2.2. A feasible solution $\hat{x} \in X$ is a properly efficient (a properly Pareto optimal) solution to (1), if it is efficient and there is some real positive number M such that for each $i \in \{1,2, \dots, p\}$ and each $x \in X$ satisfying $f_i(x) < f_i(\hat{x})$, there exists at least one index $j \in \{1,2, \dots, p\}$ such that $f_j(\hat{x}) < f_j(x)$ with $\frac{f_i(\hat{x}) - f_i(x)}{f_j(x) - f_j(\hat{x})} \leq M$.
The point $\hat{y} = f(\hat{x})$ is then called a properly non-dominated point. The sets of all properly efficient solutions and properly non-dominated points will be denoted byX_{PE} and Y_{PN}, respectively.

Definition 2.3. [8] The point $y^I = (y_1^I, \dots, y_p^I) \in \mathbb{R}^p$ in which $y_i^I = \min_{x \in X} f_i(x),\ i = 1, \dots, p$, is said to be the ideal point of (1).

Definition 2.4. [8] The point $y^U \in \mathbb{R}^p$ in which $y_i^U = y_i^I - \alpha_i, i = 1, \dots, p$, for some $\alpha = (\alpha_1, \dots, \alpha_p) > 0$, is said to be a utopia point of (1).

Definition 2.5. [8] The non-dominated set Y_N is called externally stable, if for each $y \in Y \backslash Y_N$ there exists a non-dominated point $\hat{y} \in Y_N$ such that $y \in \hat{y} + \mathbb{R}^p_{\gg}$.

MOPs are often solved indirectly by using scalarization which requires formulating a single-objective optimization problem. In general, scalarization consists in the transformation of a MOP into a suitable real-valued scalar optimization problem, possibly containing some additional parameters and/or constraints. Since there are a wide variety of procedures for solving a given scalar optimization problem, scalarization is very important in vector optimization. Solutions of a MOP can be attained as optimal solutions of a suitable scalarization problem. A large variety of scalarization techniques are discussed in [8, 9].

Pascoletti-Serafini technique [24] is a well-known famous scalarization approach for solving a given MOP. However, the Pascoletti-Serafini scalarized problem has inflexible constraints. Moreover, it is not easy to check the conditions of proper efficiency (for more details, see [1]). Therefore, in order to resolve these drawbacks, Ghaznavi et al. [12] introduced the following modification of this scalarization method, called the unified Pascoletti-Serafini scalarization problem.

Assume that $a \in \square^p$ and $r \in \mathbb{R}^p_{\gg} \backslash \{0\}$. The unified Pascoletti-Serafini problem is given by

$$
\begin{aligned}
UPS(a, r, \lambda):\ & \min\ t \\
\text{s.t.}\ \ & a_i + tr_i - f_i(x) - \max_{j \in \{1,\dots,p\}} \{\lambda_j f_j(x)\} \geq 0, \quad i = 1, 2, \dots, p \\
& x \in X, \qquad t \in \mathbb{R},
\end{aligned}
$$

where $\lambda_i \geq 0,\ i = 1,2, \dots, p$. The single-objective problem UPS(a, r, λ) can be applied to prove necessary and sufficient conditions for different types of efficient solutions of (1). In what follows, we review some theorems obtained via the proposed scalarized problem. For more details, see [12].

Theorem 2.6. [12] If $(\hat{t}, \hat{x})$ is an optimal solution of the scalarized problem UPS(a, r, λ) with $\lambda \gg 0$, then $\hat{x}$ is a weakly efficient solution of (1).

Theorem 2.7. [12] If $(\hat{t}, \hat{x})$ is an optimal solution of the scalarized problem UPS(a, r, λ) with $\lambda \gg 0$ and $a_i + \hat{t} r_i - f_i(\hat{x}) - \max_{j \in \{1,\dots,p\}} \{\lambda_j f_j(\hat{x})\} > 0$, for $i = 1, 2, \dots, p$, then $\hat{x} \in X_E$.

Theorem 2.8. [12] If $(\hat{t}, \hat{x})$ is an optimal solution of the scalarized problem UPS(a, r, λ) with $\lambda > 0$ and $a_i + \hat{t} r_i - f_i(\hat{x}) - \max_{j \in \{1,\dots,p\}} \{\lambda_j f_j(\hat{x})\} > 0$ for $i = 1, 2, \dots, p$, then $\hat{x} \in X_{PE}$.

Theorem 2.9. [12] If Y_N is externally stable and $\hat{x} \in X$ is an efficient solution of (1), then, there exist a weight vector $\lambda \gg 0$ and parameters $a \in \square^p$, $r \in \mathbb{R}^p_{\gg} \backslash \{0\}$ such that $(\hat{t}, \hat{x})$ with $\hat{t} = 0$ is an optimal solution of UPS(a, r, λ).

3. Parameter Set Restriction and Proposed Algorithm

Our goal is to construct an approximation of the efficient set of (1) by solving the scalarization problem UPS(a, r, λ), for different parameters. Ghaznavi et al. [12] proved that all efficient solutions can be obtained by varying the constant parameters $a \in \mathbb{R}^p$, $r \in \mathbb{R}^p_{\gg} \backslash \{0\}$ and $\lambda \gg 0$ (see Theorem

2.9). In the following theorem, we show that it is not necessary to consider all the parameters $a \in \mathbb{R}^p$and $r \in \mathbb{R}^p_{\gg}\backslash\{0\}$. We can restrict the sets from which we have to choose the parameters $a \in \mathbb{R}^p$and $r \in \mathbb{R}^p_{\gg}\backslash\{0\}$ so that we still find all efficient points of the MOP. We show that by considering $a = \max_{j\in\{1,..,p\}}\{\lambda_j f_j(\hat{x})\}$ and varying $r \in K = \{\beta \in \mathbb{R}^p |\ \|\beta\|_2 = 1\}$, all efficient solutions of (1) can be attained.

Theorem 3.1. Let $K = \{\beta \in \mathbb{R}^p |\ \|\beta\|_2 = 1\}$ and Y_N be externally stable. If $\hat{x} \in X$ is an efficient solution of (1), then there exists constant parameters $r \in K$, and $\lambda \geq 0$ and variable $\hat{t} \in \mathbb{R}$, such that $(\hat{t}, \hat{x})$ is an optimal solution of UPS(a, r, λ) with $a = \max_{j\in\{1,..,p\}}\{\lambda_j f_j(\hat{x})\}$.

Proof. Because Y_N is externally stable, for all $x \in X\backslash X_E$ there exists $\bar{x} \in X_E$ such that$f\ (\bar{x}) \leq f(x)$. Therefore,

$$\max_{j\in\{1,..,p\}}\{\lambda_j f_j(\bar{x})\} \leq \max_{j\in\{1,..,p\}}\{\lambda_j f_j(x)\}, \quad \forall\, \lambda \geq 0. \tag{2}$$

Define

$$\lambda_j = \begin{cases} 0\,, & f_j\,(\bar{x}) < f_j\,(\hat{x}) \\ \alpha > 0\,, & f_j\,(\bar{x}) \geq f_j\,(\hat{x}). \end{cases} \tag{3}$$

Since$f(X)\ \subseteq \mathbb{R}^p_{\geq}\backslash\{0\}$, we conclude $f(\hat{x}) \neq 0$. Also, it is obvious that $\frac{f(\hat{x})}{\|f(\hat{x})\|_2} \in K$. We set $a_i = \max\ \{\lambda_j f_j(\hat{x})\}$, $r = \frac{f(\hat{x})}{\|f(\hat{x})\|_2}$ and $\hat{t} = \|f(\hat{x})\|_2$. We have $a_i + \hat{t}r_i - f_i\,(\hat{x}) - \max_{j\in\{1,..,p\}}\{\lambda_j f_j(\hat{x})\} = 0, \forall i$. This implies that $(\hat{t}, \hat{x})$ is a feasible solution for UPS(a, r, λ). Now, we show that $(\hat{t}, \hat{x})$ is an optimal solution of UPS(a, r, λ). By contradiction, assume that there exists a solution (t, x) for UPS(a, r, λ) which is feasible and $t < \hat{t}$. Since (t, x) is feasible, we have

$$a_i + tr_i - f_i\,(x) - \max_{j\in\{1,..,p\}}\{\lambda_j f_j(x)\} \geq 0, \forall i.$$

Therefore,

$$\max_{j\in\{1,...,p\}}\{\lambda_j f_j(\hat{x})\} + t\frac{f_i(\hat{x})}{\|f(\hat{x})\|_2}\ -\ f_i(x)\ -\ \max_{j\in\{1,..,p\}}\{\lambda_j f_j(x)\} \geq 0, \qquad \forall i. \tag{4}$$

Since $t < \hat{t}$ and $\hat{t} = \|f(\hat{x})\|_2$, it concludes from (2) and (4) that

$$f_i\,(\,\hat{x}) - f_i\,(\,x) - \max_{j\in\{1,..,p\}}\{\lambda_j f_j(\bar{x}) - \lambda_j f_j(\hat{x})\} \geq 0, \qquad \forall\, i \in \{1, \dots, p\},$$

and

$$f_r(\,\hat{x}) - f_r\,(\,x) - \max_{j\in\{1,..,p\}}\{\lambda_j f_j(\bar{x}) - \lambda_j f_j(\hat{x})\} > 0, \qquad \text{for some } r. \tag{5}$$

Hence, from relations (3) and (5) we have $f_i\,(x) \leq f_i\,(\hat{x})$, $\forall\, i \in \{1, \dots, p\}$ and $f_r(x) < f_r(\hat{x})$, for some $r \in \{1,2,\dots,p\}$, which is a contradiction to efficiency of $\hat{x}$ for (1).

Now, we present Algorithm 1, utilizing the proposed scalarization technique, for obtaining an approximation of the Pareto front for bi-objective optimization problems.

Algorithm 1 (for bi-objective optimization problems).
Step 1: (Input) choose the desired number of points N.
Step 2: (Obtain the end points of the efficient curve)

2-1 Find the optimal solution $\bar{x}_0$ of $\begin{cases}\min f_1(x)\\ \text{s.t. } x \in X\end{cases}$

Let $a_i = f_i(\bar{x}_0)$, for $i = 1, 2$.

2-2 Find x_0^* that solves $\begin{cases}\min f_2(x)\\ \text{s.t. } x \in X\end{cases}$

$b_i = f_i(x_0^*)$, for $i = 1,\ 2$.

2-3 Set $l_i = \frac{|b_i - a_i|}{N}$, for $i = 1,\ 2$ and set $d = l_1^2 + l_2^2$.

2-4 Set $k = 1$. Let $Y(k) = [f_1(x_0^*), f_2(x_0^*)]$. Then, $Y(k)$ is a weakly non-dominated point.

2-5 Set $i = 1,\ s = 0,\ s1 = 0,\ s2 = 0$.

Step 3: (Select $\hat{x}$ to construct the parameters $a$and r and solve the scalarized problem)

Project the point $f(x_{k-1}^*) \in Y$ on the $f_1 -$axis, and then move l_1 units to the left. The point $\left(f_1(x_{k-1}^*) - l_1\ ,\ 0\right)$ is obtained. From this point, move in apparel with thef_2 $-$axis and find $\hat{x}$ that solves $\begin{cases}\min f_2(x)\\ \text{s.t. } f_1(x) = f_1(x_{k-1}^*) - l_1 * (i + s2)\end{cases}$.

Step 4: (solve the UPS(a, r, λ))

4-1 Select $\lambda_i \epsilon \left[0, \frac{1}{100}\right]$, for $i = 1, 2$ and set $a_i = \max\left(\lambda_1 f_1(\hat{x}), \lambda_2 f_2(\hat{x})\right)$, for $i = 1, 2$ and $r_i = \frac{f_i(\hat{x})}{\|f(\hat{x})\|_2}$, for $i = 1,2$.

4-2 Formulate the UPS(a, r, λ) problem using a and r obtained in Substep 4-1, and then solve it to find the optimal solution $(x_k^*, \hat{t})$ and set $X(s1) = [f_1(x_k^*), f_2(x_k^*)]$. We know that $X(0) \neq [f_1(x_{k-1}^*), f_2(x_{k-1}^*)]$.

Step 5: (Perform the following substeps for uniformity of the Pareto set)

5-1 Compute $d' = \|\mathrm{Y}(k) - f(x_k^*)\|_2^2$.

5-2 If $\left(d' \leq d \text{ and } f_1(x_{k-1}^*) \geq f_1(x_k^*) \text{ and } f_2(x_{k-1}^*) \leq f_2(x_k^*) \text{ and } f(x_{k-1}^*) \neq f(x_k^*)\right)$, then accept $f(x_k^*)$ as a new non-dominated solution and update Y, that is, $Y(k+1) = [f_1(x_k^*), f_2(x_k^*)]$. Set $k = k + 1$ and go to Step 7.

5-3 Else if $\left(d' \leq d \text{ and } f_1(x_{k-1}^*) \leq f_1(x_k^*) \text{ and } f_2(x_{k-1}^*) \geq f_2(x_k^*)\right)$, then we set $Y(k+1) = X(s1 - 1)$ and $s1 = s1 + 1$ and $k = k + 1$ and go to Step 7.

5-4 Else if $d' > d$, then

5-4-1 Find $\bar{x}$ that solves $\begin{cases}\min f_2(x)\\ \text{s.t. } f_1(x) \leq f_1(\hat{x})\\ \quad f_2(x) \leq f_2(\hat{x})\\ \quad x \in X\end{cases}$

5-4-2 If $\left(f_1(\bar{x}) < f_1(\hat{x}) \text{ and } f_2(\bar{x}) < f_2(\hat{x})\right)$, then accept $f(x_k^*)$ as a new non-dominated point and update Y, that is, $Y(k+1) = [f_1(x_k^*), f_2(x_k^*)]$. Set $k = k + 1$ and go to Step 7.

5-4-3 Else obtain $d'' = f_1(x_{k-1}^*) - l_1 * (i + s2 - 0.05)$ and do the following:

5-4-3-1 If $d'' < f_1(x_{k-1}^*)$ then set $s2 = s2 - 0.05$ and go to Step 3.

5-4-3-2 Else set $Y = [f_1(x_k^*), f_2(x_k^*)]$, and set $k = k + 1$ and go to Step 7.

Step 6: (Determine efficient and weak efficient points)

If $a_i + \hat{t}r_I$- $f_i(x_k^*) - \max_{j \in \{1,..p\}} \{\lambda_j f_j(x_k^*)\} > 0$, for all $i = 1,2, \ldots, p$, then x_k^* is efficient, else it is weakly efficient.

Step 7: (Condition of termination algorithm)

If $\left\| Y(k) - a \right\|_2^2 \leq d$, then set $Y(k+1) = [f_1(\overline{x_0}), f_2(\overline{x_0})]$ and stop, Y is an approximation of the Pareto front, else go to Substep 2-5.

It is important to note that in Substeps 2-1 and 2-2 of Algorithm 1, the individual objective functions subject to the constraints of (1) are minimized. The "outer" end points of the efficient curve are obtained by these minimal solutions. It is obvious that optimal solutions of these scalar optimization problems are at least weakly efficient points for (1).

In order to construct approximate solutions on the Pareto front uniformly, in Substep 2-3 we obtain two distances l_1 and l_2 and using these distances we calculate the value of d which is the maximum distance considered in the image space between two approximate solutions. In Substep 2-4 we define the set Y, which is the set of approximate points of the Pareto front and is revised during the algorithm. In Step 3, we solve an additional single-objective optimization problem to obtain a feasible solution $\hat{x}$. We note that if the feasible set in this single-objective optimization problem is non-empty, then $f(\hat{x})$ locates in the front. If the Pareto front is connected, then the feasible set of the single-objective optimization problem is non-empty.

In Substep 4-2, by solving the scalarized problem UPS(a, r, λ), we find the optimal solution $(x_k^*, \hat{t})$ and set $X = \{(f_1(x_k^*), f_2(x_k^*))\}$. We note that based on Theorem 2.6, x_k^* is a weakly efficient solution for (1). Hence, X is the set of all weakly efficient points that are obtained in an implementation of the algorithm. If, at the end of a run, X contains a single point, then we accept the same point as a new non-dominated point.

In the literature, many qualitative criteria have been suggested to evaluate the quality of approximations of the Pareto front (see [3, 4, 5, 9], for example). For the scalarization approaches, three of the most interesting metrics are the coverage error, cardinality and uniformity that were discussed by Sayin [28]. An approximation possesses a high quality if the generated approximate points are almost equidistant. Therefore, we perform Step 5 to find an almost equidistant approximation and generate the approximation of the Pareto front uniformly. In Substep 5-1, we obtain the distance between $f(x_k^*)$ and the previous optimal point obtained by the algorithm (i.e, $f(x_{k-1}^*)$). If this distance is less than or equal to the specified distance d (Substep 5-2), then we accept $f(x_k^*)$ as a new point on the Pareto front. If this distance is less than or equal to the specified distance d (Substep 5-3), that is, the point $f(x_k^*)$ places on $f(x_{k-1}^*)$ or lies after or below it, then we do not accept the point $f(x_k^*)$ and select a point from the set $X \backslash \{f(x_k^*)\}$, which has the smallest distance to $f(x_{k-1}^*)$, as a new point on the Pareto front. Consider this distance to be bigger than the specified distance d (Substep 5-4). If $d' > d$, then we solve the problem in Substep 5-4-1. In Substep 5-4-2, we accept $f(x_k^*)$ as a new point on the Pareto front. Otherwise, transfer to Substep 5-4-3.

Therefore, to obtain a weak efficient point with a maximum distance from the point $f(x_{k-1}^*)$, transfer is made to Substep 5-4-3. In this substep, we reduce the distance l_1 to get a new point between $f(x_k^*)$ and the previous optimal point obtained by the algorithm (i.e., $f(x_{k-1}^*)$). If by decreasing l_1, the new point is placed before the previous optimal point $f(x_{k-1}^*)$, then Substep 5-4-3-1 is performed. In this Substep, by decreasing $s2$, we decrease the value of l_1 to obtain another new point with distance being less than d'. Otherwise, Substep 5-4-3-2 is performed. In Step 6, we utilize the theorems provided in Section 2 to determine (properly) efficient and weak efficient points. In Step 7, if distance of the optimal point, obtained by the algorithm, from the end point of the front is less than the specified distance d, then the algorithm terminates. It is obvious that this algorithm is finite.

4. Numerical Results

In order to test the numerical algorithm developed in Section 3, in this section some test problems with specific difficulties such as non-convexity and disconnectivity of the feasible region and/or Pareto front are solved. By these test problems it can be seen that the proposed algorithm can generate approximation points that cover all regions of the Pareto front and keep an almost even distribution of the Pareto points. The results of our algorithm are compared with the numerical results due to Algorithm 7 of [26] using the Pascoletti–Serafini scalarization technique.

The algorithm was coded in MATLAB (R2015a) and all test problems were implemented on a laptop with a core i5 processor with 4 GB RAM and 2.5 GHz running Windows 7 Ultimate system. Moreover, the SQP optimization algorithm has been used in the MATLAB's optimization solver fmincon to solve the nonlinear single-objective optimization problems.

Test problem 1. Consider the following convex bi-objective problem of [10] as considered in [26]:

$$\begin{cases} \min & \left(\sqrt{1+{x_1}^2}, {x_1}^2 - 4x_1 + x_2 + 5\right) \\ \text{s.t.} & {x_1}^2 - 4x_1 + x_2 + 5 \geq 3.5 \\ & x_1 \geq 0\,, x_2 \geq 0. \end{cases}$$

The Pareto front of this test problem is convex and connected. We ran Algorithm 7 of [26], which uses the Pascoletti-Serafini scalarization technique, with $N = 15$ and $a = (-1, -10)$ and uniformly distributed weights. Also, we ran our proposed algorithm with $N = 15$ and $\lambda_i = \frac{1}{500}$, for $i = 1,2$. Fig. 1 shows the obtained non-dominated points with the algorithm using the Pascoletti-Serafini method. This algorithm generates 16 points. In Fig. 2, the generated Pareto front utilizing our proposed algorithm is shown. As seen in this figure, Algorithm 1 produces 21 efficient points, a larger number of points than those obtained by Algorithm 7 of [26]. By comparing the results of Figs. 1 and 2 it is clear that a uniform distribution of the approximation is generated by Algorithm 1. Moreover, by the same number of iterations ($N = 15$), the number of approximation points obtained via the proposed algorithm (21 points) is greater than that of Algorithm 7 of [26] (16 pints).

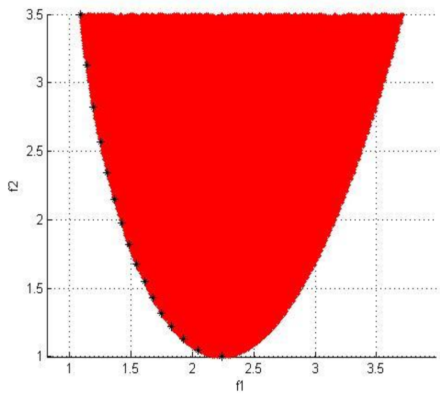

Figure 1. Points found by Algorithm 7 of [26].

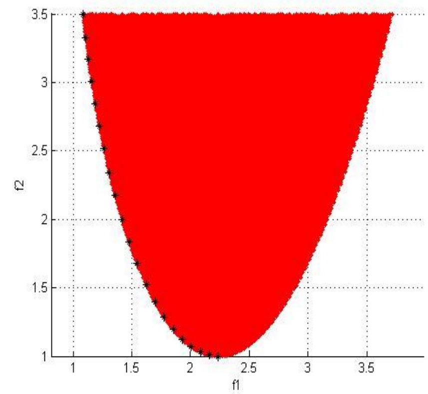

Figure 2. Points found by Algorithm 1.

Test problem 2. Consider the following test problem which has a non-convex image, and additionally its non-dominated set is disconnected and non-convex:

$$\begin{cases} \min \ (x_1, x_2) \\ \text{s.t.} \ -{x_1}^2 - {x_2}^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x_1}{x_2}\right)\right) \leq 0 \\ (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0 \\ x_1,\ x_2 \in [0, \pi]. \end{cases}$$

For Algorithm 7 of [26], we took $a = (-1, -10), N = 22$ and uniformly distributed weights and then ran the algorithm. Also, we ran Algorithm 1 with $N = 22$ and $\lambda_i = \frac{1}{500}$, for $i = 1, 2$. In an implementation with $N = 22$, Algorithm 7 of [26] and Algorithm 1 generated the points depicted in Figs. 3 and 4, respectively. Algorithm 7 of [26] found 20 points, two of which were not non-dominated. Furthermore, this algorithm could not construct the Pareto front uniformly. With Algorithm 1, 23 non-dominated points were found and the points were uniformly distributed in the front. Moreover, our Algorithm 1 produced all the end points of the Pareto front, which demonstrates the capability of the suggested technique for solving problems with disconnected and non-convex Pareto fronts.

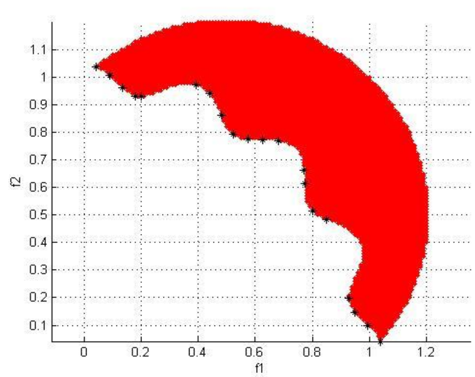

Figure 3. Points found by Algorithm 7 of [26].

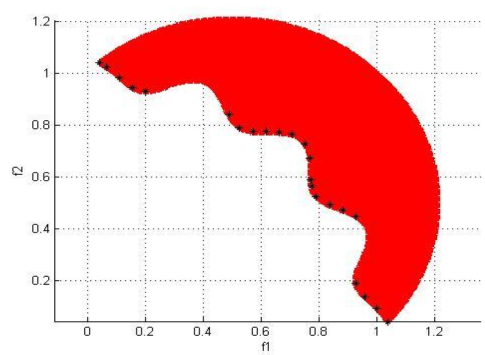

Figure 4. Points found by Algorithm 1.

Test problem 3. For the third test problem, we investigate the following non-convex bi-objective test problem from [26], that not only the efficient curve but also the feasible set itself is non-convex and disconnected.

$$\begin{cases} \min \ (x_1, x_2) \\ \text{s.t.} \ -{x_1}^2 - {x_2}^2 + 1 + 0.1\cos\left(16\arctan\left(\frac{x_1}{x_2}\right)\right) \leq 0 \\ \quad 1.69{x_1}^2 + 1.01{x_2}^2 - 2.60x_1x_2 - 0.02 \geq 0 \\ \quad (x_1 - 0.5)^2 + (x_2 - 0.5)^2 - 0.5 \leq 0 \\ \quad x_1,\ x_2 \in [0, \pi]. \end{cases}$$

For Algorithm 7 of [26], we took $a = (-1, -10)$, $N = 22$ and uniformly distributed weights. Then, we implemented this algorithm. Also we set $N = 22$ and $\lambda_i = \frac{1}{500}$, for $i = 1,2$ in Algorithm 1 and ran this algorithm. Figs. 5 and 6 show the generated points by algorithm 7 of [26] and Algorithm 1, respectively. As it can be seen in Fig. 5 with $N = 22$, Algorithm 7 of [26] produced 17 points that two of them were not non-dominated. Furthermore, the non-dominated points found by this algorithm were not uniformly distributed. Hence, this algorithm approximated the Pareto front poorly. However, with our proposed algorithm, 21 non-dominated points were found and the points were uniformly distributed. Moreover, this algorithm generated all the end-points of the Pareto front, which demonstrates the capability of the proposed algorithm for solving problems with disconnected and non-convex feasible sets.

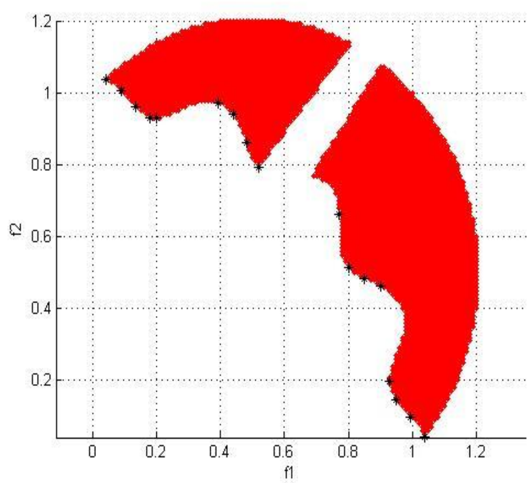

Figure 5. Points found by Algorithm 7 of [26].

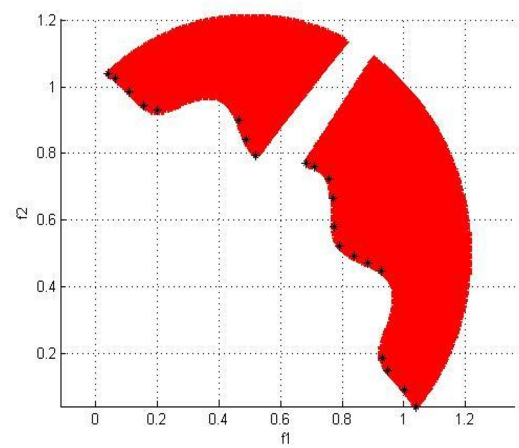

Figure 6. Points found by Algorithm 1.

Test problem 4. The fourth test problem, from [27], is

$$\begin{aligned} &\min \ f_1(x) = x_1^4 + x_2^4 - x_1^2 + x_2^2 - 10x_1x_2 + 0.25x_1 + 20 \\ &\min \ f_2(x) = (x_1 - 1)^2 + x_2^2 \\ &\text{s.t.} \ x_1, x_2 \in [-3,3]. \end{aligned}$$

The Pareto front of this test problem is disconnected and non-convex. In Algorithm 7 of [26], we chose the point $a = (-1, -10)$ to be the utopia point. With $N = 50$ and uniformly distributed weights we ran the algorithm. We also ran Algorithm 1 for $N = 50$ and $\lambda_i = \frac{1}{500}$, for $i = 1,2$. In an implementation with $N = 50$, Algorithm 7 of [26] and Algorithm 1 found the points depicted in Figs. 7 and 8, respectively. As seen in Fig. 7, Algorithm 7 of [26] produced only 36 efficient points from the lower part of the Pareto front. However, our proposed algorithm produced equidistant points over the whole Pareto front. The number of points computed by our algorithm is 55. A comparison of Figs. 7 and 8 verifies that the distribution of points generated by our proposed algorithm is better than that of [26], so that the points generated by Algorithm 1 can cover the whole Pareto front by having a good spread.

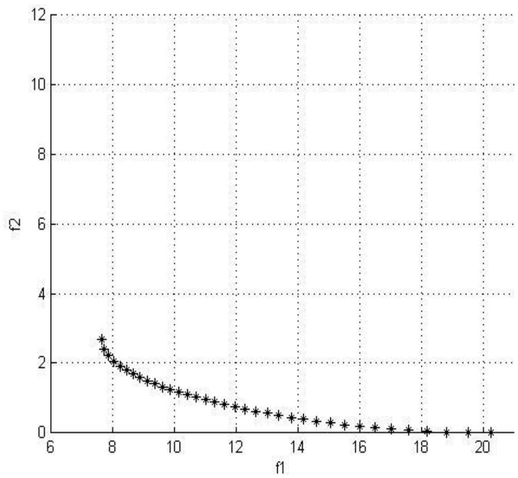

Figure 7. Points found by Algorithm 7 of [26].

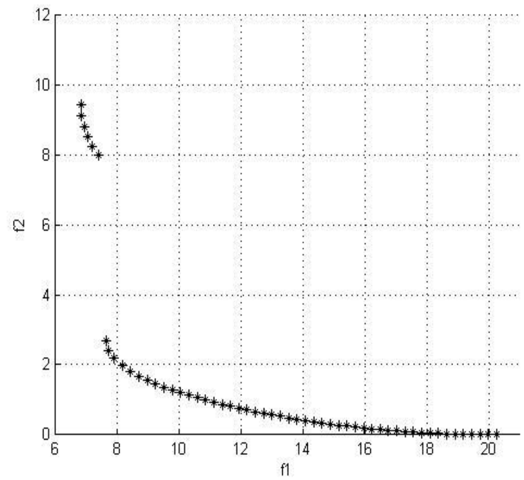

Figure 8. Points found by Algorithm 1.

5. Conclusions

A numerical algorithm was proposed for generating an approximation of the Pareto front for a bi-objective optimization problem. This algorithm was based on a new scalarization method called the unified Pascoletti-Serafini method. The proposed approach can be utilized for solving any bi-objective problem with connected, disconnected, convex and non-convex feasible and efficient sets, and does not require any extra condition. In the proposed algorithm, a parameter restriction was utilized and by a cutting procedure an almost even approximations of the Pareto front were obtained. The effectiveness of the algorithm was shown by different test problems with non-convex or disconnected efficient curve and it was seen that the distribution of the points computed by the proposed algorithm was even and the whole Pareto front was covered by the points generated by the proposed algorithm having a good spread.

Extending the proposed algorithm for a multi-objective optimization problem with more than two objective functions can be a worthwhile research area for a future work.

References

[1] Akbari, F., Ghaznavi, M., and Khorram, E. (2018), A revised pascoletti–serafini scalarization method for multiobjective optimization problems, *Journal of Optimization Theory and Applications*, 178(2), 560-590.

[2] Burachik, R.S., Kaya, C.Y. and Rizvi, M.M. (2014), A new scalarization technique to approximate Pareto fronts of problems with disconnected feasible sets, *Journal of Optimization Theory and Applications*, 162(2), 428-446.

[3] Das, I. and Dennis, J.E. (1998), Normal-boundary intersection: A new method for generating the Pareto surface in nonlinear multicriteria optimization problems, *SIAM journal on optimization*, 8(3), 631-657.

[4] Das, I. and Dennis, J.E. (1999), An improved technique for choosing parameters for Pareto surface generation using normal-boundary intersection, In *Short paper proceedings of the third world congress of structural and multidisciplinary optimization* (Vol. 2, pp. 411-413),

[5] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T.A.M.T. (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6(2), 182-197.

[6] Deb, K., Thiele, L., Laumanns, M., and Zitzler, E. (2005), Scalable test problems for evolutionary multiobjective optimization, In *Evolutionary multiobjective optimization* (pp. 105-145), Springer, London.

[7] Dolatnezhadsomarin, A., and Khorram, E. (2019), Two efficient algorithms for constructing almost even approximations of the Pareto front in multi-objective optimization problems, *Engineering Optimization*, 51(4), 567-589.

[8] Ehrgott, M. (2005), Multicriteria Optimization, Springer, Berlin.

[9] Eichfelder, G. (2008), Adaptive Scalarization Methods in Multiobjective Optimization, Springer-Verlag, Berlin, Heidelberg.

[10] Eichfelder, G. (2009), An adaptive scalarization method in multiobjective optimization, *SIAM Journal on Optimization*, 19(4), 1694-1718.

[11] Geoffrion, A.M. (1968), Proper efficiency and the theory of vector maximization, *Journal of mathematical analysis and applications*, 22(3), 618-630.

[12] Ghaznavi, M., Akbari, F., and Khorram, E. (2019), Optimality conditions via a unified direction approach for (approximate) efficiency in multiobjective optimization, *Optimization Methods and Software*, In press.

[13] Ghaznavi, M. (2017), Optimality conditions via scalarization for approximate quasi efficiency in multiobjective optimization, *Filomat*, 31(3), 671-680.

[14] Ghaznavi, M., & Azizi, Z. (2017), An algorithm for approximating nondominated points of convex multiobjective optimization problems, *Bulletin of the Iranian Mathematical Society*, 43(5), 1399-1415.

[15] Ghaznavi-Ghosoni, B.A., Khorram, E. and Soleimani-Damaneh, M. (2013), Scalarization for characterization of approximate strong/weak/proper efficiency in multi-objective optimization. *Optimization*, 62(6), 703-720.

[16] Kaliszewski, I. (1987), A modified weighted Tchebycheff metric for multiple objective programming, *Computers & operations research*, 14(4), 315-323.

[17] Khaledian, K. (2016), Restricting the parameter set of the Pascoletti-Serafini scalarization, *Bulletin of the Iranian Mathematical Society*, 42(7 (Special Issue)), 89-112.

[18] Khaledian, K., and Soleimani-Damaneh, M. (2015), A new approach to approximate the bounded Pareto front, *Mathematical Methods of Operations Research*, 82(2), 211-228.

[19] Khorram, E., Khaledian, K. and Khaledyan, M. (2014), A numerical method for constructing the Pareto front of multi-objective optimization problems, *Journal of Computational and Applied Mathematics*, 261, 158-171.

[20] Messac, A., Ismail-Yahaya, A. and Mattson, C. A. (2003), The normalized normal constraint method for generating the Pareto frontier, *Structural and multidisciplinary optimization*, 25(2), 86-98.

[21] Messac, A., and Mattson, C. A. (2004), Normal constraint method with guarantee of even representation of complete Pareto frontier, *AIAA journal*, 42(10), 2101-2111.

[22] Miettinen, K. (1999), Nonlinear Multi-Objective Optimization, Kluwer Academic, Dordrecht.

[23] Mueller-Gritschneder, D., Graeb, H. and Schlichtmann, U. (2009), A successive approach to compute the bounded Pareto front of practical multiobjective optimization problems, *SIAM Journal on Optimization*, 20(2), 915-934.

[24] Pascoletti, A. and Serafini, P. (1984), Scalarizing vector optimization problems, *Journal of Optimization Theory and Applications*, 42(4), 499-524.

[25] Rastegar, N. and Khorram, E. (2014), A combined scalarizing method for multiobjective programming problems, *European Journal of Operational Research*, 236(1), 229-237.

[26] Rizvi, M.M. (2013), New optimality conditions for non-linear multiobjective optimization problems and new scalarization techniques for constructing pathological Pareto fronts, Ph.D. Thesis, University of South Australia.

[27] Preuss, M., Naujoks, B. and Rudolph, G. (2006), Pareto set and EMOA behavior for simple multimodal multiobjective functions, In *Parallel Problem Solving from Nature-PPSN IX* (pp. 513-522), Springer, Berlin, Heidelberg.

[28] Sayın, S. (2000), Measuring the quality of discrete representations of efficient sets in multiple objective mathematical programming, *Mathematical Programming*, 87(3), 543-560.