J. Nonlinear Var. Anal. 10 (2026), No. 4, pp. 845-866
Available online at http://jnva.biemdas.com
https://doi.org/10.23952/jnva.10.2026.4.08

COMPUTING THE SET OF OPTIMAL POINTS FOR NONCONVEX MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

REFAIL KASIMBEYLI[1,2,*], K. GULNAZ BULBUL[3], BEHNAM SOLEIMANI[4], GURKAN OZTURK[1]

[1]*Department of Industrial Engineering, Faculty of Engineering, Eskisehir Technical University, Eskisehir, Turkiye*
[2]*UNEC Mathematical Modeling and Optimization Research Center, Azerbaijan State University of Economics, Baku, Azerbaijan*
[3]*Department of Geography and Environmental Management, University of Waterloo, Waterloo, Canada*
[4]*Institute of Mathematics, Martin-Luther-University Halle-Wittenberg, Halle, Germany*

Abstract. This paper proposes a new method for computing a set of optimal points, namely, minimal, properly minimal, and weakly minimal points, and approximate optimal points for nonconvex multi-objective problems, based on a given set of weights and reference points. This is an iterative method that utilizes the scalarizing functions of both the Conic Scalarization and the Pascoletti-Serafini methods at each iteration. The convergence of the proposed method is proven, and it is demonstrated that the method terminates in a finite number of iterations for a specified error accuracy. The performance of the method is illustrated through numerical examples.

Keywords. Conic scalarization; Multi-objective programming; Nonconvex optimization; Pascoletti-Serafini method; Scalarization.

1. INTRODUCTION

This paper presents a method for computing a subset of optimal points and approximate optimal points for nonconvex multi-objective optimization problems, corresponding to the given reference or ideal points and the given set of weights for objective functions. Note that the problem of generating the set of optimal points for multi-objective optimization problems is not new. It is a very interesting and difficult problem which has been studied in the literature earlier for various multi-objective optimization problems under different conditions. An alternative way is to approximate the set of optimal points instead of finding the whole optimal set. These problems can be classified as linear and nonlinear, and convex and nonconvex, generally. The methods for linear problems can be found in [2, 4, 23]. Most of the methods for nonlinear problems use inner or outer approximation of the efficient or weakly efficient set (see, e.g., [5, 6, 8, 9, 17, 13, 24, 26, 29, 31]). Approximation methods for nonconvex multi-objective problems are presented in [13] and [24], where a scalarizing function called the polyhedral gauge function is used. Note that scalarization methods for nonconvex multi-objective problems use

---

[*] Corresponding author.

E-mail address: rkasimbeyli@eskisehir.edu.tr (R. Kasimbeyli), gbulbul@uwaterloo.ca (K.G.Bulbul), behnam.soleimani@gmail.com (B. Soleimani), gurkan.o@eskisehir.edu.tr (G. Ozturk)

Received 25 June 2025; Accepted 16 April 2026; Published online 5 May 2026.

this or similar scalarizing functions (see, e.g., [28, 30]). The analysis of many methods shows that one of the main factors that affects the quality of approximation methods or the methods for generating the (weakly) minimal set is a scalarizing function used for this purpose. The quality of the set of optimal points generated may depend on conditions such as whether the given scalarizing function requires convexity of the problem under consideration, boundedness of the objective space, differentiability; whether the set of (scalarizing) parameters of this function involves some preferences of the decision-maker, such as weights and reference points; whether this function is able to generate minimal and/or properly minimal points or only weakly minimal points are guaranteed when using this scalarizing function and so on. It is clear that different scalarizing functions may provide different opportunities in this respect. Among these conditions, the convexity of the problem is very important, because for nonconvex problems there may exist the so-called "unsupported" (or "hidden") minimal and properly minimal points corresponding to the same weight vector or reference point. Therefore, the scalarizing function, which is used to compute such a kind of optimal points and which takes into account the weight and reference point preferences of decision-makers, is a very important issue.

Recent advancements have significantly enhanced the field of multi-objective optimization. For example, an improved proximal method was introduced with quasi-distances, which addresses nonconvex multi-objective optimization problems under locally Lipschitz conditions, providing new convergence results and handling constraints more effectively [1]. Globally convergent Newton-type methods were developed for nonconvex problems by modifying Hessians and incorporating safeguard strategies [14]. A dynamic proximal bundle algorithm was introduced that improves performance for nonsmooth nonconvex problems without needing scalarization [16]. In mixed-integer optimization, new branch-and-bound frameworks incorporate preference information and advanced cuts to address complex settings [7]. Furthermore, a reference-point-based branch-and-bound algorithm was developed to guide the search towards preferred regions on the Pareto front, incorporating decision-maker preference information and ensuring ε-efficient solutions [33]. Adaptive piecewise linear relaxations were used to handle nonconvex mixed-integer problems without convexity requirements [25]. Additionally, a non-monotone projected gradient method was also proposed for nonconvex multi-objective optimization, establishing convergence to Pareto stationary points under mild assumptions and showing effectiveness through numerical experiments [34].

As the field of multi-objective optimization continues to evolve with these promising developments, it becomes increasingly crucial to address the challenges and limitations that persist. In this paper, we aim to compute a subset of minimal, properly minimal, weakly minimal, and approximate optimal points, and we use two kinds of scalarizing functions for this purpose. One of them is the scalarizing function of the conic scalarization method suggested in [20], which does not require any kind of convexity or boundedness conditions on the problem and guarantees to generate all properly minimal points corresponding to the given reference point and the given set of weight parameters for objective functions (see also [11, 19, 21, 22]). In the method presented in this paper, besides the conic scalarizing function, the Pascoletti-Serafini scalarizing function from [28, 30] is also used. The convergence of the method is proven, and it is shown that the method terminates after a finite number of iterations for the given accuracy level. This is an iterative method that generates a new set of reference points at every iteration and for each new reference point, both the scalarizing functions are used to generate a "new" set of optimal points. These points are used to update the set of reference points and so on. We prove that the distance between the reference points and the objective space strictly decreases at every iteration and goes to zero. The method terminates if this distance becomes less than a given error accuracy. Note that the set of generated optimal and approximately optimal points is as large as the given accuracy level is small. At every iteration, the first scalarizing function (that is, one of the conic scalarization methods) is used to generate the minimal and properly minimal points with respect to the newly performed reference points. It may happen that some or all minimal and properly minimal points obtained by using the first scalarizing function at some iteration coincide with the ones obtained in the previous iteration. In such a case, the method will still generate a new set of reference points, as a result of the use of the second scalarizing function (that is, the one of the Pascoletti-Serafini method). Although the use of the second scalarizing function may lead to only weakly minimal points, it generates new ones if the stopping criteria are not satisfied yet. In the case when both the scalarizing functions generate new optimal points, this situation will positively affect the convergence speed of the method.

The rest of the paper is organized as follows. Section 2 gives some preliminaries. The computing algorithm is presented in Section 3. In this section, we also give some explanatory properties related to the algorithm and the convergence theorem. In Section 4, illustrative examples and compares the obtained results with related computational results from the literature. Finally, Section 5 draws some conclusions from the paper.

2. PRELIMINARIES

Throughout the paper, $\mathbb{R}^n_+ := \{y = (y_1, ..., y_n) : y_i \geq 0, i = 1, \ldots, n\}$, $\mathbb{R}_+$ denotes the set of nonnegative real numbers, and $\mathsf{cl}(\mathbb{Y})$, $\mathsf{bd}(\mathbb{Y})$, $\mathsf{int}(\mathbb{Y})$, and $\mathsf{co}(\mathbb{Y})$ denote the *closure*, the *boundary*, the *interior*, and the *convex hull* of a set $\mathbb{Y} \subset \mathbb{R}^n$, respectively.

Definition 2.1. Let $\mathbb{Y} \subset \mathbb{R}^n$ be a nonempty set.

(1) An element $y \in \mathbb{Y}$ is called a minimal point of $\mathbb{Y}$ if $(\{y\} - \mathbb{R}^n_+) \cap \mathbb{Y} = \{y\}$.
(2) An element $y \in \mathbb{Y}$ is called a weakly minimal point of $\mathbb{Y}$ if $(\{y\} - \mathsf{int}(\mathbb{R}^n_+)) \cap \mathbb{Y} = \emptyset$.
(3) An element $y \in \mathbb{Y}$ is called a properly minimal point of $\mathbb{Y}$ in the sense of Benson [3] if y is a minimal point of $\mathbb{Y}$ and the zero element of $\mathbb{R}^n$ is a minimal point of $\mathsf{cl}(\mathsf{cone}(\mathbb{Y} + \mathbb{R}^n_+ - \{y\}))$, where $\mathsf{cone}(\mathbb{Y}) := \{\lambda y : \lambda \geq 0, y \in \mathbb{Y}\}$.
(4) An element $\bar{y} \in \mathbb{Y}$ is called a properly minimal point of $\mathbb{Y}$ in the sense of Henig [15] if it is a minimal element of $\mathbb{Y}$ with respect to some convex cone $\mathbb{K}$ with $\mathbb{R}^n_+ \setminus \{0_{\mathbb{R}^n}\} \subset \mathsf{int}(\mathbb{K})$.

Consider a multi-objective optimization problem:

$$\min_{x \in \mathbb{X}} [f_1(x), ..., f_n(x)], \tag{2.1}$$

where $\mathbb{X}$ is a nonempty set of feasible solutions and $f_i : \mathbb{X} \to \mathbb{R}, i = 1, ..., n$ are real-valued functions. Let $f(x) = (f_1(x), \ldots, f_n(x))$ for every $x \in \mathbb{X}$ and let $\mathbb{Y} := f(\mathbb{X})$.

Definition 2.2. A feasible solution $x \in \mathbb{X}$ is called efficient, weakly efficient or properly efficient solution of multi-objective optimization problem (2.1) if $y = f(x)$ is a minimal, weakly minimal or properly minimal point of $\mathbb{Y} := f(\mathbb{X})$, respectively.

A nonempty subset $\mathbb{C}$ of $\mathbb{R}^n$ is called a *cone* if $y \in \mathbb{C}, \lambda \geq 0 \Rightarrow \lambda y \in \mathbb{C}$. Pointedness of $\mathbb{C}$ means that $\mathbb{C} \cap (-\mathbb{C}) = \{0_{\mathbb{R}^n}\}$. Let $\mathbb{C}$ be a given cone in $\mathbb{R}^n$. Recall that the dual cone $\mathbb{C}^*$ of $\mathbb{C}$ and its quasi-interior $\mathbb{C}^\#$ are defined by

$$\mathbb{C}^* = \{y^* \in \mathbb{R}^n : y^{*T} y \geq 0 \text{ for all } y \in \mathbb{C}\}$$

and

$$\mathbb{C}^\# = \{y^* \in \mathbb{R}^n : y^{*T} y > 0 \text{ for all } y \in \mathbb{C} \setminus \{0\}\},$$

respectively, where y^{*T} denotes the transpose of vector y^*, and $y^{*T} y = \sum_{i=1}^n y_i^* y_i$ is the scalar product of vectors $y^* = (y_1^*, \ldots y_n^*)$ and $y = (y_1, \ldots, y_n)$. The elements of these cones define monotone and strongly monotone linear functionals whose level sets (hyperplanes) are used to characterize support points of convex sets.

2.1. **Characterization of optimal points via Conic Scalarization (CS) method.** In this paper, the scalarizing function of the CS method is used to generate a set of properly minimal points. This method was introduced in [19] in general form for infinite-dimensional spaces, and was investigated in detail for $\mathbb{R}^n$ in [20] and [21, 22]. Scalarizing functions of this method were defined by using elements of augmented dual cones given below (see [20]).

$$\mathbb{C}^{a*} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha \leq w_i, w_i > 0, i = 1, \ldots, n\},$$

$$\mathbb{C}^{a\circ} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha \leq w_i, w_i > 0, i = 1, \ldots, n \\ \text{and there exists } k \in \{1, \cdots, n\} \text{ such that } w_k > \alpha\},$$

and

$$\mathbb{C}^{a\#} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha < w_i, i = 1, \ldots, n\},$$

where $\mathbb{C}$ is assumed to have a nonempty interior in the definition of $\mathbb{C}^{a\circ}$.

The following theorem, proved in [20, Theorems 4, 5, 6], characterizes optimal points computed by using scalarizing functions of the conic scalarization method.

Theorem 2.1. *Let $\mathbb{Y} \subset \mathbb{R}^n$ be a given nonempty set and let $r \in \mathbb{R}^n$ be a given (reference) vector. Denote $\mathbb{C} = \mathbb{R}^n_+$. Let $(w, \alpha) \in \mathbb{C}^{a*} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha \leq w_i, w_i > 0, i = 1, \ldots, n\}$, and let $Sol(SP)$ be the set of optimal solutions of the scalar optimization problem (SP):*

$$\min_{y \in \mathbb{Y}} \{w^T (y - r) + \alpha \|y - r\|_1\},$$

where $\|y\|_1 = |y_1| + ... + |y_n|$. Suppose that $Sol(SP) \neq \emptyset$ for a given pair $(w, \alpha) \in \mathbb{C}^{a*}$. Then the following hold.

(i) *If*

$$(w, \alpha) \in \mathbb{C}^{a\circ} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha \leq w_i, w_i > 0, i = 1, \ldots, n \\ \text{and there exists } k \in \{1, \cdots, n\} \text{ such that } w_k > \alpha\},$$

then every element of $Sol(SP)$ is a weakly minimal point of $\mathbb{Y}$.

(ii) *If $Sol(SP)$ consists of a single element, then this element is a minimal point of $\mathbb{Y}$.*

(iii) *If $(w, \alpha) \in \mathbb{C}^{a\#} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha < w_i, i = 1, \ldots, n\}$, then every element of $Sol(SP)$ is a properly minimal point of $\mathbb{Y}$ (in the sense of both Henig and Benson).*

(iv) *$\bar{y}$ is a properly minimal point of $\mathbb{Y}$ (in the sense of Benson or Henig) if and only if there exists a pair $(w, \alpha) \in \mathbb{C}^{a\#} = \{((w_1, \ldots, w_n), \alpha) : 0 \leq \alpha < w_i, i = 1, \ldots, n\}$ such that $\min_{y \in \mathbb{Y}} \{w^T (y - \bar{y}) + \alpha \|y - \bar{y}\|_1\}$ attains its minimum at $\bar{y}$.*

3. COMPUTING OPTIMAL POINTS CORRESPONDING TO THE GIVEN PREFERENCE PARAMETERS

Consider multi-objective optimization problem (2.1). In this section, we describe the method for computing minimal, properly minimal, and weakly minimal points of $Y+R^n_+$, corresponding to the given set of weights and the reference points. The method also generates the set of approximate weakly minimal points that remain between the given reference (or ideal) point $r \in R^n$ and the set $cl\left((\{r\}+R^n_+)\backslash(Y+R^n_+)\right)$. For the rest of this paper, we have the following assumption on Y.

Assumption 3.1. We assume that $f(\mathbb{X})$ is a closed set. In the case that problem

$$r_i = \min_{y\in f(\mathbb{X})} y_i, \quad i=1,\dots,n. \tag{3.1}$$

has a finite solution $r=(r_1,\dots,r_n)$, this solution is called the ideal point of the set $Y=f(\mathbb{X})$, which is used as the initial reference point in the method suggested in this paper. In this case, if problem (3.1) has no finite solution for all $i=1,\dots,n$, we ask the decision-maker to select a reference point $r=(r_1,\dots,r_n)\notin f(\mathbb{X})$ such that $Y=\mathrm{cl}((r+\mathbb{R}^n_+)\cap f(\mathbb{X}))\neq\emptyset$ and $\mathrm{cl}((r+\mathbb{R}^n_+)\backslash(Y+\mathbb{R}^n_+))$ is a compact set. It is clear that, in this case, problem

$$\min_{y\in Y} y_i \tag{3.2}$$

has a finite solution r_i for all $i=1,\dots,n$. Without loss of generality, we assume that the above conditions are satisfied, and the unique notation Y is used for the sets $Y=f(\mathbb{X})$ or $Y=\mathrm{cl}((r+\mathbb{R}^n_+)\cap f(\mathbb{X}))$. We also assume that the reference point is computed according to (3.1) or (3.2).

Let $w=(w_1,\dots,w_n)$ be a given vector of weights, and let

$$L=\{\alpha_0,\alpha_1,\alpha_2,\dots,\alpha_P\} \tag{3.3}$$

be a partition of $[0,\min_{i=1,\dots,n}\{w_i\}]$ such that

$$0=\alpha_0<\alpha_1<\alpha_2<\ldots<\alpha_P=\min_{i=1,\dots,n}\{w_i\}.$$

To compute the minimal and properly minimal points corresponding to the given set of parameters (w,α_p,r) with $p=0,1,2,\dots,P$, the following problem will be solved:

$$\min_{y\in Y} g_{(w,\alpha_p,r)}(y)=w(y-r)+\alpha_p\|y-r\|_1. \tag{3.4}$$

From the definition, we have $(w,\alpha_p)\in C^{a\#}$ for every $p=0,1,2,\dots,P-1$. Theorem 2.1 (iii)-(iv) yield that all the solutions obtained by solving problems (3.4) are properly minimal points for all $p=0,1,2,\dots,P-1$.

The partition L is used here to generate a set of augmentation parameters α_p. It is clear that there may exist different properly minimal points corresponding to the same weight vector w for different choices of augmentation parameters. This becomes available due to the fact that the sublevel set of the scalarizing function g used in (3.4) is a conic set and actually solving the problem (3.4) gives supporting points of Y with respect to this cone. By changing the augmentation parameter α, the pair (w,α) will generate a "new" scalarizing function with a different sublevel set. Since $\alpha_p>\alpha_{p-1},\forall p=1,\dots,P$, by construction, the cone corresponding to the sublevel set of $g_{(w,\alpha_p,r)}$ is narrower than the one for $g_{(w,\alpha_{p-1},r)}$, and therefore the solution set of the problem (3.4) using the functions $g_{(w,\alpha_p,r)}$ for all values of the augmentation parameter $\alpha_p \in L$ may guarantee the finding of different supporting points of Y w.r.t. the parameter set $\{(w,\alpha_p,r) : p = 0,1,\dots,P\}$, which will become the properly minimal or minimal points. The reason for using a partition L is that in nonconvex cases, the set Y may have "hidden" (properly) minimal points. These points cannot be supported by hyperplanes but can be supported by a sufficiently sharp conic. surfaces that are close to the cone $-R^n_+$. A sublevel set of the function $g_{(w,\alpha_p,r)}$ contains the cone $-R^n_+$ and becomes sharper and closer to the cone. $-R^n_+$ as α_p is closer to its maximum possible value $\alpha_P = \min_{i=1,\dots,n}\{w_i\}$, with

$$[(\{r\} - R^n_+) \setminus \{r\}] \subset \operatorname{int}(\{y \in R^n : g_{(w,\alpha_p,r)}(y) \leq 0\}),$$

for all $p = 0,1,\dots,P-1$ (see [20, Remark 2]).

On the other hand, the presented method will also generate the set of weakly minimal points. For this reason, we use solutions of the following problem:

$$\min\ t \quad \text{s.t.} \quad r + tc \in Y + R^n_+, t \geq 0, \tag{3.5}$$

where $c \in int(R^n_+)$. The solutions to this problem will be used to generate some part of the weakly minimal boundary of the set $\operatorname{cl}((r+\mathbb{R}^n_+)\cap(Y+\mathbb{R}^n_+))$. These points are used to generate a grid of "new reference points" in the set $\operatorname{cl}((r+\mathbb{R}^n_+)\setminus(Y+\mathbb{R}^n_+))$, by removing a definite part of this set at every iteration such that the sequence of remaining parts will converge to the weakly minimal boundary. The generated set of "new reference points" will also be used to compute "possibly" new minimal and/or properly minimal points of Y with respect to the same weight vector w, for all possible values of augmentation parameters $\alpha \in L$.

3.1. **Algorithm.** In this section, we introduce the algorithm for computing optimal and approximately optimal points of the multi-objective optimization problem (2.1), corresponding to a given set of weights w and the given initial reference point r. In regard to the set $Y = f(X)$ and the reference point r, we assume that the conditions of Assumption 3.1 are satisfied. We also assume that the partition L is defined as in (3.3).

Algorithm: Initialization Let $w = (w_1,\dots,w_n)$ be a given vector of weights, and let $c = (c_1,\dots,c_n) \in \operatorname{int}(R^n_+)$. Let $r^0_{0i} = \min_{y\in Y} y_i,\quad i = 1,\dots,n$, and $r^0_0 = (r^0_{0_1}, r^0_{0_2},\dots,r^0_{0_n})$. Let $y^i_{0_i} = \min\{y_i : y = (y_1,\dots,y_n) \in Y, y_j = r^0_{0j} \quad j = 1,\dots,n, j \neq i\}$, $i = 1,\dots,n$. Then, we denote $y^1_0 = (y^1_{0_1}, r^0_{0_2},\dots,r^0_{0_n})$, $y^2_0 = (r^0_{0_1}, y^2_{0_2}, r^0_{0_3},\dots,r^0_{0_n})$, and $y^n_0 = (r^0_{0_1},\dots,r^0_{0_{n-1}}, y^n_{0_n})$. Let the partition L be defined as in (3.3), and let $\varepsilon \in (0,1)$ be the accuracy level, which is used for the stopping criterion in Steps 2 and 3.

Let

0a. $Y_0 = \operatorname{cl}((\{r^0_0\} + R^n_+) \setminus (Y + R^n_+))$.

0b. $E_0 = \{y^1_0, y^2_0,\dots,y^n_0.\}$

0c. $R_0 = \{r^0_0\}, V_0 = \emptyset, J_0 = \{0\}$.

0d. $k = 0$.

Step 1. 1a. For all $j \in J_k, \alpha \in L$, and $r^j_k \in R_k$, solve the following two problems:

$$\min_{y\in Y} g_{(w,\alpha,r^j_k)}(y) = w(y - r^j_k) + \alpha\|y - r^j_k\|_1, \tag{3.6}$$

$$\min\ t \quad \text{s.t.} \quad r^j_k + tc \in Y + R^n_+, t \geq 0. \tag{3.7}$$

Let $y^j_{k\alpha}$ be an optimal solution to (3.6), let t^j_k be an optimal solution to (3.7), and let $a^j_k = r^j_k + t^j_k c$ for all $j \in J_k$.

1b. Let

$$E(r^j_k) := \cup_{\alpha \in L}\{y^j_{k\alpha} \in Y:\ y^j_{k\alpha} = \operatorname*{argmin}_{y \in Y} g_{(w,\alpha,r^j_k)}(y),\} \text{ for every } j \in J_k,$$

$$G_k = \cup_{j \in J_k} \cup_{r^j_k \in R_k} E(r^j_k),$$

$$T_k = \cup_{j \in J_k} \cup_{r^j_k \in R_k} \{a^j_k = r^j_k + t^j_k c\}.$$

$$E_{k+1} = E_k \cup G_k \cup T_k.$$

Step 2. Let

$$T_k = T_k \setminus \cup_{j \in J_k}\{a^j_k : t^j_k < \varepsilon\},$$
$$V_k = V_k \cup_{j \in J_k} \{r^j_k \in R_k : t^j_k < \varepsilon\}.$$

Step 3. If $T_k = \emptyset$, then stop: E_{k+1} is the set of weakly minimal, minimal, and properly minimal points, and V_k is the set of approximate weakly minimal points of $Y + R^n_+$ corresponding to the given set of weights $\{w_1, \ldots, w_n\}$ and the reference points $\cup_{i=0}^{i=k} \cup_{j \in J_k} r^j_i$.

Otherwise, let

3a. $Y_{k+1} = Y_k \setminus \cup_{h \in (E_{k+1})}(\{h\} - \operatorname{int}(R^n_+))$,

3b. Set $R_{k+1} = \min(Y_{k+1})$ and $J_{k+1} = \{1, \ldots, |R_{k+1}|\}$. Use the following steps 3b1-3b3 to compute R_{k+1} and go to Step 3c.

3b1. Let S be a set of all subsets $H \subseteq (E_{k+1})$ with $2 \leq |H| \leq n:$

$$S = \{H = \{h_1, \cdots, h_q\} \subseteq (E_{k+1}) : 2 \leq q \leq n.\}$$

For every set $H = \{h_1, \cdots, h_q\} \in S$, where $2 \leq q \leq n$, define

$$b_H := (\min\{h_{11}, \cdots, h_{q1}\}, \cdots, \min\{h_{1n}, \cdots, h_{qn}\}).$$

3b2. Set $B_{k+1} := \{b_H :\ H \in S \text{ and there exists no } h \in (E_{k+1}) \text{ such that } b_H < h.\}$

3b3. Set $R_{k+1} := B_{k+1} \backslash \{b_H \in B_{k+1} :\ \text{there exists } b_{H'} \in B_{k+1} \text{ such that } b_{H'} \leq b_H\}$.

3c. Set $R_{k+1} = R_{k+1} \setminus E_{k+1}$, $k = k+1$ and go to Step 1.

3.2. **Discussion on the algorithm.** This section starts with an explanation of the algorithm presented above.

At every iteration k, the algorithm generates and/or updates the sets $Y_k, R_k, E_k, G_k, T_k, V_k$ and J_k where $k = 0, 1, 2, \ldots$. The set R_0 consists of a single point $r^0_0 = r$, where r is the initial reference or ideal point. Y_0 is the initial set which is defined as the part of the set $\{r^0_0 + R^n_+\}$ remaining between r^0_0 and the weakly minimal boundary of $Y + R^n_+$. The set E_0 is the initial set of minimal points of Y, at least one component of which equals the corresponding component of the ideal point r^0_0. The set J_0 consists of a single number 0, but all consequent sets are defined as a set of positive integers from 1 to the number of points in R_k: $J_k = \{1, \ldots, |R_k|\}$.

At Step $1a$ of every iteration k, two kinds of optimization problems are solved.

Problem (3.6) minimizes the scalarizing function $g_{(w,r_k^j,\alpha)}$ of the conic scalarization method on Y, for every $j \in J_k, \alpha \in L$, and $r_k^j \in R_k$. For the given weight w and each "reference point" $r_k^j \in R_k$, the problem (3.6) is solved for all augmentation parameters $\alpha \in L$ so that to "catch" all possible minimal and properly minimal points corresponding to these scalarizing parameters. All these optimal points obtained are collected in the set G_k at every iteration k. Thus, the set G_k is formed as a current set of minimal and properly minimal points at iteration k.

The problem (3.7), solved at Step 1a for every $j \in J_k$, and $r_k^j \in R_k$, minimizes the scalarizing function of the Pascoletti-Serafini (PS) method and computes a weakly minimal element $a_k^j = r_k^j + t_k^j c$ for every "reference point" $r_k^j \in R_k$ and the "direction vector" $c \in \operatorname{int}(R_+^n)$. Then it is clear that, $r_k^j \in (\{a_k^j\} - \operatorname{int}(R_+^n))$. All these weakly minimal points are collected in the set T_k at every iteration k.

The set E_{k+1} is then updated as a cumulative set of all weakly minimal, minimal, and properly minimal sets. Points are computed up to the iteration $k+1$ at Step 1b. At this step, we also update the set T_k by removing the newly obtained optimal points a_k^j, whose "distances" t_k^j to the corresponding "reference" points r_k^j are less than the given accuracy level ε. This helps us not to generate the useless "reference points" for the next iteration. On the other hand, the "reference points" r_k^j corresponding to $t_k^j < \varepsilon$ are collected in the set V_k of approximate weakly minimal points.

For every minimal, properly minimal, and weakly minimal element $h \in (E_{k+1})$, the combination of all sets $\cup_{h\in(E_{k+1})}(h - \operatorname{int}(R_+^n))$ is subtracted from Y_k and by this way, the set Y_{k+1} is performed. The resulting set Y_{k+1} is strongly smaller than the set Y_k, because it is obtained by subtracting a set $(h - \operatorname{int}(R_+^n))$ from Y_k, where at least the element $h \in T_k$ is new, and is different from all points computed as solutions of the problem (3.7) in previous iterations. This is due to the fact that the vector c in problem (3.7) is chosen from the interior of R_+^n, every element r_k^j of R_k is removed from Y_{k+1} together with the set $h - \operatorname{int}(R_+^n)$ for $h = a_k^j$ at the iteration k, and the set R_{k+1} is formed as a set of minimal points of Y_{k+1}. The algorithm terminates if T_k updated at Step 2 becomes an empty set, which means that all the new optimal points a_k^j computed as optimal solutions to the problem (3.7) at Step 1a correspond to the numbers t_k^j with $t_k^j < \varepsilon$.

Remark 3.1. In this paper, we do not aim to give methods for solving the scalar problems (3.6) and (3.7). We assume that the reader has access to methods for solving nonconvex, nonsmooth single-objective problems and can use those to find optimal solutions; see, for example, [12] and [18].

Definition 3.1. $Y_{min} := (Y + R_+^n) \cap Y_0$ is called the weakly minimal boundary of Y.

The following proposition explains the steps of the Algorithm.

Proposition 3.1. (a) *$Y_{k+1} \subset Y_k$ for all k.*
(b) *$Y_{min} \subseteq Y_k$ for all k.*
(c) *$A_k \subset Y_k$ for all k.*
(d) *There exists an optimal solution t_k^j to (3.7) for every k and every weakly minimal set is a scalarizing function $j \in J_k$ such that $a_k^j = r_k^j + t_k^j c \in Y + R_+^n$ and a_k^j is a weakly minimal element of $Y + R_+^n$.*

Proof. (a), (b), and (c) are obvious by construction.

(d) Obviously there exists $t \in \mathbb{R}$ such that $r_k^j + tc \in Y + \mathbb{R}_+^n$. The proof follows from the compactness of the set $(\{r_k^j + tc\} - \mathbb{R}_+^n) \cap (Y + \mathbb{R}_+^n)$ and continuity of the objective function of the problem (3.7) (see also [28, 30]).

□

The following proposition proves that the subalgorithm given in Step 3b computes the set of minimal points of Y_k for every k = 1, 2, ...

Proposition 3.2. *Let R_{k+1} be a set given in Steps 3b1 - 3b3 of the Algorithm. Then $R_{k+1} = \min(Y_{k+1})$ for every $k = 1, 2, \dots$.*

Proof. For every subset $H = \{h_1, \cdots, h_q\} \in S$, the elements b_H in Step 3b1 are obtained as intersection $\cap_{m=1}^q (h_m - \partial\mathbb{R}_+^n)$ for some elements $h_m \in (E_{k+1})$, $m = 1, 2, \dots, q$. Then B_{k+1} is formed in Step 3b2, by removing elements b_H such that $b_H < h$ for some $h \in (E_{k+1})$. Finally, at Step 3b3, R_{k+1} is formed as the set of minimal points of B_{k+1}. Note that the same way is used in Step 3a to obtain the set Y_{k+1}. It follows from the definition of Y_{k+1} that

$$\min(Y_{k+1}) \subseteq \{\cup_{h \in (E_{k+1})} (h - \partial\mathbb{R}_+^n)\} \cap Y_{k+1}. \tag{3.8}$$

Denote $M_{k+1} = \min(Y_{k+1})$ for every $k = 1, 2, \dots$. We first prove $M_{k+1} \subseteq R_{k+1}$. Let $y \in M_{k+1}$. By construction of Y_{k+1} and (3.8), for all $i \in \{1, \cdots, n\}$ there exists $h \in (E_{k+1})$ such that

$$y \in h - \partial\mathbb{R}_+^n \quad \text{and} \quad \forall i \in \{1, \cdots, n\} \quad \exists h \in (E_{k+1}) \quad s.t. \quad y_i = h_i. \tag{3.9}$$

Let $H := \{h^1, \cdots, h^l\} \subseteq (E_{k+1})$ be a set of all $h \in (E_{k+1})$ satisfying (3.9), and let

$$b_H := \left(\min\{h_1^1, \cdots, h_1^l\}, \cdots, \min\{h_n^1, \cdots, h_n^l\}\right). \tag{3.10}$$

Obviously, by construction and by (3.8), (3.9), (3.10),and Step 3b3, we have $b_H \in R_k, y = b_H$ and hence $y \in R_k$.

Now, we prove that $R_{k+1} \subseteq M_{k+1}$. Obviously, $R_{k+1} \subseteq Y_{k+1}$. Suppose, to the contrary, that there exists $b \in R_{k+1}$ such that $b \notin M_{k+1}$. Then, by definition of $M_{k+1} = \min(Y_{k+1})$, there exists $y \in M_{k+1}$ such that $y \leq b$. By the first part ($M_{k+1} \subseteq R_{k+1}$), we get $y \in R_{k+1}$, and hence $y \in B_{k+1}$. If $b = y$, then $b \in M_{k+1}$ and the proof is complete. In the case when $b \neq y$, then, by the inequality $y \leq b$, $y \in B_{k+1}$ and, by Steps 3b2-3b3 in subalgorithm, we get $b \notin B_{k+1}$. This is a contradiction which proves the proposition. □

The following theorem proves that the sequence of sets Y_k, iteratively constructed by the Algorithm, converges to the weakly minimal boundary of the set $Y + R_+^n$.

Theorem 3.1. *Let Y_k be the sequence of sets generated at Step 3a of the Algorithm. Then $\lim_{k\to\infty} Y_k = Y_{min}$, where Y_{min} is a weakly minimal boundary of $Y + R_+^n$ defined by Definition 3.1.*

Proof. By Proposition 3.1(a), we have $Y_{k+1} \subset Y_k$ for all k. By Remark 3.1, Y_{k+1} is obtained by removing a set with nonempty interior from Y_k which makes it strongly smaller than Y_k. On the other hand, the elements of R_k become closer to the set Y_{min} at every iteration k than the elements of R_{k-1}, which can be explained using the steps of the Algorithm. It is clear that, by $R_0 = \{r_0^0\}$ and the Step 3b of the algorithm, this set can also be determined as $\cap_{h \in E_0} \left(\{h\} - \partial\mathbb{R}_+^n\right)$ (see

(3.8)). Clearly, the elements of E_0 are at the same time elements of Y_{min}, and at the initial iteration, are all on the axes $\left(\{r_0^0\}+\partial\mathbb{R}^n_+\right)$ which are parallel to coordinate axes. Recall that these elements are denoted by $y_0^1, y_0^2, \ldots, y_0^n$, where y_0^i lies on the axis parallel to the x_i-axis for $i=1,\ldots,n$. Then, since $y_0^1=(y_{0_1}^1, r_{0_2}^0, \ldots, r_{0_n}^0), y_0^2=(r_{0_1}^0, y_{0_2}^2, 0, \ldots, r_{0_n}^0)$, and $y_0^n=(r_{0_1}^0, \ldots, r_{0_{n-1}}^0, y_{0_n}^n)$, the distances between the point $r_0^0 \in R_0$ and $y_0^i \in Y_{min}$ for $i=1,\ldots,n$, can be evaluated as $y_{0_i}^i - r_{0_i}^0$ for each $i=1,\ldots,n$. At the first iteration, new minimal and/or weakly minimal points in Y_{min} are computed as a result of solving the problems (3.6) and (3.7). In the Step 1b, the solutions of problem (3.6) are collected in G_0, and the solutions of (3.7) are first collected in T_0 and then all are combined with E_0 to obtain E_1. For simplicity of the explanation, assume that the solution of the problem (3.6) coincides with some of the points existing in E_0, and just one new point is added to the set $G_0 \cup T_0$ as a solution to the problem (3.7), (by using the reference point r_0^0). Denote this new element by a_0^0 (see Step 1a). Note that the problem (3.7) always will generate new elements while R_k is not empty, because of the condition $c \in \operatorname{int}(R^n_+)$. Then, set Y_1 will be obtained as $Y_0 \setminus (\{a_0^0\} - \operatorname{int}(R^n_+))$. Obviously, the set Y_1 is smaller than Y_0, and the elements of R_1 will be determined as minimal points of Y_1, which can be calculated according to the Step 3a of the Algorithm, as the points of intersection of the set $(\{a_0^0\} - \partial\mathbb{R}^n_+)$ with the set $(\{r_0^0\} + \partial\mathbb{R}^n_+)$. For this special case, it is clear that the sets J_1 and R_1 can be determined as $J_1=\{1,\ldots,n\}$, and $R_1=\{r_1^1, r_1^2, \ldots, r_1^n\}$. Again, without loss of generality, we can assume r_1^j is on the line, parallel to the x_j axis for $j=1,\ldots,n$. Then it is clear from these explanations that we have $r_{0_j}^0 < r_{1_j}^j < y_{0_j}^j$. Note that the set Y_1 can be considered as a set which consists of n sets like Y_0, every jth of which has a "reference point" r_1^j, like r_0^0 for Y_0. The process will continue in this manner, and clearly, the points of R_k at each iteration k will be strictly closer to $Y_{\min}$ than those of R_{k-1}. Since $Y_{\min} \subseteq Y_k$ for all k by Proposition 3.1(b), we conclude that $Y_k \to Y_{\min}$, and the proof is complete. □

4. NUMERICAL RESULTS

In this section, we provide solutions for an illustrative example as well as for well-known multi-objective optimization test problems to demonstrate the performance of the proposed method. The algorithm has been implemented in Python 3.11 using the VS Code Studio 1.87.2 source-code editor. The SciPy library is utilized for the optimization of the functions.

4.1. **Illustrative example.** Here, we present an illustrative example that explains the steps of the algorithm given in the previous section.

Example 4.1.

$$\begin{aligned}
\textit{Let}\quad & f_1(x_1,x_2)=x_1,\\
& f_2(x_1,x_2)=x_2,\\
& g(x_1,x_2)=x_1^2+x_2^2-4,\\
& X=\{(x_1,x_2): g(x_1,x_2)\le 0, -2\le x_1\le 1, -2\le x_2\le 1\}\\
& \cup\{(x_1,x_2): -4\le x_1\le 0, 0\le x_2\le 1\}\cup\{(x_1,x_2): -4\le x_2\le 0, 0\le x_1\le 1\}.
\end{aligned}$$

Consider the following problem: Min$[f_1(x_1,x_2), f_2(x_1,x_2)]$ subject to $(x_1,x_2)\in X$.

Now we demonstrate the five iterations of the Algorithm on Example 4.1. All iterations are graphically illustrated in Figures 1 - 5 where the reference points r_k^j are shown in red, the

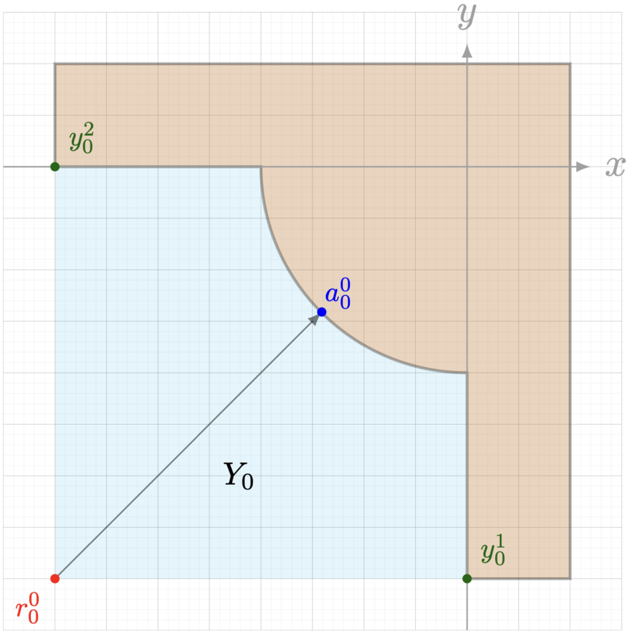


FIGURE 1 Iteration 1

efficient solutions y_k^j obtained by solving the Conic Scalarization method in green, and the efficient solutions a_k^j obtained by solving the Pascoletti-Serafini method in blue.

Initialization. Set $k = 0$. Let $\varepsilon = 0.15$, $w = (25,24)$, $c = (1,1)$, $\alpha \in \{1,\dots,24\}$, $r_0^0 = (r_{0_1}^0, r_{0_2}^0) = (-4,-4)$, $y_0^1 = (0,-4), y_0^2 = (-4,0), Y = X$. Then we obtain $V_0 = \emptyset, J_0 = \{0\}$, $E_0 = \{(0,-4),(-4,0)\}, R_0 = (-4,-4)$, $Y_0 = \mathrm{cl}((\{r_0^0\} + R_+^2) \setminus (Y + R_+^2))$.

Iteration 1.
Solving CS with the reference point $r_0^0 = (-4,-4)$ leads to $y_0^0 = (-4,0)$.
Solving PS with the reference point $r_0^0 = (-4,-4)$ leads to $a_0^0 = (-1.41,-1.41)$.
Hence we obtain $G_0 = \{y_0^1, y_0^2\} = \{(0,-4),(-4,0)\}$ and $T_0 = \{a_0^0\} = \{(-1.41,-1.41)\}$.
$E_1 = E_0 \cup G_0 \cup T_0 = \{(0,-4),(-4,0),(-1.41,-1.41)\}$.
Since $t_0^0 = 2.59 > \varepsilon = 0.15$, we continue with $V_0 = \emptyset$.
$Y_1 = Y_0 \setminus \cup_{h \in (E_1)} (\{h\} - \mathrm{int}(R_+^2))$.
$R_1 = \min Y_1 = \{r_1^1, r_1^2\} = \{(-1.41,-4),(-4,-1.41)\}, J_1 = \{1,2\}, k = 1$ (or illustration, see Figure 1).

Iteration 2.
Solving CS with $r_1^1 = (-1.41,-4)$ and $r_1^2 = (-4,-1.41)$ leads to $y_1^1 = (0,-4)$ and $y_1^2 = (-4,0)$, respectively.
Solving PS with $r_1^1 = (-1.41,-4)$ and $r_1^2 = (-4,-1.41)$ leads to $a_1^1 = (0,-2.59)$ and $a_1^2 = (-2.59,0)$, respectively. The reference points and efficient solutions together with the t_k^j numbers obtained in Step 1 of Iteration 2 are depicted in Table 1.
Set $G_1 = \{y_1^1, y_1^2\} = \{(0,-4),(-4,0)\}$, $T_1 = \{a_1^1, a_1^2\} = \{(0,-2.59),(-2.59,0)\}$.
$E_2 = E_1 \cup G_1 \cup T_1 = \{(0,-4),(-4,0),(-1.41,-1.41),(0,-2.59),(-2.59,0)\}$.

r_1^1	-1.41	-4	a_1^1	0	-2.59	t_1^1	1.41	y_1^1	-0	-4
r_1^2	-4	-1.41	a_1^2	-2.59	0	t_1^2	1.41	y_1^2	-4	0

TABLE 1 Reference points used and efficient solutions obtained in Iteration 2.

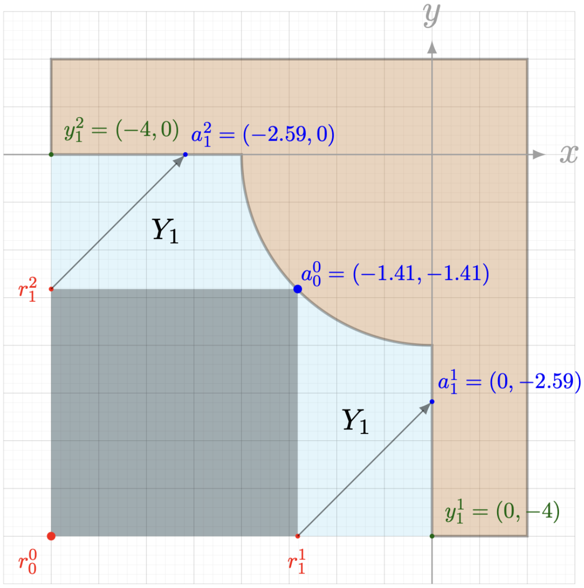


FIGURE 2 Iteration 2

Since $t_1^j > \varepsilon = 0.15$ for $j = 1, 2$, we continue with $V_1 = \emptyset$.

$Y_2 = Y_1 \setminus \cup_{h \in (E_2)} (\{h\} - \text{int}(R_+^2))$.

$R_2 = \min Y_2 = \{r_2^1, r_2^2\} = \{(-1.41, -2.59), (-2.59, -1.41), \}$, $J_2 = \{1, 2\}$, $k = 2$ (for illustration, see Figure 2).

Iteration 3.

Solving CS with $r_2^1 = (-1.41, -2.59)$ and $r_2^2 = (-2.59, -1.41)$ leads to $y_2^1 = (-1.40, -1.42)$ and $y_2^2 = (-1.51, -1.30)$, respectively.

Solving PS with $r_2^1 = (-1.41, -2.59)$ and $r_2^2 = (-2.59, -1.41)$ leads to $a_2^1 = (-0.69, -1.87)$ and $a_2^2 = (-1.87, -0.69)$, respectively. The reference points and efficient solutions together with the t_k^j numbers obtained in Step 1 of Iteration 2 are depicted in Table 2.

Set

$$G_2 = \{y_2^1, y_2^2\} = \{(-1.40, -1.42), (-1.51, -1.30)\}$$

and

$$T_2 = \{a_2^1, a_2^2\} = \{(-0.69, -1.87), (-1.87, -0.69)\}.$$

$E_3 = E_2 \cup G_2 \cup T_2 = \{(0, -4), (-4, 0), (0, -2.59), (-2.59, 0), (-1.40, -1.42), (-1.51, -1.30), (-0.69, -1.87), (-1.87, -0.69), (-1.41, -1.41)\}$.

Since $t_2^j > \varepsilon = 0.15$ for $j = 1, 2$, we continue with $V_2 = \emptyset$.

$Y_3 = Y_2 \setminus \cup_{h \in (E_3)} (\{h\} - \text{int}(R_+^2))$.

r_1^1	-1.41	-2.59	a_2^1	-0.69	-1.87	t_2^1	0.72	y_2^1	-1.40	-1.42
r_1^2	-2.59	-1.41	a_2^2	-1.87	-0.69	t_2^2	0.72	y_2^2	-1.51	-1.30

TABLE 2 Reference points used and efficient solutions obtained in Iteration 3.

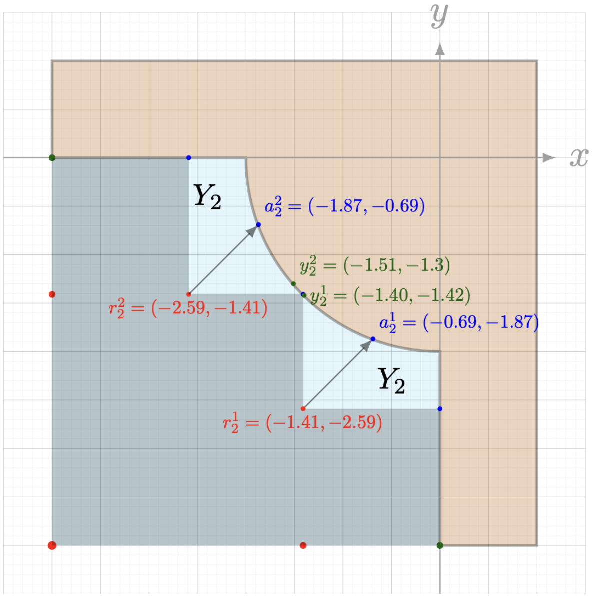


FIGURE 3 Iteration 3

$R_3 = \min Y_3 = \{r_3^1, r_3^2, r_3^3, r_3^4, r_3^5, r_3^6\}$, $J_3 = \{1,2,3,4,5,6\}$, $k = 3$ (for illustration see Figure 3).

Iteration 4. The reference points and efficient solutions together with the t_k^j numbers obtained in Step 1 of Iteration 4 are depicted in Table 3.

Set

$$G_3 = \{y_3^1, y_3^2, y_3^3, y_3^4, y_3^5, y_3^6\}$$

and

$$T_3 = \{a_3^1, a_3^2, a_3^3, a_3^4, a_3^5, a_3^6\}.$$

As it is presented in Table 3, $t_3^3 = 0.002 < 0.1 = \varepsilon$ and $t_3^4 = 0.04 < 0.1 = \varepsilon$. Therefore, the corresponding optimal points $a_3^3 = (-1.409, -1.419)$, and $a_3^4 = (-1.463, -1.363)$ should be removed from T_3 when we compute

$$Y_4 = Y_3 \setminus \cup_{h \in (E_4)} (\{h\} - \operatorname{int}(R_+^2)).$$

On the other hand, this situation leads to the updated set of approximate weakly minimal points $V_3 = \{r_3^3, r_3^4\}$.

$R_4 = \min Y_4 = \{r_4^1, \ldots, r_4^{12}\}$, $J_4 = \{1, \ldots, 12\}$, $k = 4$ (for illustration, see Figure 4).

Iteration 5.

r_3^1	-0.69	-2.59	a_3^1	-0.1	-2.0	t_3^1	0.59	y_3^1	0	-4.0
r_3^2	-1.40	-1.87	a_3^2	-1.16	-1.63	t_3^2	0.242	y_3^2	-1.335	-1.485
r_3^3	-1.41	-1.42	a_3^3	-1.409	-1.419	t_3^3	0.002	y_3^3	-1.405	-1.415
r_3^4	-1.51	-1.41	a_3^4	-1.463	-1.363	t_3^4	0.04	y_3^4	-1.485	-1.335
r_3^5	-1.87	-1.30	a_3^5	-1.67	-1.1	t_3^5	0.2	y_3^5	-1.515	-1.305
r_3^6	-2.59	-0.69	a_3^6	-2.0	-0.1	t_3^6	0.59	y_3^6	-4.0	0

TABLE 3 Reference points used and efficient solutions obtained in Iteration 4.

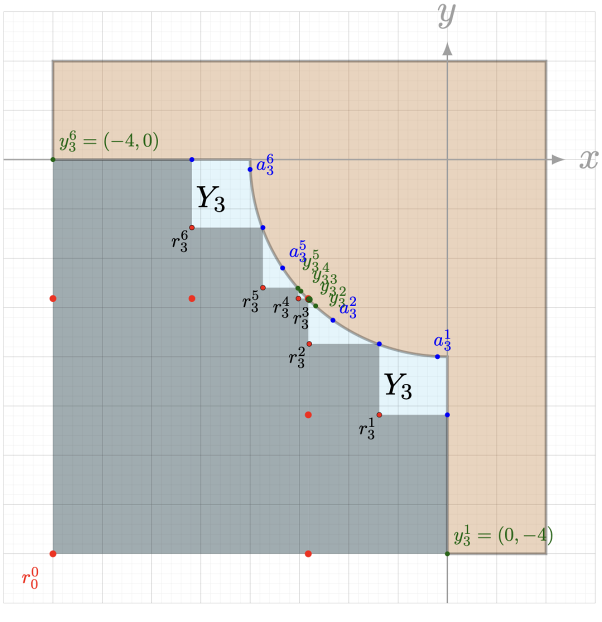


FIGURE 4 Iteration 4

The reference points and efficient solutions together with the t_k^j numbers obtained in Step 1 of Iteration 5 are presented in Table 4 and depicted in Figure 5. According to these calculations, we obtain $G_4 = \{y_4^1, \dots, y_4^{12}\}$ and $T_4 = \{a_4^1, \dots, a_4^{12}\}$.

As it can be seen from Table 4, $t_4^j < 0.1$ for $j = 2, 4, 5, 6, 7, 8, 9, 11$. Therefore the corresponding optimal points a_4^j should be removed from the set T_4 when we compute

$$Y_5 = Y_4 \setminus \cup_{h \in (E_5)} (\{h\} - \text{int}(R_+^2)).$$

On the other hand, this situation leads to the updated set of approximate weakly minimal points $V_4 = \{r_3^3, r_3^4, r_4^2, r_4^4, r_4^5, r_4^6, r_4^7, r_4^8, r_4^9, r_4^{11}\}$. For more clarity, the points $r_4^2, r_4^4, r_4^5, r_4^6, r_4^7, r_4^8, r_4^9, r_4^{11}$ are shown with orange in Fig. 5.

Not to lengthen the article artificially, we do not continue the iterations, as we think the given four iterations explain the algorithm well.

r_4^1	-0.1	-2.59	a_4^1	0	-2.49	t_4^1	0.1	y_4^1	0	-4
r_4^2	-0.69	-2.0	a_4^2	-0.598	-1.908	t_4^2	0.093	y_4^2	-0.693	-1.876
r_4^3	-1.165	-1.87	a_4^3	-1.017	-1.722	t_4^3	0.149	y_4^3	-1.164	-1.625
r_4^4	-1.30	-1.625	a_4^4	-1.242	-1.567	t_4^4	0.059	y_4^4	-1.305	-1.515
r_4^5	-1.41	-1.52	a_4^5	-1.358	-1.468	t_4^5	0.053	y_4^5	-1.405	-1.415
r_4^6	-1.458	-1.41	a_4^6	-1.438	-1.390	t_4^6	0.02	y_4^6	-1.445	-1.375
r_4^7	-1.485	-1.368	a_4^7	-1.471	-1.354	t_4^7	0.015	y_4^7	-1.485	-1.335
r_4^8	-1.515	-1.335	a_4^8	-1.501	-1.321	t_4^8	0.015	y_4^8	-1.485	-1.335
r_4^9	-1.67	-1.30	a_4^9	-1.587	-1.217	t_4^9	0.084	y_4^9	-1.515	-1.305
r_4^{10}	-1.87	-1.1	a_4^{10}	-1.745	-0.975	t_4^{10}	0.12	y_4^{10}	-1.665	-1.104
r_4^{11}	-2.0	-0.69	a_4^{11}	-1.908	-0.598	t_4^{11}	0.093	y_4^{11}	-1.876	-0.693
r_4^{12}	-2.59	-0.1	a_4^{12}	-2.49	0	t_4^{12}	0.1	y_4^{12}	-4	0

TABLE 4 Reference points used and efficient solutions obtained in Iteration 5.

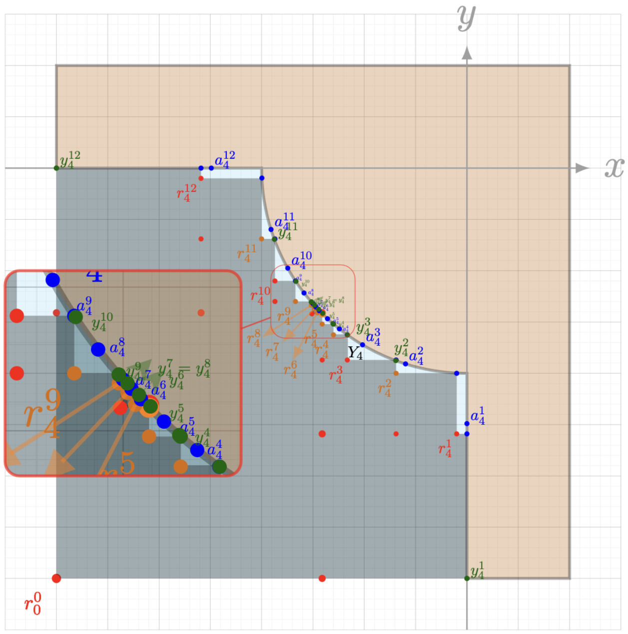
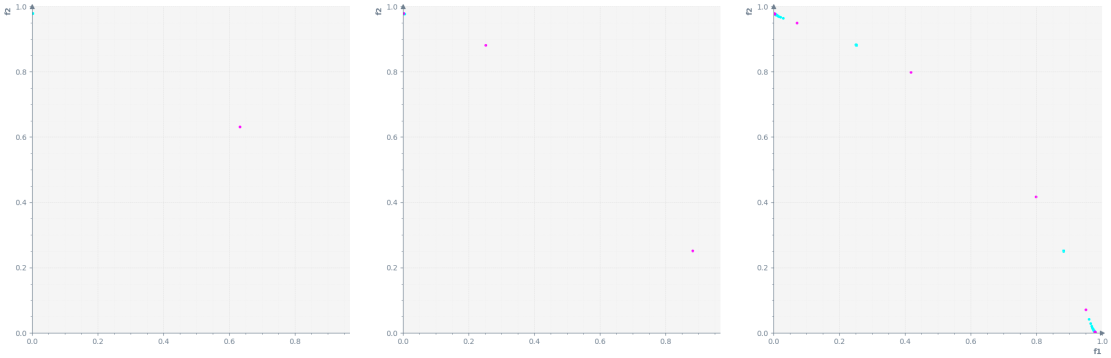


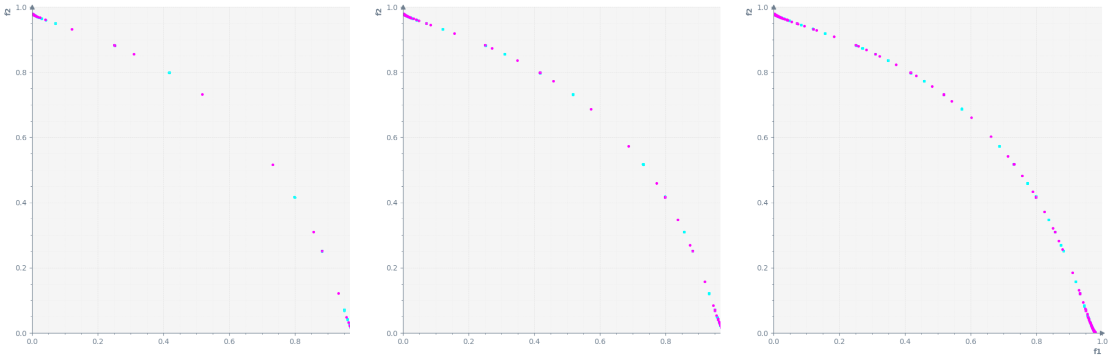
FIGURE 5 Iteration 5

4.2. **Results from Test Problems.**

This section presents results for multi-objective test problems from the literature. We tested the algorithm on test problems with varying characteristics, including the number of objective



Iteration 1 Iteration 2 Iteration 3



Iteration 4 Iteration 5 Iteration 6

FIGURE 6 Points found in each iteration for Test Problem 4.1 (n=3)

functions, the dimension of the preimage space, and with and without a convex constraint.

Test Problem 4.1. *This test problem is based on* [10] *where* $n \in \mathbb{N}$ *can be arbitrarily chosen.*

$$f(x) = \begin{pmatrix} 1 - e^{-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2} \\ \\ 1 - e^{-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2} \end{pmatrix} \quad \text{where} \quad x_i \in [-2,2] \text{ for all } i = 1,2,\ldots,n.$$

The parameters used in the algorithm are as follows: $\varepsilon = 0.05$, $\alpha \in 0,\ldots,38$, $c = (1.0,1.0)$, $w = (37,38)$ where $y_0^1 = (0.00,0.98)$, $y_0^2 = (0.98,0.00)$ and $r_0^0 = (0.0,0.0)$.

Figure 6 presents the points found in each iteration, in the space for n = 3. Cyan points represent the elements of set G_k (i.e. points found by CS), whereas magenta points represent the elements of set T_k (i.e. points found by PS) at iteration k.

Table 5 displays the number of points found in each iteration of the algorithm in the space for n = 3. The first column, k, represents the iteration number. The second column illustrates the total number of weakly minimal, minimal, and properly minimal points obtained by CS and PS at iteration k, while the third column shows the total number of weakly minimal, minimal, and properly minimal points up to iteration k. The last column presents the total number of approximate weakly minimal points obtained thus far.

k	$\lvert G_k \cup T_k \rvert$	$\lvert E_{k+1} \rvert$	$\lvert V_k \rvert$
1	19	19	0
2	38	50	3
3	131	172	22
4	269	378	83
5	499	740	220
6	860	1347	488

TABLE 5 Number of Points for Test Problem 4.1 (n=3)

k	$\lvert L \rvert$
41	34
359	210
3055	1268
20966	7644

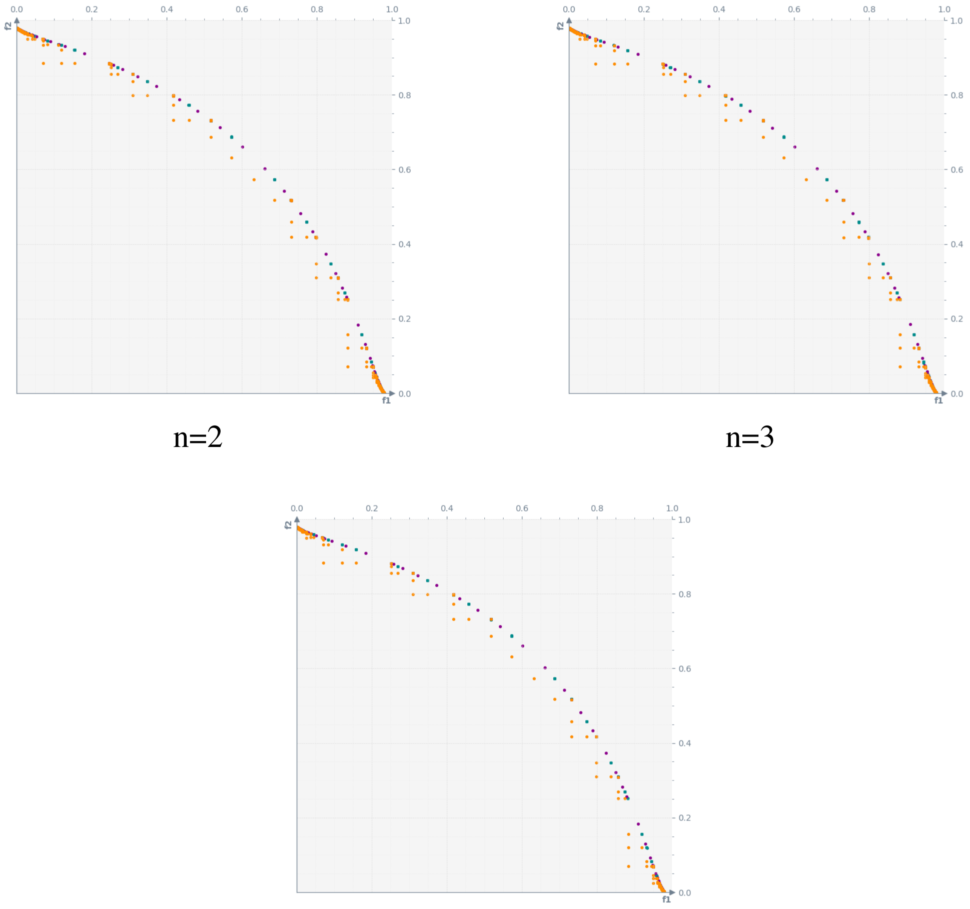
TABLE 6 Results from [27] (n=3)

n	k	$\lvert E \rvert$	$\lvert V \rvert$
2	6	1311	552
4	6	1414	452

TABLE 7 Number of Points for Test Problem 4.1 for n=2 and n=4

In [27], the authors proposed an algorithm for finding an approximation of the set of globally optimal solutions for nonconvex multi-objective optimization problems within a predefined quality in finite time. They also tested the algorithm with Test Problem 4.1. In Table 6, we provide the results for n=3 from this study, where k is the number of iterations and $|L|$ denotes the number of points found (for illustration of their results see Fig. 7(b) in [27]). However, please note that a direct comparison cannot be made. This is primarily because our method differs from the one presented in [27] (and the majority of the other methods in the literature) as our approach involves finding a set of optimal points based on a specific weight vector and reference points. In that sense, the method proposed in this paper fills an important gap in nonconvex multi-objective optimization.

We further investigate the test problem for n = 2 and n = 4 using the same parameters. The total number of iterations (k), the total number of weakly minimal, minimal, and properly minimal points ($|E|$), and the total number of approximate weakly minimal points ($|V|$) found are presented in Table 7. The resulting points are given in Figure 7. The dark cyan points represent the weakly minimal, minimal, and properly minimal points obtained by the CS method, while the dark magenta points represent those obtained by the PS method. Orange points represent the approximate weakly minimal points.



n=2

n=3

n=4

FIGURE 7 Resulting minimal points for Test Problem 4.1

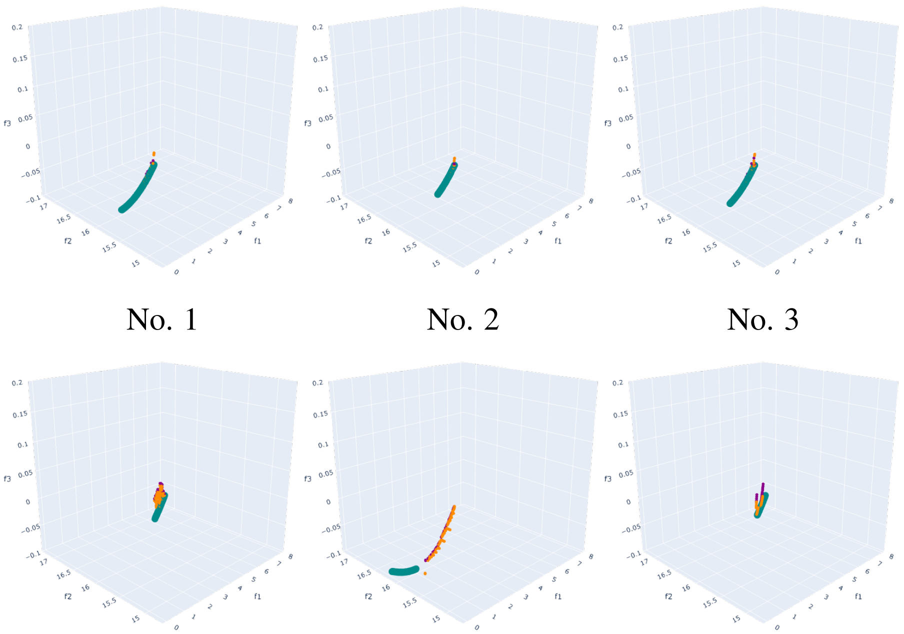
Test Problem 4.2. *This test problem is introduced in* [32]. *There is another version of this problem where a constraint is introduced in* [27]. *We will refer to the original form as "Test Problem 4.2.a" and the constrained version as "Test Problem 4.2.b", and Test Problem 4.2.a is as follows,*

$$f(x) = \begin{pmatrix} 0.5(x_1^2+x_2^2)+\sin(x_1^2+x_2^2) \\ \frac{(3x_1-2x_2+4)^2}{8}+\frac{(x_1-x_2+1)^2}{27}+15 \\ \frac{1}{x_1^2+x_2^2+1}-1.1\exp(-x_1^2-x_2^2) \end{pmatrix} \quad \text{where} \quad x_1,x_2 \in [-3,3].$$

where Test Problem 4.2.b considers the additional convex constraint given below.

No.	$w=(w_1,w_2,w_3)$	ε
1	(60, 40, 40)	0.005
2	(40, 60, 40)	0.005
3	(40, 40, 60)	0.005
4	(21, 105, 21)	0.005
5	(105, 21, 21)	0.02
6	(21, 105, 21)	0.02

TABLE 8 Parameters used for solving Test Problem 4.2.a



No. 1 No. 2 No. 3

No. 4 No. 5 No. 6

FIGURE 8 Results for Test Problem 4.2.a

$$g(\mathbf{x})=x_1^2+x_2^2-4.$$

We tested our algorithm with different weight vectors, w, and ε values. The w and ε values used for solving Test Problem 4.2.a and Test Problem 4.2.b are given in Table 8 and Table 9, respectively. The other parameters used in each setting are:$c=(5.0,1.0,1.0),(y_0^1,y_0^2,y_0^3)=((0.00,17.03,-0.1),(11.54,15.00,0.16),(0.00,17.04,-0.1))$ and $r_0^0=(0.00,15.00,-0.10)$.

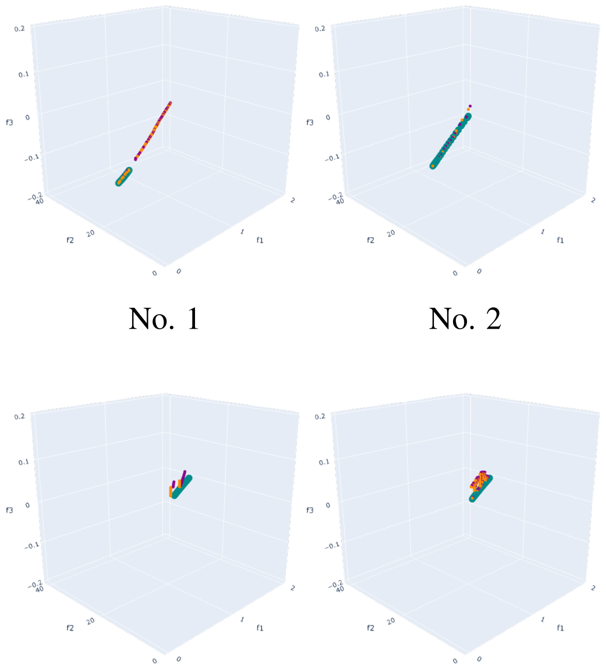
The results for different w and ε settings are given in Figure 8 and Figure 9 for the two test problems.[1]. Note that colors are used in the same fashion as in Test Problem 4.1. The dark cyan points represent the weakly minimal, minimal, and properly minimal points obtained by the CS method, while the dark magenta points represent those obtained by the PS method. Orange points represent the approximate weakly minimal points.

---

[1] An interactive version of the images can be accessed through the link

$No.$	$w = (w_1, w_2, w_3)$	ε
1	(35, 7, 7)	0.02
2	(21, 21, 105)	0.02
3	(21, 105, 21)	0.02
4	(21, 105, 21)	0.005

TABLE 9 Parameters used for solving Test Problem 4.2.b



No. 1 No. 2

No. 3 No. 4

FIGURE 9 Results for Test Problem 4.2.b

As mentioned above, Test Problem 4.2.b was introduced by [27]. For illustration of the results obtained in this study, we refer to Fig. 10(d) in [27].

The results obtained using the method we propose, from solving test problems with varying characteristics and parameter settings, demonstrate that the algorithm successfully finds weakly minimal, minimal, properly minimal, and approximate weakly minimal points, regardless of the size of the objective space and preimage space.

5. CONCLUSION

In this paper, we present a method for computing the set of optimal points for the given multi-objective optimization problem, corresponding to the given set of weights w, without convexity assumption. The nonconvexity implies that such a problem may have more than one minimal point corresponding to the same set of weights. Assuming that a reference point r and a grid of augmentation parameters $L = \{\alpha_1, \ldots, \alpha_P\}$ are given, we introduce a special way for iteratively generating a subset R_k of "reference points", and compute the corresponding set of minimal points for a given triple consisting of a weight vector w, an augmentation parameter $\alpha \in L$ and a reference point $r \in R_k$, by using the scalarizing function of the conic scalarization method. At every iteration, besides computing a possible set of minimal and properly minimal points by using the scalarizing function of the conic scalarization method, we also use the scalarizing function of the Pascoletti-Serafini method, and for every set of the newly generated set of reference points in R_k (at the current iteration k), we compute the corresponding weakly minimal points of the problem. By using the sets of minimal, properly minimal, and weakly minimal points, we construct a sequence of sets Y_k, which converges to the set of weakly minimal points of $(Y + R^n_+)$, where $Y = f(X)$.

The conic scalarization method is guaranteed to find properly efficient and efficient points, unlike the Pascoletti-Serafini method. The main reason behind integrating the Pascoletti-Serafini method into the algorithm is to increase the exploration efficiency of the conic scalarization method. In this context, the Pascoletti-Serafini method plays a supporting role by facilitating the quicker narrowing of the search space and accelerating the ability of conic scalarization to identify new points. The results obtained from solving the test problems show that the method presented in this paper effectively computes the set of minimal and properly minimal points on one side, and approximate weakly minimal points on the other.

REFERENCES

[1] F. Amir, A. Farajzadeh, J. Alzabut, An improved proximal method with quasi-distance for nonconvex multi-objective optimization problem, J. Appl. Anal. 28 (2022), 333–340.

[2] P. Armand, Finding all maximal efficient faces in multi-objective linear programming, Math. Program. 61 (1993), 357–375.

[3] H. P. Benson, An improved definition of proper efficiency for vector maximization with respect to cones, J. Math. Anal. Appl. 71 (1979), 232–241.

[4] H. P. Benson, An outer approximation algorithm for generating all efficient extreme points in the outcome set of a multiple objective linear programming problem, J. Global Optim. 13 (1998), 1–24.

[5] G. Eichfelder, Scalarizations for adaptively solving multi-objective optimization problems, Comput. Optim. Appl. 44 (2009), 249–273.

[6] G. Eichfelder, L. Warnow, An approximation algorithm for multi-objective optimization problems using a box-coverage, J. Glob. Optim. 83 (2022), 329—357.

[7] G. Eichfelder, O. Stein, L. Warnow, A solver for multi-objective mixed-integer convex and nonconvex optimization, J. Optim. Theory Appl. 203 (2024), 1736-1766.

[8] A. Ed-dahdah, M. Laghdir, M. Mabrouk, A. Ammar, Approximate weak and proper subdifferentials of the difference of two vector convex mappings and applications, J. Appl. Numer. Optim. 7 (2025), 69-83.

[9] C. Fan, G. Yu, S. Hua, Optimality and scalarization of approximate solutions for vector equilibrium problems via Michel-Penot subdifferential, Appl. Set-Valued Anal. Optim. 6 (2024), 233-244.

[10] C.M. Fonseca, P.J. Fleming, multi-objective genetic algorithms made easy: selection sharing and mating restriction, In: First International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications, IET, pp. 45–52, 1995.

[11] R.N. Gasimov, Characterization of the Benson Proper Efficiency and Scalarization in Nonconvex Vector Optimization, In: M. Koksalan, S. Zionts (eds.) Multiple Criteria Decision Making in the New Millennium, Book Series: Lecture Notes in Econom. and Math. Systems, vol. 507, pp. 189–198, 2001.

[12] R. N. Gasimov, Augmented Lagrangian duality and nondifferentiable optimization methods in nonconvex programming, J. Global Optim. 24 (2002), 187–203.

[13] D. Gourion, D.T. Luc, Generating the weakly efficient set of nonconvex multi-objective problems, J. Global Optim. 41 (2008), 517–538.

[14] M.L.N. Gonçalves, F. S. Lima, L.F. Prudente, Globally convergent Newton-type methods for multi-objective optimization, Comput. Optim. Appl. 83 (2022), 403–434.

[15] M.I. Henig, Proper efficiency with respect to cones, J. Optim. Theory Appl. 36 (1982), 387–407.

[16] N. Hoseini Monjezi, S. Nobakhtian, A dynamic multiple proximal bundle algorithm for locally Lipschitz multi-objective optimization problems, Optimization, 74 (2025), 3419–3443.

[17] I.C. Kaya, F. Ulus, An iterative vertex enumeration method for objective space based vector optimization algoritihm, RAIRO - Oper. Res. 55 (2021), 2471–2485.

[18] R. Kasimbeyli, O. Ustun, A. Rubinov, The modified subgradient algorithm based on feasible values, Optimization 5 (2009), 535–560.

[19] R. Kasimbeyli, A nonlinear cone separation theorem and scalarization in nonconvex vector optimization, SIAM J. Optim. 20 (2010), 1591–1619.

[20] R. Kasimbeyli, A conic scalarization method in multi-objective optimization, J. Global Optim. 56 (2013), 279-297.

[21] R. Kasimbeyli, Z.K. Ozturk, N. Kasimbeyli, G.D. Yalcin, B. Icmen, Conic scalarization method in multi-objective optimization and relations with other scalarization methods, Proceedings of the 3rd International Conference on Modelling, Computation and Optimization in Information Systems and Management Sciences-MCO 2015-Part I, In: L.T.H. An, P.D. Tao, N. N. Thanh (Eds.), Volume of Series: Advances in Intelligent Systems and Computing, Springer, Vol. 359, pp. 319–329, 2015.

[22] R. Kasimbeyli, Z.K. Ozturk, N. Kasimbeyli, G.D. Yalcin, B. Icmen, Comparison of some scalarization methods in multi-objective optimization, Bull. Malaysian Math. Sci. Soc. 42 (2019), 1875–1905.

[23] N.T.B. Kim, D.T. Luc, Normal cones to a polyhedral convex set and generating efficient faces in linear multi-objective programming, Acta Math. Vietnam. 25 (2000), 101–124.

[24] K. Klamroth, J. Tind, M.M. Wiecek, Unbiased approximation in multicriteria optimization, Math. Meth. Oper. Res. 56 (2002), 413–457.

[25] M. Link, S. Volkwein, Adaptive piecewise linear relaxations for enclosure computations for nonconvex multi-objective mixed-integer quadratically constrained programs, J. Global Optim. 87 (2023), 97–132.

[26] K. Miettinen, Nonlinear Multi-objective Optimization, Kluwer, Boston, 1999.

[27] J. Niebling, G. Eichfelder, A branch–and–bound-based algorithm for nonconvex multi-objective optimization, SIAM J. Optim. 29 (2019), 794–821.

[28] A. Pascoletti, P. Serafini, Scalarizing vector optimization problems, J. Optim. Theory Appl. 42 (1984), 499–524.

[29] S. Ruzika, M.M.Wiecek, Approximation methods in multi-objective programming, J. Optim. Theory Appl. 126 (2005), 473–501.

[30] C. Tammer, P. Weidner, Nonconvex separation theorems and some applications in vector optimization, J. Optim. Theory Appl. 67 (1990), 297–320.

[31] F. Ulus, Approximation algorithm variants for convex multi-objective optimization problems, GU J. Sci. Part C. 8 (2020), 224–236.

[32] R. Viennet, C. Fonteix, M. Ivan, Multicriteria optimization using a genetic algorithm for determining a Pareto set, Int. J. Sys. Sci. 27 (1996), 255–260.

[33] W. Wu, X. Yang, Reference-point-based branch and bound algorithm for multi-objective optimization, J. Global Optim. 88 (2024), 927–945.

[34] X. Zhao, L. O. Jolaoso, Y. Shehu, J. C. Yao, Convergence of a nonmonotone projected gradient method for nonconvex multi-objective optimization, J. Nonlinear Var. Anal. 5 (2021), 441–457.