Journal of Mathematical Modeling
Vol. -, No. -, 2026, pp. -. Research Article

JMM

An accelerated method for solving constrained multi-objective optimization

Niloofar Salehi Mokari[†], **Hadi Basirzadeh**[†*], **Vahid Morovati**[§]

[†] *Department of Mathematics, Faculty of Mathematical Sciences and Computer, Shahid Chamran University of Ahvaz, Ahvaz, Iran*
[§] *Department of Mathematics, University of Hormozgan, Bandarabbas, Iran*
Email(s): niloofarsalehi74@gmail.com, basirzad@scu.ac.ir, vahidmath99@gmail.com

Abstract. A novel non-parametric algorithm is introduced for solving constrained multi-objective optimization problems. At each iteration, a convex subproblem is solved to determine the search direction, while a non-monotone line search technique is used to determine the step size. An adaptive acceleration term, computed from changes in the search directions, is incorporated to scale the step and dynamically enhance convergence performance. The algorithm's effectiveness relies on a diverse set of initial feasible solutions to accurately approximate the non-dominated boundary. Benchmark tests validate the approach, with Pareto fronts compared to those obtained using the Zoutendijk method. Numerical evaluations demonstrate superior performance in terms of convergence rate and solution quality. The algorithm is also applied to a real-world engineering design problem involving speed reduction, highlighting its computational efficiency and robustness in practical applications.

Keywords: Constrained multi-objective optimization problems, feasible direction methods, line search techniques, pareto critical point.
AMS Subject Classification 2010: 90C29, 90C30, 90C52, 49M37.

1 Introduction

Multi-objective optimization problems (MOOPs) involve optimizing multiple conflicting objectives, which presents a challenge in fields such as statistics [11], environmental analysis [33], machine learning [43,46,49], and management science [7]. Simultaneous minimization of all objectives is often infeasible due to their non-comparable and contradictory nature. Unlike single-objective optimization problems (SOOPs), which yield unique or alternative optima, MOOPs produce pareto optimal solutions, where improving one objective may result in the compromise of another [51].

[*] Corresponding author

Received: 20 May 2025/ Revised: 04 October 2025/ Accepted: 03 November 2025
DOI: 10.22124/jmm.2025.30721.2753

Let us consider a problem of constrained multi-objective optimization problems (CMOOPs):

$$\begin{aligned} &\min_{x\in S} C(x) = (C_1(x), C_2(x), \dots, C_m(x))^T \\ &\text{s.t. } G_j(x) \leq 0, j = 1, \dots, p. \end{aligned} \tag{1}$$

Indicate $I_p = \{1, \dots, p\}$, $I_m = \{1, \dots, m\}$ and $\mathbb{R}^n_+ = \{x \in \mathbb{R}^n | x_i \geq 0, i = 1, \dots, n\}$. The feasible space of CMOOPs is delineated by $S = \{x \in \mathbb{R}^n : G_j(x) \leq 0, j \in I_p\}$. Here, $C_r, G_j : \mathbb{R}^n \to \mathbb{R}$ are continuously differentiable functions for $r \in I_m$ and $j \in I_p$.

Solving MOOPs often involves scalarization methods [22,34] or heuristic approaches [39,45], which generate efficient solutions by adjusting parameters. However, unclear parameter selection, a significant limitation, complicates even simple problems and may yield infinite solutions, heavily relying on the decision maker's judgment. The weighted sum approach, a widely used scalarization technique, minimizes a linear combination of MOOP objectives using parameters defined by the decision maker to generate efficient solutions. However, selecting appropriate parameters is a challenging task, particularly for complex problems. In contrast, heuristic approaches often lack guaranteed convergence while approximating the pareto front. Recent single-objective techniques, which employ parameter-free line search methods [3,24,25,37,44], have gained attention for their effectiveness in addressing MOOPs.

Sequential quadratic programming (SQP), a widely used technique for constrained optimization, has been adapted by Fliege and Vaz for CMOOPs [26]. Their method uses linear approximations for constraints and quadratic approximations for objectives, solving a subproblem to effectively address complex problems. Morovati et al. extended the Zoutendijk method, a nonparametric approach that uses quadratic subproblem directions, to solve CMOOPs iteratively [38]. Comparative studies show that it outperforms previous methods, particularly parametric approaches, in solving CMOOPs.

Sequential quadratically constrained quadratic programming (SQCQP), introduced by Pan [42], uses quadratic approximations for both objectives and constraints. It was later enhanced by Anitescu [2] to achieve superlinear convergence using the Mangasarian-Fromovitz condition. For convex programming, Fukushima et al. [27] proposed a globally convergent method based on the Slater condition, although modifications by [47] are limited to SOOPs.

Line search techniques are essential for MOOPs. The Armijo method ensures monotonicity but requires numerous inequalities, which leads to smaller steps and slower convergence compared to SOOPs. In contrast, nonmonotone line search approaches [29,35,36,52], adopted in this study, allow for function expansion, enabling larger steps and enhancing convergence rates by flexibly satisfying objective-related inequalities.

Acceleration techniques, crucial in optimization fields such as machinery and imaging, were significantly advanced by Nesterov's seminal work [8, 40], primarily for SOOPs. While their application to MOOPs was initially limited, El Moudden et al. [21] adapted an extended Nesterov method in 2020 to develop an expedited diagonal gradient approach for MOOPs. Building on this foundation, Wang and Tana [48, 50] introduced innovative numerical methods in 2023, leveraging accelerated computing to enhance solutions for MOOPs.

Motivated by the challenges associated with parametric methods in CMOOPs, this study develops a non-parametric algorithm that introduces a novel quadratic modeling framework. The algorithm is further enhanced by a nonmonotone line search and an acceleration strategy to effectively address complex CMOOPs. The method's robustness is demonstrated through rigorous convergence analysis, providing a versatile approach for various applications, such as speed reduction design.

This paper is structured to present the proposed approach to CMOOPs systematically. Section 2 defines fundamental concepts of CMOOPs and introduces the proposed algorithm, which features a novel quadratic model and nonmonotone line search for optimized step sizing. Section 3 rigorously examines the algorithm's convergence properties, providing theoretical insights to validate its robustness. Section 4 evaluates the algorithm through numerical experiments, comparing its performance with advanced methods to highlight its contributions to solving CMOOPs.

2 Notations

This section introduces key definitions relevant to the research presented. The primary objective of this paper is to propose a solution for CMOOPs, denoted by

$$\begin{aligned} &\min_{x\in U} C(x) \\ &\text{s.t. } G(x) \leq 0. \end{aligned} \tag{CMO}$$

Let U be a subset of $\mathbb{R}^n$, where n is a natural number. Consider vector functions $C : U \to \mathbb{R}^m$ and $G : U \to \mathbb{R}^p$, with m and p also natural numbers. The Euclidean distance in $\mathbb{R}^n$ is denoted by the norm $\|\cdot\|$. Assume that $C = (C_1,\ldots,C_m)$ and $G = (G_1,\ldots,G_p)$ are twice continuously differentiable. The feasible region S is defined as

$$S = \{x \in U \mid G_j(x) \leq 0,\ \forall j \in I_p\}. \tag{2}$$

Each function G_j must be non-positive for a point x to belong to S, a condition that is crucial for analyzing solutions to CMOOPs.

Definition 1. ([34]) An efficient solution (also known as a pareto optimal solution) for MOOPs satisfies

$$C_r(x) \leq C_r(x^*),\ \forall r \in I_m$$

and there exists at least one objective $r \in I_m$ such that

$$C_r(x) \neq C_r(x^*).$$

This definition indicates that an efficient solution cannot be improved in any objective without deteriorating another, thereby establishing pareto optimality for MOOPs.

Definition 2. ([34]) A solution $x^* \in U$ is a weakly efficient solution for MOOPs if no other solution $y \in U$ exists such that $C(y) > C(x^*)$, or equivalently, $C_r(y) < C_r(x^*)$ does not hold for every $r \in I_m$.

Weakly efficient solutions cannot be improved in all objectives simultaneously but may allow for improvement in some objectives without worsening others, offering greater flexibility than pareto optimal solutions.

Definition 3. ([24]) If $Z \subseteq U$ and x^* is optimal in Z, then $x^* \in U$ is a local optimal point (or local weak optimal point). If x^* is a local efficient (or local weak efficient) point, then $C(x^*)$ is a local non-dominated (or local weak non-dominated) point.

Satisfying necessary conditions is critical for obtaining optimal solutions and advancing theoretical research in this field.

Theorem 1. ([34]) Given $x^* \in S$, with $C_r, r \in I_m$, and $G_j, j \in I_p$ continuously differentiable, if x^* is a weakly efficient solution of MOOPs, the vectors γ and β exist such that

$$\sum_{r \in I_m} \gamma_r \nabla C_r(x^*) + \sum_{j \in I_p} \beta_j \nabla G_j(x^*) = 0, \tag{3}$$

$$\beta_j G_j(x^*) = 0, \quad \forall j \in I_p, \tag{4}$$

$$0 \leq \gamma \in \mathbb{R}^m, \quad 0 \leq \beta \in \mathbb{R}^p, \quad (\gamma, \beta) \neq 0^{m+p}. \tag{5}$$

The Fritz John condition alone does not guarantee that $\gamma_r > 0$ for some $r \in I_m$, and thus, constraint qualifications are required to verify $\gamma_r \neq 0$. Maciel in [32] examines various constraint qualifications. This paper assumes the Slater constraint qualification, which states that there exists a feasible point $\bar{x}$ such that $G_j(\bar{x}) < 0$, $\forall j \in I_p$. Under this condition, the Fritz John multipliers γ and β are unique, with $\gamma_r > 0$ for some $r \in I_m$, which allows for stronger results regarding optimal solutions. The Slater conditions not strictly necessary for Karush-Kuhn-Tucker (KKT) conditions; weaker qualifications, such as the Mangasarian-Fromovitz constraint qualification (MFCQ), may suffice and are considered in this study.

Definition 4. ([41]) The Mangasarian-Fromovitz constraint qualification holds at $x \in S$ if a vector $q \in \mathbb{R}^n$ exists for problem CMO such that

$$\nabla G_j(x)^T q < 0, \ \forall j \in I_p. \tag{6}$$

This criterion ensures a consistent optimization framework, facilitating KKT conditions. Verifying the MFCQ allows for practical analysis of optimal solutions and their associated multipliers.

Definition 5. ([38]) A point $x \in S$ is a critical point of problem CMO if, for every $q \in \mathbb{R}^n$ satisfying

$$\nabla G_j(x)^T q \leq 0, \ \forall j \in I_p \tag{7}$$

there exists at least one $r \in I_m$ such that

$$\nabla C_r(x)^T q \geq 0. \tag{8}$$

This definition ensures that critical points satisfy the optimality conditions derived from the Fritz John conditions.

With the definitions and theorems established, this section explores the determination of the search direction by solving a quadratic problem for CMOOPs. A directional vector q for a feasible point x_k is obtained by analyzing

$$\begin{aligned} &\min_{q \in \mathbb{R}^n} \max_{r \in I_m} \nabla C_r(x)^T q + \frac{1}{2} \|q\|^2 \\ &\text{s.t. } G_j(x) + \nabla G_j(x)^T q \leq 0, \ j \in I_p. \end{aligned} \tag{CQMO}$$

This subproblem CQMO is equivalent to

$$\min O(\vartheta, q) = \vartheta$$

$$\begin{cases} s.t. \nabla C_r(x)^T q + \frac{1}{2} \|q\|^2 \leq \vartheta, \ r \in I_m, \\ \quad G_j(x) + \nabla G_j(x)^T q \leq \eta \vartheta, \ j \in I_p. \end{cases} \tag{9}$$

Solving this subproblem yields the directional vector q, which indicates movement from x_k toward an optimal solution.

This constrained problem derived from Chen-Kasterova's formulation [12], transforming the objectives of subproblem CQMO into constraints with an upper bound denoted as ϑ, which is minimized in the objective function. A small coefficient applied to ϑ distinguishes this model from previous ones, which set the constraint right-hand sides to zero. When $\eta \neq 0$, the constraints approach zero, effectively disregarding zero values and allowing for more nuanced constraint management.

Assume x^* is a locally optimal solution for (9). Lagrange multiplier vectors $\gamma \in \mathbb{R}^m$ and $\beta \in \mathbb{R}^p$ satisfy

$$L(\vartheta, q; \gamma, \beta) = \vartheta - \sum_{r\in I_m} \gamma_r \left(\nabla C_r(x)^T q + \frac{1}{2} \|q\|^2 - \vartheta \right) - \sum_{j\in I_p} \beta_j \left(G_j(x) + \nabla G_j(x)^T q - \eta\vartheta \right). \tag{10}$$

Given MFCQ at any point (ϑ, q), there exist $0 \leq \gamma \in \mathbb{R}^m$, $0 \leq \beta \in \mathbb{R}^p$, with $\gamma \neq 0$, $\beta \neq 0$, fulfilling KKT conditions

$$\sum_{r\in I_m} \gamma_r + \eta \sum_{j\in I_p} \beta_j = 1, \tag{11}$$

$$\sum_{r\in I_m} \gamma_r \nabla C_r(x) + q + \sum_{j\in I_p} \beta_j \nabla G_j(x) = 0, \tag{12}$$

$$\gamma_r \geq 0,\ \gamma_r \left(\nabla C_r(x)^T q + \frac{1}{2} \|q\|^2 - \vartheta \right) = 0,\ r \in I_m, \tag{13}$$

$$\beta_j \geq 0,\ \beta_j \left(G_j(x) + \nabla G_j(x)^T q - \eta\vartheta \right) = 0,\ j \in I_p, \tag{14}$$

$$\nabla C_r(x)^T q + \frac{1}{2} \|q\|^2 - \vartheta \leq 0,\ r \in I_m, \tag{15}$$

$$G_j(x) + \nabla G_j(x)^T q - \eta\vartheta \leq 0,\ j \in I_p. \tag{16}$$

From (12), a descent direction is obtained as

$$q(x) = -\left(\sum_{r\in I_m} \gamma_r \nabla C_r(x) + \sum_{j\in I_p} \beta_j \nabla G_j(x) \right), \tag{17}$$

where $\gamma(x) = (\gamma_1(x), \ldots, \gamma_m(x))$ and $\beta(x) = (\beta_1(x), \ldots, \beta_p(x))$. Substituting (17) into (10) leads to a constrained dual problem

$$\begin{aligned} \max_{\gamma,\beta} & -\frac{1}{2} \left\| \sum_{r\in I_m} \gamma_r \nabla C_r(x) + \sum_{j\in I_p} \beta_j \nabla G_j(x) \right\|^2 + \sum_{j\in I_p} \beta_j G_j(x) \\ \text{s.t.} & \sum_{r\in I_m} \gamma_r + \eta \sum_{j\in I_p} \beta_j = 1, \\ & \gamma_r \geq 0,\ \beta_j \geq 0,\ r \in I_m,\ j \in I_p. \end{aligned} \tag{18}$$

Assuming strong duality, (17) yields

$$-\frac{1}{2} \left\| \sum_{r\in I_m} \gamma_r \nabla C_r(x) + \sum_{j\in I_p} \beta_j \nabla G_j(x) \right\|^2 = -\frac{1}{2} \|q(x)\|^2. \tag{19}$$

Thus, (18) is equivalent to

$$\begin{aligned} &\max_{\gamma,\beta} \phi(x) = -\frac{1}{2}\|q\|^2 + \sum_{j\in I_p} \beta_j G_j(x) \\ &\text{s.t.} \sum_{r\in I_m} \gamma_r + \eta \sum_{j\in I_p} \beta_j = 1, \\ &\quad \gamma_r \geq 0,\ \beta_j \geq 0,\ r \in I_m,\ j \in I_p. \end{aligned} \tag{20}$$

The following lemma is a prerequisite for the existence of a critical point. Subsequent lemmas generalize constrained SOOP results to CMOOPs.

Lemma 1. A feasible point x^* of problem CMO is critical if it satisfies the Fritz John conditions and the MFCQ.

Proof. Assuming x^* satisfies the Fritz John conditions, vectors $0 \leq \gamma \in \mathbb{R}^m$, $0 \leq \beta \in \mathbb{R}^p$, with $\gamma \neq 0$, $\beta \neq 0$, exist such that for $q \neq 0$

$$\sum_{r\in I_m} \gamma_r \nabla C_r(x)^T q + \sum_{j\in I_p} \beta_j \nabla G_j(x)^T q = 0. \tag{21}$$

To confirm criticality, no $q \in \mathbb{R}^n$ should satisfy $\nabla C_r(x^*)^T q < 0$, $\forall r \in I_m$ and $\nabla G_j(x^*)^T q < 0$, $\forall j \in I_p$. If such a $q^* \in \mathbb{R}^n$ exists

$$\begin{aligned} &\nabla C_r(x^*)^T q^* < 0,\ \forall r \in I_m \Rightarrow \sum_{r\in I_m} \gamma_r \nabla C_r(x^*)^T q^* < 0, \text{ as } \gamma_r \geq 0, \\ &\nabla G_j(x^*)^T q^* < 0,\ \forall j \in I_p \Rightarrow \sum_{j\in I_p} \beta_j \nabla G_j(x^*)^T q^* < 0, \text{ as } \beta_j \geq 0. \end{aligned} \tag{22}$$

Since x^* satisfies MFCQ, it follows that

$$\sum_{r\in I_m} \gamma_r \nabla C_r(x^*)^T q^* + \sum_{j\in I_p} \beta_j \nabla G_j(x^*)^T q^* < 0. \tag{23}$$

This contradicts (21), confirming that x^* is a critical point. □

Lemma 2. Given $x \in S$ satisfying MFCQ, if (ϑ, q) is the algorithm's solution and $q = 0$, then x is a critical point of problem CMO.

Proof. If $(\vartheta, 0)$ is the solution, substituting $q = 0$ into equations (11) - (16) yields

$$\begin{aligned} &\sum_{r\in I_m} \gamma_r \nabla C_r(x) + \sum_{j\in I_p} \beta_j \nabla G_j(x) = 0, \\ &1 - \sum_{r\in I_m} \gamma_r - \eta \sum_{j\in I_p} \beta_j = 0, \end{aligned} \tag{24}$$

$$\begin{aligned} &-\vartheta \leq 0,\ r \in I_m, \\ &G_j(x) - \eta\vartheta \leq 0,\ j \in I_p, \\ &\gamma_r \geq 0,\ \gamma_r \vartheta = 0,\ r \in I_m, \end{aligned} \tag{25}$$

$$\beta_j \geq 0,\ \beta_j(G_j(x) - \eta\vartheta) = 0,\ j \in I_p. \tag{26}$$

Since x is feasible, $G_j(x) \leq 0$, $\forall j \in I_p$. Thus, from equation (26), we have $\beta_j \geq 0$ and $\beta_j(G_j(x) - \eta\vartheta) = 0$. Combined with equation (24), these conditions satisfy the Fritz John conditions. Since the MFCQ holds at x, Lemma 1 implies that x is a critical point of problem CMO. □

2.1 Computing step-size using nonmonotone line search technique

Accurately determining the search direction and line search conditions is crucial for optimization problems. This subsection explores a nonmonotone line search approach for MOOPs, derived from Ahookhosh et al.'s method for SOOPs [1]:

$$C(x_k+\alpha_k q_k) \leq P_k + \rho \alpha_k \phi(x_k), \tag{27}$$

where

$$P_k = \begin{cases} C(x_0), & \text{if } k = 0, \\ (1-\theta)C(x_k) + \theta P_{k-1}, & \text{if } k \geq 1, \end{cases} \tag{28}$$

with $\rho, \theta \in (0,1)$. This nonmonotone line search replaces the right-hand side of Armijo technique with a convex combination of $C(x_k)$ and P_k, incorporating prior iteration values. When $\theta = 0$, (27) reduces to the Armijo technique. For MOOPs, the approach is given by

$$C_r(x_k+\alpha_k q_k) \leq P_k^l + \rho \alpha_k \phi(x_k), \quad \theta \in (0,1), \tag{29}$$

where

$$P_k^l = \begin{cases} C_r(x_0), & \text{if } k = 0, \\ (1-\theta)C_r(x_k) + \theta P_{k-1}^l, & \text{if } k \geq 1, \end{cases} \tag{30}$$

with $P_0 = C(x_0)$. This nonmonotone line search demonstrates improved convergence over monotone techniques, such as Armijo, by combining current and previous objective function values. This results in more suitable step sizes for directions from (20).

2.2 Accelerated parameter

The article by Andrei investigates the limitations of the gradient descent method, particularly how it can impede the convergence rate. To address this, the paper proposes the introduction of an acceleration parameter, which has been shown to significantly enhance convergence speed and overall efficiency. Building on the successful application of this parameter in gradient reduction methods for SOOPs, we have integrated it into unconstrained MOOPs. . Furthermore, we found that incorporating this parameter into constrained problems offers substantial improvement in algorithmic efficiency. Based on the findings presented in that paper, we have extended our study to examine the effects of this parameter on CMOOPs. Thus, the technique outlined in this paper incorporates the acceleration parameter into the sequence generated by the current approach. Further discussion is provided below.

Algorithm 1: ACMOP Algorithm

```
S.0: Choose $0 < \rho, \theta, L < 1$, $x_0 \in S$, $\varepsilon > 0$.
For $k = 0, 1, 2, \dots$

S.1: Solve sub-problem (18) at $x_k$ to obtain $q := q(x_k)$ by (17) and $\phi(x_k)$ by (20).

S.2: If $\phi(x_k) > -\varepsilon$, then $x_k$ is the optimal solution.

S.3: If $\phi(x_k) < -\varepsilon$, set $\alpha_k = 1$. If $G(x_k + \alpha_k q_k) > -||q_k||^{2.95}$, consider $\alpha_k = L\alpha_k$ until
     $G(x_k + \alpha_k q_k) \leq -||q_k||^{2.95}$.

S.4: Set $t_k = \alpha_k$ and proceed until the following nonmonotone line search technique holds:

     $$C_r(x_k + t_k q_k) \leq P_r(x_k) + \rho t_k T_k \phi(x_k), \quad \text{for } r \in I_m. \qquad (31)$$

     where $T_0 = 1$, $P_0 = C(x_0)$, and

     $$P_r = (1-\theta)C(x_k) + \theta P_{r-1}, \quad \text{consider } t_k = Lt_k. \qquad (32)$$

S.5: Then, $\alpha_k = t_k$. Solve sub-problem (18) at $z_1 = x_k + \alpha_k q_k$ to obtain $q_1 := q(z_1)$ by (17) and $\phi(z_1)$ by
     (20). Then, set $y_1 = q_k - q_1$, $s_1 = z_1 - x_k$.

S.6: Calculate Accelerated parameter:
     $T_k = max(0, \frac{\|q_k\|}{y_k^T q_k})$; $z_2 = x_k + T_k \alpha_k q_k$; $s_2 = z_2 - x_k$.
     Solve sub-problem (18) at $z_2$ to obtain $q_2 := q(z_2)$ by (17) and $\phi(z_2)$ by (20), and set $y_2 = q_k - q_2$.

     If $y_2^T s_2 > \varepsilon$,
         $x_{k+1} = z_2$; $q_{k+1} = q_2$; $\phi_{k+1} = \phi(z_2)$; $s_{k+1} = s_2$; $y_{k+1} = y_2$.
     Else;
         $x_{k+1} = z_1$; $q_{k+1} = q_1$; $\phi_{k+1} = \phi(z_1)$; $s_{k+1} = s_1$; $y_{k+1} = y_1$.

S.7: Update $k = k+1$ and proceed to S.2.
```

A flowchart of the algorithmic process is presented in Figure 1 to enhance clarity.

3 Convergence analysis of ACMOP Algorithm

Fliege and Vaz [26], along with Ansary and Panda [3–5], provided empirical evidence supporting the proposition that each accumulation point of the sequence x_k, generated by their algorithm, can be regarded as a pareto critical point. However, none of these studies have yet analyzed the convergence rate of their respective algorithms. Therefore, further research is necessary to determine the convergence rate, which would offer valuable insights into the effectiveness and efficiency of these algorithms.

Theorem 2. *Let (ϑ_k, q_k) denote the solution of the subproblem (20). Assume that the sequences $\{x_k\}$*

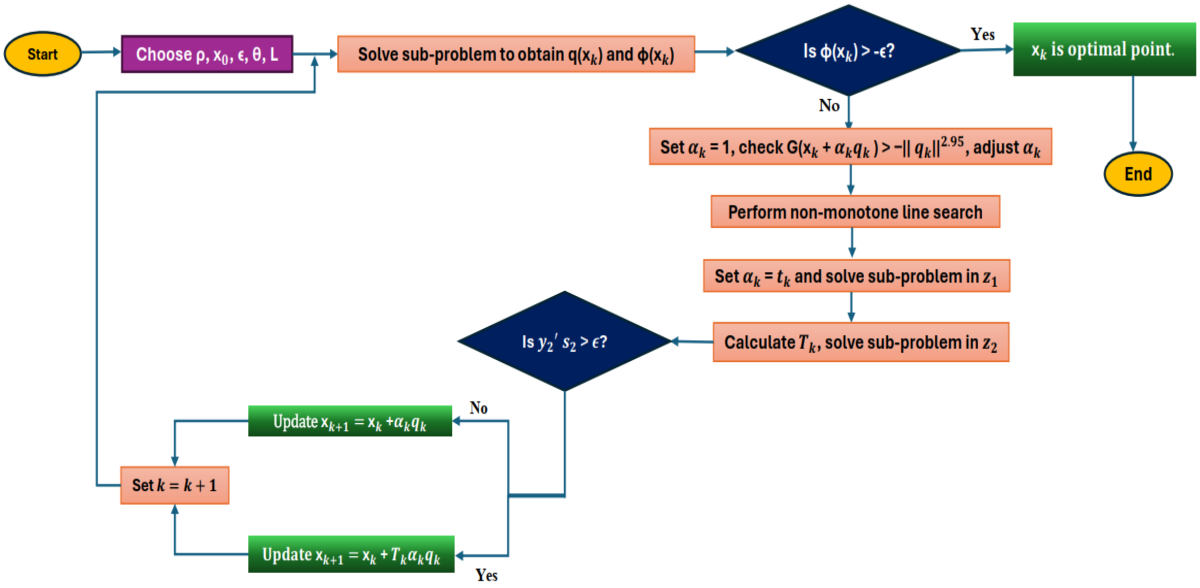

Start
Choose $p, x_0, \epsilon, \theta, L$
Solve sub-problem to obtain $q(x_k)$ and $\phi(x_k)$
Is $\phi(x_k) > -\epsilon$?
Yes
x_k is optimal point.
End
No
Set $\alpha_k = 1$, check $G(x_k + \alpha_k q_k) > -\|q_k\|^{2.95}$, adjust α_k
Perform non-monotone line search
Set $\alpha_k = t_k$ and solve sub-problem in z_1
Calculate T_k, solve sub-problem in z_2
Is $y_2' s_2 > \epsilon$?
No
Update $x_{k+1} = x_k + \alpha_k q_k$
Set $k = k+1$
Update $x_{k+1} = x_k + T_k \alpha_k q_k$
Yes

Figure 1: Flowchart of the ACMOP Algorithm

and $\{(\vartheta_k, q_k)\}$ *are bounded, and that* $x_k \to x^*$ *as* $k \to +\infty$*. Then the sequence* $\{(\vartheta_k, q_k)\}$ *converges to* (ϑ^*, q^*)*, where* (ϑ^*, q^*) *is the solution of (20) associated with* x^**. Furthermore, if* $q_k \to 0$ *and the MFCQ holds at each* x_k*, then* x^* *is a strongly critical point of the MOOP.*

Proof. Assume the sequence $\{x_k\}$ converges to x^*, but $(\vartheta_k,\ q_k)$ does not converge to $(\vartheta^*,\ q^*)$. Also, since (ϑ_k, q_k) is bounded, a subsequence $(\vartheta_k,\ q_k)$ for $k \in K$ exists that converges to $(\bar{\vartheta},\ \bar{q}) = (\vartheta^*,\ q^*)$.

Given that $(\vartheta_k,\ q_k)$ represents the optimal solution of the quadratic programming subproblem $QP(x_k)$, there exist (γ_k, β_k) such that $(\vartheta_k,\ q_k; \gamma_k,\ \beta_k)$ satisfies the KKT conditions (11) - (16). From (11), it follows that both γ_k and β_k are bounded. Therefore, a convergent subsequence of $(\gamma_k,\ \beta_k)$ exists for $k \in K$.

For simplicity, assume that as $k \to \infty$ and $k \in K$, $\gamma_k \to \gamma^*$ and $\beta_k \to \beta^*$. Consequently, taking the limit as $k \to \infty$ for $k \in K$ in equations (11) - (16), yields

$$\sum_{r\in I_m} \gamma_r^* + \eta \sum_{j\in I_p} \beta_j^* = 1, \tag{33}$$

$$\sum_{r\in I_m} \gamma_r^* \nabla C_r(x) + \bar{q} + \sum_{j\in I_p} \beta_j^* \nabla G_j(x) = 0, \tag{34}$$

$$\gamma_r^* \geq 0,\ \gamma_r^*(\nabla C_r(x)^T \bar{q} + \frac{1}{2}||\bar{q}||^2 - \vartheta^*) = 0,\ r \in I_m, \tag{35}$$

$$\beta_i^* \geq 0,\ \beta_j^*(G_j(x) + \nabla G_j(x)^T \bar{q} - \eta\vartheta^*) = 0,\ j \in I_p \tag{36}$$

$$\nabla C_r(x)^T \bar{q} + \frac{1}{2}||\bar{q}||^2 - \vartheta^* \leq 0,\ r \in I_m, \tag{37}$$

$$G_j(x) + \nabla G_j(x)^T \bar{q} - \eta\vartheta^* \leq 0,\ j \in I_p. \tag{38}$$

These conditions imply that $(\bar{\vartheta}_k,\ \bar{q}_k;\ \gamma_k^*,\ \beta_k^*)$ satisfies the first-order KKT necessary conditions for the quadratic programming sub-problem (20). Therefore, $\bar{\vartheta}_k,\ \bar{q}_k$ represents the optimal solution for (20). However, this contradicts the assertion that $(\vartheta^*,\ q^*)$ represents the unique optimal solution for (20). As a result, (ϑ_k, q_k) converges to (ϑ^*, q^*).

Specifically, if q_k tends towards 0 and the MFCQ is satisfied at every x_k, then by substituting $q^* = 0$ into (11) - (16) for $(x^*,\ \vartheta^*,\ q^*, \gamma^*,\ \beta^*)$, and following the steps in Lemma 2, it can be readily demonstrated that x^* is a strongly critical point for the MOOPs. □

4 Numerical results

This section provides a comprehensive analysis of the ACMOP algorithm through numerical experiments. It includes an application in structural engineering and demonstrates its MATLAB implementation across a variety of test problems, including both convex and non-convex cases with linear and nonlinear constraints. The steps of the ACMOP algorithm are presented clearly and concisely to facilitate reader comprehension, supported by practical examples of its implementation and an application that highlights its relevance in structural engineering. The analysis evaluates the algorithm's effectiveness using carefully selected test problems, showcasing its versatility and theoretical foundations. The initial points are chosen randomly using a uniform distribution within specified bounds, ensuring a fair comparison across different problems.

4.1 Numerical experiment with test problems

The ACMOP algorithm underwent rigorous testing to evaluate its effectiveness, employing a diverse set of benchmark problems, including both convex and non-convex cases. The resulting data were thoroughly analyzed to assess performance, with compasrions made against the Zuo method [38], using MATLAB 2018b on a system with an Intel Core i7 processor, 12 GB RAM, and 3.40 GHz CPU.

The dual problem was solved using the GUROBI package with $\eta = 10^{-4}$. Table 1 summarizes the test problems, including their names, dimensions, objective function counts, bounds on variables, references, and the counts linear and nonlinear constraints in columns 7 and 8.

The nonmonotone line search described earlier was employed to determine the step size in each iteration, with parameters $\theta = 0.9$, $L = 0.83$, $\rho = 0.1$, and a minimum step size of 10^{-3}. These parameter values were selected empirically after preliminary testing to ensure stable and efficient convergence. The same step-size strategy was applied to both the ACMOP and Zuo methods. The stopping criterion for the ACMOP is defined as $\phi(x_k) > -\varepsilon$, where $\varepsilon = 10^{-6}$.

Testing was conducted with 100 identical starting points per problem, each consisting 100 iterations, using randomly uniformly distributed initial values within specified bounds. Table 2 presents average number of iterations (Iter), function evaluations (Fval), constraint evaluations (Gval), and CPU time (Time) for the nonmonotone line search techniques across test problems with two to three objective functions. The initial points for each problem were selected using a uniform random distribution to ensure unbiased sampling.

The number of initial points selected for an optimization problem is critical to the performance of the ACMOP algorithm, as the algorithm is highly dependent on the choice of starting points. For simple or convex problems, 100 initial points are typically sufficient. This number ensures comprehensive coverage of the search space, enabling the algorithm to explore the space efficiently and identify the global optimum without excessive computational cost.

However, for complex and non-convex problems, the search space is often larger and more intricate, typically containing multiple local optima and complex constraints. In these cases, a higher number of initial points is necessary to ensure thorough exploration of the search space. The exact number of points should depend on the complexity of the problem, as increasing the number of initial points enhances the algorithm's robustness, helps avoid local optima, and improves the likelihood of finding the global optimum.

For high-dimensional or particularly challenging problems, where both the objective functions and constraints are complex, a larger number of initial points becomes even more critical. This ensures better coverage of the objective space and increases the algorithm's exploration potential, leading to improved performance and convergence.

A reasonable approach is to start with 100 initial points for simpler problems and then increase the number of points for more complex cases based on the results. For non-convex or high-dimensional problems, scaling up the number of initial points is crucial to ensure comprehensive coverage of the search space and enhance the chances of convergence to the global optimum. It is important to recognize that the performance of the algorithm is heavily influenced by the choice of starting points, as they form the foundation for the search process and significantly affect both the quality and speed of convergence.

This study demonstrates the superior performance of the proposed ACMOP algorithm compared to the Zuo method [38], particularly in terms of convergence speed and computational efficiency. It is noteworthy that in the study by Morovati et al., the Zuo method was compared with population-based methods, analyzing the computational cost differences.

Since this previous work already provides a comprehensive comparison with population-based approaches, the focus here is on highlighting the specific advantages of the ACMOP algorithm over the Zuo method, rather than revisiting comparisons with population-based methods. Including such a comparison again would introduce redundancy and unnecessarily complicate the narrative.

Although a direct comparison with population-based methods was not conducted in this study, it is well-established that these methods typically incur higher computational costs due to the need to evaluate multiple candidate solutions simultaneously in each generation. In contrast, ACMOP employs an iterative, point-based approach, which, for many problems, facilitates faster convergence with fewer function evaluations, particularly in smaller or moderately complex search spaces.

Numerical comparisons demonstrate the superior performance of ACMOP over the Zuo method in evaluating parameters using non-monotone techniques. Approximate pareto fronts for the test problems are shown in Figures 2–7.

While Algorithm 1 was primarily designed for convex problems, it was also applied to non-convex problems. To address potential non-convergence in such cases, iterations were limited to 500, in addition to the stopping criterion $\phi(x_k) > -\varepsilon$. These findings suggest that the proposed algorithm could significantly contribute to optimization research, promoting the development of more efficient methods.

The ACMOP algorithm, a highly efficient solution, has been rigorously tested across a range of real-world and complex optimization problems, each featuring multiple variables and intricate constraints.

Table 1: Comprehensive overview of applied test problems

Problem	**m**	**n**	x_L	x_U	**linear-Consts**	**nonlinear-Consts**	**Reference**
ABC-Comp	2	2	[0;0]	[10;10]	1	0	[30]
E2	2	2	[0;0]	[7;7]	1	0	[18]
IYK	3	3	[0;0;1.2]	[π;10;10]	1	0	[19]
test4	2	5	[-2;-2;-2;-2;-2]	[2;2;2;2;2]	2	0	[9]
Binh	2	2	[0;0]	[5;3]	2	0	[15]
OSY	2	6	[0;0;1;0;1;0]	[10;10;5;6;5;10]	4	2	[15]
Hanne2	2	2	[0;0]	[5;5]	1	0	[13]
Hanne3	2	2	[0;0]	[10;10]	1	0	[13]
Hanne4	2	2	[0;0]	[10;10]	0	1	[13]
SRN	2	2	[-20;-20]	[20;20]	1	1	[15]
TNK	2	2	[0;0]	[π;π]	0	2	[15]
CTP1	2	2	[0;0]	[1;1]	0	2	[14]
CTP2	2	4	[0;-5;-5;-5]	[1;5;5;5]	0	2	[14]
DTLZ0	3	3	[0;0;0]	[2;1;1]	1	1	[16]
KITA	2	2	[0;0]	[4;4]	3	0	[31]
T5	3	4	[-2;-2;-2;-2]	[2;2;2;2]	1	0	[17]
T6	2	3	[-2;-2;-2]	[2;2;2]	1	0	[17]
P1	2	5	[0;0;0;0;-4]	[1;1;1;1;1]	1	0	[20]
P2	3	4	[-2;-2;-2;-2]	[2;2;2;2]	2	1	[20]
ex1	2	3	[.4;0;0]	[2.5;1;1]	1	0	[10]
DAS-CMOOPI.a	2	5	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPI.b	2	30	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPI.c	2	100	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPI.d	2	500	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPII.a	2	5	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPII.b	2	30	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPII.c	2	100	[0;...; 0]	[1;...; 1]	0	11	[23]
DAS-CMOOPII.d	2	500	[0;...; 0]	[1;...; 1]	0	11	[23]

Two problems from the DAS-CMOOP set were selected for testing, each evaluated in four different configurations. These test cases were specifically chosen to challenge the algorithm's robustness and scalability, and the results confirm the algorithm's efficiency even as problem complexity increases.

The algorithm demonstrated exceptional scalability, with both execution time and evaluation counts (including function and constraint evaluations) increasing predictably and linearly as the problem size grows. Specifically, for DAS-CMOOPI.a (n = 5), the execution time was 0.04 seconds, whereas for DAS-CMOOPI.d (n = 500), it increased to 1.51 seconds. This linear progression aligns with the expected O(n) time complexity, reinforcing the algorithm's remarkable capacity to handle large-scale optimization tasks. When compared to Zuo's method and other benchmark algorithms, ACMOP outperformed them not only in execution speed but also in solution accuracy, identifying more pareto-optimal solutions and

Table 2: Comparison of ZMO and ACMOP numerical methods using nonmonotone line search on multiple test problems based on iterations (Iter), CPU time (Time), function evaluations (Feval), and constraint evaluations (Geval)

Problem	**Zoutendijk**				**ACMOP**			
Name	**Iter**	**Time**	**Feval**	**Geval**	**Iter**	**Time**	**Feval**	**Geval**
ABC_Comp	14.07	0.102	331.33	22.79	7.74	0.09	7.92	7.93
E2	5.60	0.040	4.60	4.59	2.42	0.033	2.42	2.39
IYK	7.31	0.057	6.33	6.31	1.82	0.028	1.82	1.82
Test4	15.21	0.107	741.11	32.94	8.01	0.09	53.53	8.66
Binh	82.61	1.18	82.07	91.13	3.35	0.054	3.36	2.94
Osy	52.98	0.36	86.88	59.36	7.25	0.091	7.25	6.96
Hanne2	20.62	0.17	19.62	19.44	1.00	0.04	1.00	1.00
Hanne3	32.29	0.24	31.29	30.47	2.53	0.036	3.53	2.72
Hanne4	22.24	0.14	21.91	22.44	6.78	0.09	39.53	3.08
SRN	14.32	0.20	14.31	14.31	10.89	0.12	10.99	10.42
TNK	2.66	0.021	1.66	1.67	2.48	0.038	2.49	2.45
CTP1	164.21	1.32	163.21	184.38	2.22	0.028	2.22	2.22
CTP2	183.43	1.42	182.43	216.71	1.39	0.020	1.38	1.41
DTLZ0	10.46	0.09	10.91	9.23	7.19	0.09	8.23	7.26
KITA	4.65	0.05	3.65	3.66	2.14	0.03	2.14	1.87
T5	9.59	0.07	8.59	8.65	1.94	0.03	1.94	1.93
T6	6.68	0.04	5.68	5.68	1.74	0.02	1.74	1.73
P1	45.57	0.38	44.59	44.36	37.70	0.47	37.70	36.63
P2	5.37	0.05	4.37	4.29	2.79	0.04	2.79	2.73
ex1	31.20	0.30	30.20	24.73	2.10	0.03	2.10	2.17
DAS-CMOOPI.a	327.27	3.11	638.26	326.27	2.02	0.04	1.02	1.02
DAS-CMOOPI.b	379.89	4.62	755.61	324.18	2.61	0.06	12.66	1.62
DAS-CMOOPI.c	180.11	3.80	359.64	172.47	4.68	0.38	33.87	3.53
DAS-CMOOPI.d	364.55	29.38	727.99	359.93	5.52	1.51	17.52	4.48
DAS-CMOOPII.a	346.08	3.33	676.59	345.08	2.02	0.05	1.02	1.02
DAS-CMOOPII.b	411.64	4.26	822.74	369.78	2.82	0.06	15.23	1.87
DAS-CMOOPII.c	332.51	5.07	663.82	295.26	4.62	0.23	18.16	3.47
DAS-CMOOPII.d	386.57	32.19	772.18	354.88	7.79	2.06	75.41	7.16

demonstrating superior efficiency in addressing complex constraints.

In addition, the ACMOP algorithm, a practical and applicable tool, was tested on more challenging configurations to assess its ability to balance computation time and evaluation costs. Despite the added complexity, the algorithm successfully maintained optimal performance while scaling effectively. These findings highlight ACMOP's practical applicability, making it a robust tool for solving complex, high-dimensional MOOPs that require both high solution accuracy and computational efficiency.

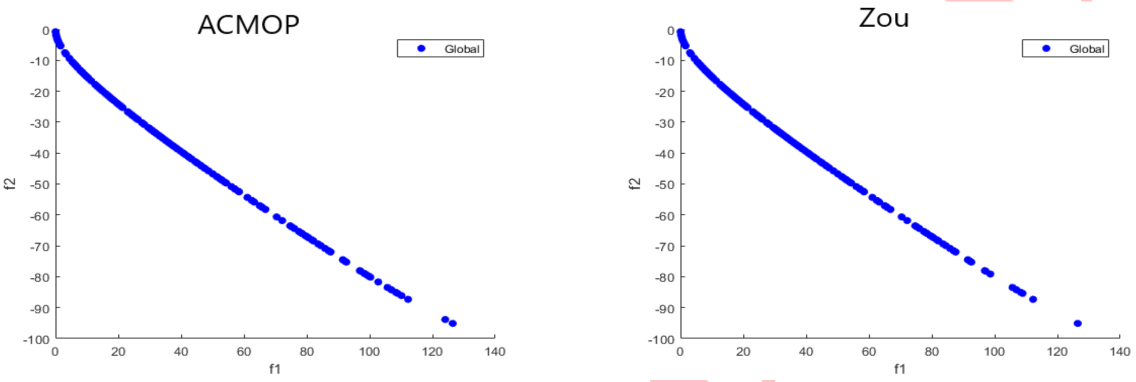


Figure 2: Pareto front for the ABC_Comp Problem

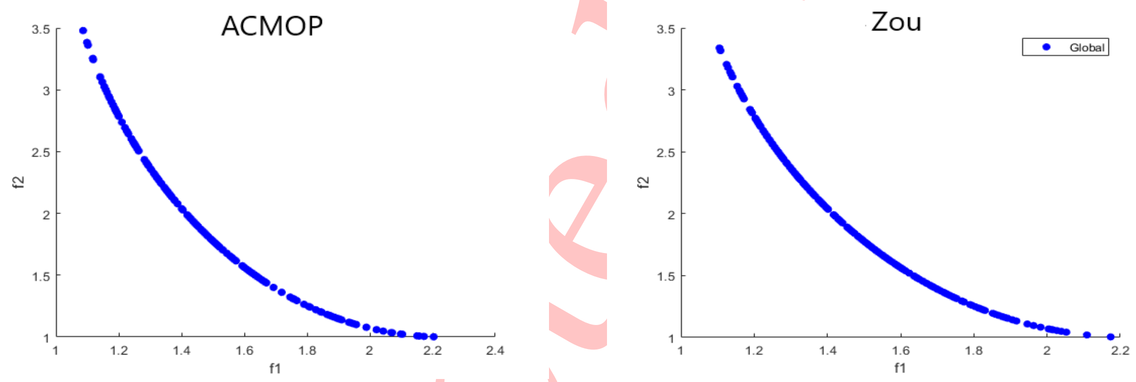


Figure 3: Pareto front for the E2 Problem

5 Performance Evaluation with Performance Profiles

In this section, Performance Profiles are introduced as an effective method for evaluating and comparing solvers in MOOPs. A performance profile is represented by a cumulative distribution function, denoted as $\rho_s(\tau)$, which characterizes the performance of solver s across a set of benchmark problems P. The performance ratio $r_{p,s}$ for solver s on problem p is defined as:

$$r_{p,s} = \frac{t_{p,s}}{\min\{t_{p,s} : s \in S\}},$$

where $t_{p,s}$ is the performance of solver s on problem p, and $\min t_{p,s} : s \in S$ refers to the best performance of any solver on problem p. A lower value of $r_{p,s}$ indicates superior performance of solver s on problem p.

The cumulative distribution function $\rho_s(\tau)$ for solver s is defined as:

$$\rho_s(\tau) = \frac{1}{|P|}\{p \in P : r_{p,s} \leq \tau\}.$$

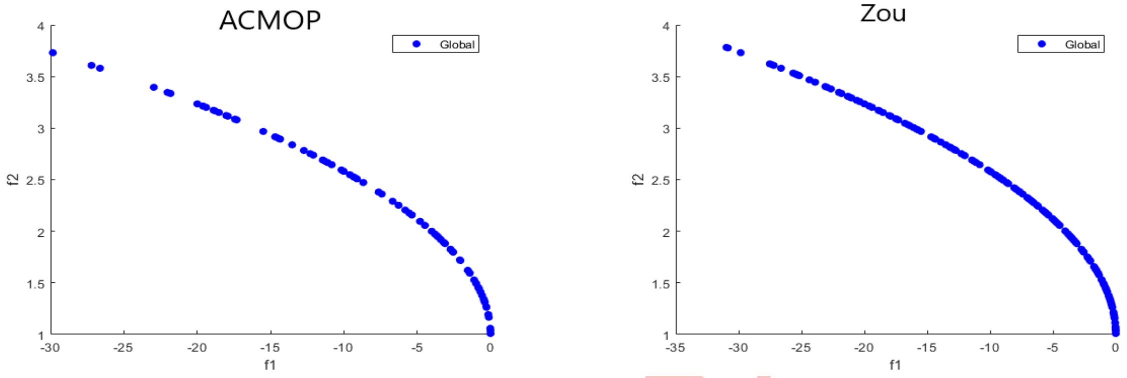


Figure 4: Pareto front for the KITA Problem

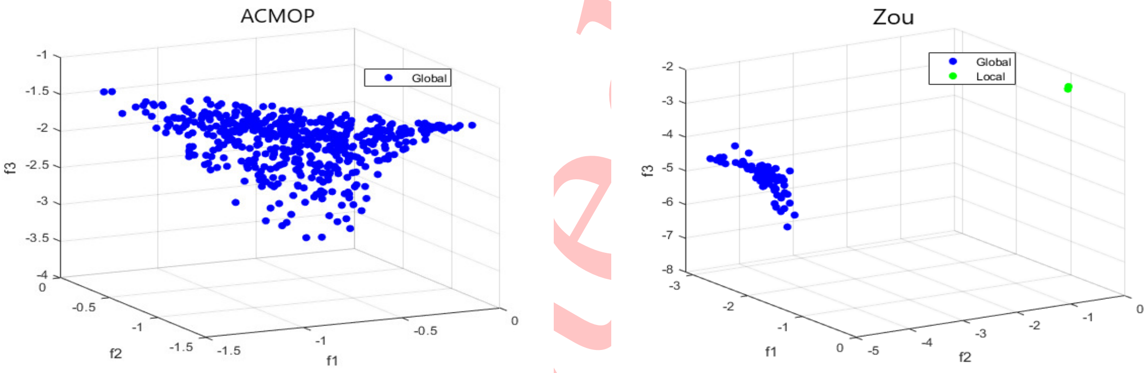


Figure 5: Pareto front for the IYK Problem

Thus, $\rho_s(\tau)$ represents the proportion of problems in the set P for which the performance ratio $r_{p,s}$ is less than or equal to τ. A higher value of $\rho_s(1)$ indicates that solver s performs better overall, while solvers that exhibit higher values for larger τ are considered more robust, demonstrating their ability to solve a broader range of problems.

Performance profiles provide a means for comparative analysis of solvers at various performance thresholds. These profiles highlight key aspects of solver performance, including convergence speed, solution quality, and the ability to tackle both simple and complex problem instances. By analyzing these profiles, the strengths and weaknesses of each solver can be identified, contributing to the development of more efficient and robust algorithms.

Depicted as cumulative distribution function plots, performance profiles facilitate the comparison of solvers across different performance levels. These plots not only reveal the solvers' convergence characteristics but also assess their robustness in addressing problems of varying complexity, particularly for non-convex cases. Solvers that maintain higher $\rho_s(\tau)$ values at larger τ are more resilient, showcasing their capability to address a wide spectrum of problem instances. The iterative refinement of solvers based on performance profiles ultimately fosters the design of more efficient and robust algorithms, assisting researchers in optimizing solver performance and making informed adjustments for

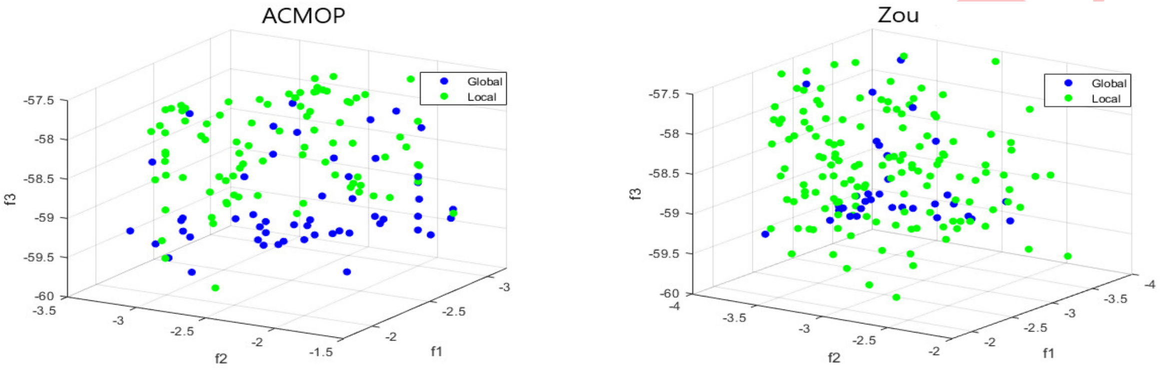


Figure 6: Pareto front for the P2 Problem

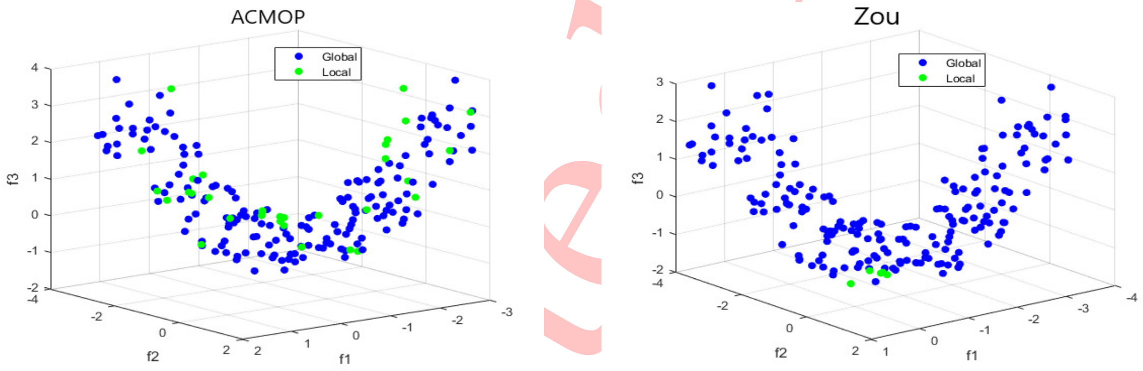


Figure 7: Pareto front for the T5 Problem

future iterations.

5.1 Evaluation of Different Solvers

In this section, the performance of five solvers is evaluated and compared: Zoutendijk, ACMOP, and three variants of ACMOP, each with different acceleration bounds for the parameter T_k. These variants are as follows:

- ACMOP (standard): With acceleration parameter $0 < T_k$.
- ACMOPI: With acceleration parameter $0 < T_k < 0.2$.
- ACMOPII: With acceleration parameter $0 < T_k < 0.5$.
- ACMOPIII: With acceleration parameter $0 < T_k < 0.95$.

These bounds were empirically determined, and the goal of this experiment is to compare the performance of these methods across several key metrics.

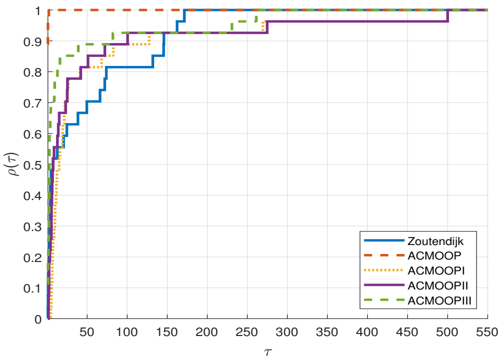


Figure 8: Comparison of algorithm performance in terms of mean iterations across all test problems. The plots show the pareto front for the overall performance of each algorithm, evaluating the efficiency of each method in terms of the number of iterations required to reach optimal solutions.

To evaluate and compare the solvers, five key metrics introduced in the Numerical Results section are utilized:

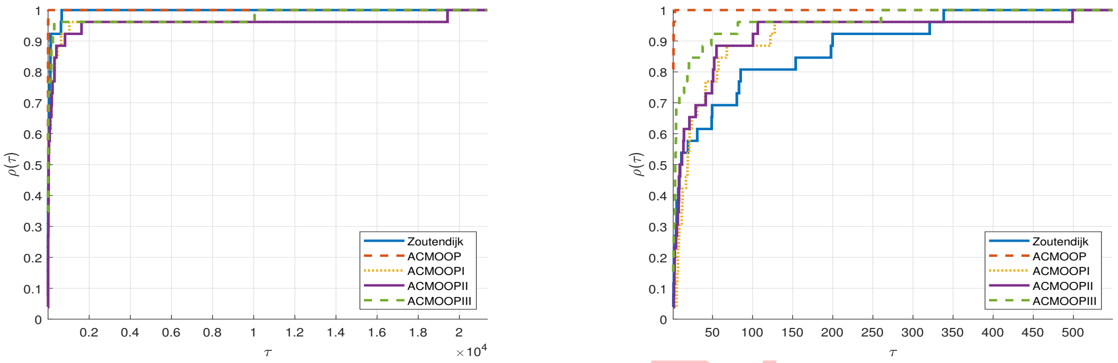
- **Number of points on pareto Front:** The number of solutions found by the solver that lie on the pareto front, indicating the solver's ability to approximate the entire pareto optimal set.

The performance of the solvers is compared using Performance Profiles, which are visualized by plotting cumulative distribution functions $\rho_s(\tau)$. These profiles represent the performance ratio for different solvers at various thresholds τ. A higher value of $\rho_s(1)$ indicates superior overall performance, and solvers that maintain high performance at larger τ values are considered more robust.

Figures 8-10 present a comprehensive comparison of the solvers across five key metrics: iterations, function evaluation values (Feval), constraint evaluation values (Geval), CPU time, and the number of pareto front points. These performance profiles illustrate the solvers' efficiency in balancing solution accuracy with computational cost. The Feval and Geval plots assess the solvers' capability to minimize objective functions while satisfying constraints. The iterations and CPU time charts evaluate their speed and efficiency in converging to optimal solutions. Collectively, these figures provide an in-depth evaluation, aiding in the identification of algorithms that perform optimally under specific conditions, whether in terms of faster convergence or higher-quality solutions.

The results obtained from the performance profiles demonstrate that the ACMOP algorithm consistently outperforms the other algorithms across all evaluation metrics. ACMOP achieves optimal results in the shortest time and with the fewest iterations, showcasing its high efficiency in solving complex optimization problems. When compared to other methods, ACMOP identifies more pareto front points and performs better in both satisfying the constraints (Geval) and minimizing the objective function values (Feval). These characteristics are particularly important in MOOPs that require a balance between solution accuracy and computational efficiency.

The algorithms ACMOPII and ACMOPIII also exhibited relatively good performance but fell short in certain metrics when compared to ACMOP. Specifically, ACMOP outperformed these methods in terms of execution time and the number of iterations. The Zoutendijk algorithm, widely recognized as a reference method in many studies, showed acceptable performance in some metrics such as execution



(a) Feval

(b) Geval

Figure 9: Pareto front for the overall performance across all test problems in terms of function values (Feval) and constraint values (Geval). The left plot shows the performance of the algorithms in terms of objective function values, and the right plot illustrates their performance in satisfying the constraints across all problems tested.

cpu time and iterations; however, it required more time to achieve similar results when compared to ACMOP. This highlights that ACMOP excels in both speed and efficiency.

In conclusion, the findings of this study identify ACMOP as the most optimal algorithm among the compared methods and a suitable choice for solving complex and diverse MOOPs. This algorithm is particularly advantageous in scenarios that require a precise balance between execution speed and result accuracy. The analyses and comparisons clearly confirm ACMOP's superiority over the other algorithms.

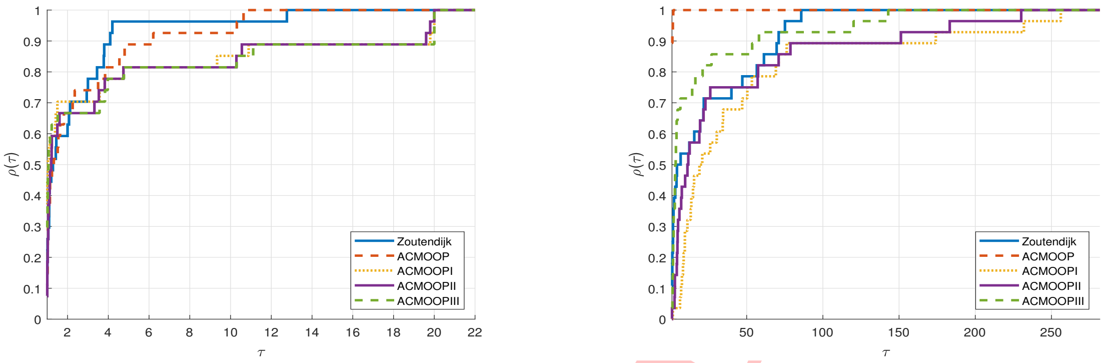
5.2 Optimization of a speed reducer design problem

Gearboxes are essential mechanical systems, with the speed reducer being a critical component. Designing this component presents a complex optimization problem, initially formulated by Golinski [28]. Achieving the desired design requires addressing seven design variables, each serving distinct purposes across various applications. Azarm et al. [6] proposed a method to reformulate the problem as a two-objective optimization problem, deviating from the original three-objective formulation.

The illustration below depicts a typical gearbox configuration commonly used in lightweight aircraft. An alternative representation of the speed reducer is also provided, with the design variables clearly identified. The problem formulation involves seven design variables: gear face width (x_1), teeth module (x_2), the number of teeth on the pinion (x_3), the distance between bearings on shaft 1 (x_4), the distance between bearings on shaft 2 (x_5), the diameter of shaft 1 (x_6), and the diameter of shaft 2 (x_7).

The primary design objective is to minimize the volume, denoted as C_1. The second objective, C_2, focuses on minimizing the stress in one of the two gear shafts.

The speed reduction design is subject to various constraints based on gear and shaft design principles. Each of the seven design variables is bound by specific limits, and there are 11 inequality constraints to consider. One of these constraints is specifically imposed on the first objective. These constraints are summarized as follows: G_1 denotes the maximum allowable bending stress of the gear tooth; G_2 signifies the maximum allowable contact stress of the gear tooth. In contrast, G_3 and G_4 denote the upper limits for the transverse deflection of the shafts. Constraints G_5 through G_7 represent dimensional restrictions



(a) Pareto front points

(b) CPU time

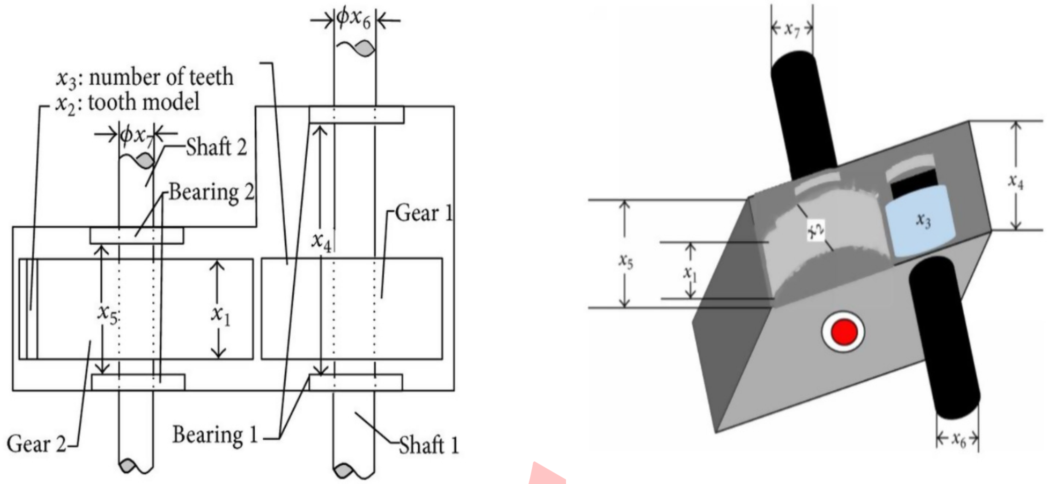
Figure 10: Comparison of algorithms based on the mean number of pareto front points and the mean CPU time required to reach the optimal solution across all test problems. These figures illustrate the performance and computational efficiency of each algorithm.

derived from spatial considerations and/or empirical knowledge. G_8 and G_9 pertain to shaft design requirements based on empirical knowledge, whereas G_{10} and G_{11} impose limits on the stress within the gear shafts.

The speed-reduction design problem is formulated mathematically as follows:

$$\begin{aligned}
&min\ C_1 = 0.7854x_1x_2^2\left(\frac{10}{3}x_3^2 + 14.933x_3 - 43.0934\right) - 1.508x_1\left(x_6^2 + x_7^2\right) + \\
&7.477\left(x_6^3 + x_7^3\right) + 0.7854\left(x_4x_6^2 + x_5x_7^2\right) \\
&min\ C_2 = \frac{\sqrt{745\frac{x_4}{x_2x_3} + 1.69\times 10^7}}{0.1x_6^3} \\
&s.t.
\end{aligned}$$

$$\begin{aligned}
&G1: \frac{1}{x_1x_2^2x_3} - \frac{1}{27} \le 0, && G2: \frac{1}{x_1x_2^2x_3^2} - \frac{1}{397.5} \le 0, \\
&G3: \frac{x_4^3}{x_2x_3x_6^4} - \frac{1}{1.93} \le 0, && G4: \frac{x_5^3}{x_2x_3x_7^4} - \frac{1}{1.93} \le 0, \\
&G5: x_2x_3 - 40 \le 0, && G6: \frac{x_1}{x_2} - 12 \le 0, \\
&G7: 5 - \frac{x_1}{x_2} \le 0, && G8: 1.9 - x_4 + 1.5x_6 \le 0, \qquad (39) \\
&G9: 1.9 - x_5 + 1.1x_7 \le 0, && G10: \frac{\sqrt{745\frac{x_4}{x_2x_3} + 1.69\times 10^7}}{0.1x_6^3} \le 1800, \\
&G11: \frac{\sqrt{745\frac{x_5}{x_2x_3} + 1.575\times 10^8}}{0.1x_7^3} \le 1100,
\end{aligned}$$



(a) **(b)**

Figure 11: Speed reducer design problem

$$2.6 \le x_1 \le 3.6, \quad 0.7 \le x_2 \le 0.8, \quad 17 \le x_3 \le 28, 7.3 \le x_4, x_5 \le 8.3, \quad 2.9 \le x_6 \le 3.9, \quad 5.0 \le x_7 \le 5.5.$$

The speed-reduction optimization problem is addressed using the ACMOP algorithm. A total of 100 initial points were employed, with the parameters uniformly distributed within their respective lower and upper bounds. This paper exclusively reports the numerical results obtained using the ACMOP algorithm. The average number of iterations is 1.3154, and the mean number of function evaluations is 4.86. Additionally, the CPU time required to complete the task is measured at 0.0251 seconds.

Figure 12 presents the non-dominated frontier obtained using the ACMOP algorithm. As detailed in this paper, these results highlight the algorithm's effectiveness in solving the speed-reduction optimization problem.

6 Conclusions

This study introduces a novel and efficient approach for CMOOPs through the ACMOP algorithm, a non-parametric, direction-based iterative method. The algorithm begins by solving a convex sub-problem to determine the optimal search direction using the dual form of the original problem. An acceleration strategy is incorporated into the iterative framework to enhance convergence and solution quality, while a nonmonotone line search technique adaptively adjusts the step size to improve robustness. A key feature of the method is the careful initialization of feasible solutions, which is crucial for accurately estimating the non-dominated frontier. The effectiveness of the algorithm is validated through standard benchmark problems, with numerical comparisons against the Zoutendijk method demonstrating ACMOP produces superior or competitive Pareto fronts. Additionally, the successful application of ACMOP to the practical speed-reduction design problem, supported by extensive numerical results, highlights its practical potential. In summary, ACMOP emerges as a powerful and versatile optimization tool for addressing

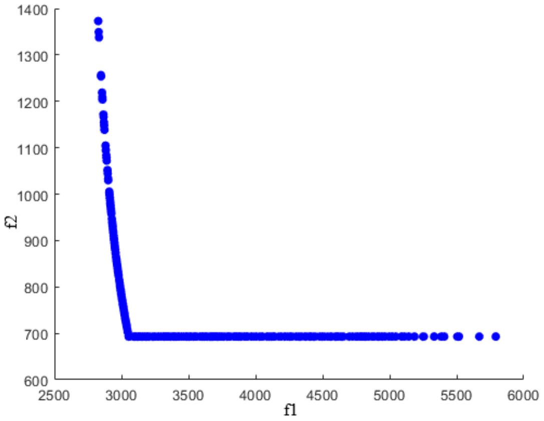


Figure 12: The design problem of the speed reducer's pareto front

complex CMOOPs. Its promising performance suggests significant potential for broader applications across various engineering and scientific fields. This paper offers a detailed proof showing that the ACMOP algorithm reliably converges when applied to convex problems. However, we are still grappling with the challenge of ensuring convergence for non-convex problems. These types of problems often have multiple local solutions and intricate constraints, which complicate the process of finding a global solution. While this study focuses on convex optimization, it also recognizes the significance of addressing the challenges posed by non-convex problems. Future research aimed at developing methods that can ensure convergence in non-convex situations could significantly enhance the applicability of ACMOP across a broader range of MOOPs.

References

[1] M. Ahookhosh, K. Amini, S. Bahrami, *A class of nonmonotone Armijo-type line search method for unconstrained optimization*, Optimization **61** (2012) 387-404.

[2] M. Anitescu, *A superlinearly convergent sequential quadratically constrained quadratic programming algorithm for degenerate nonlinear programming*, SIAM J. Optim. **12** (2002) 949-978.

[3] M.A.T. Ansary, G. Panda, *A modified quasi-newton method for vector optimization problem*, Optimization **64** (2015) 2289–2306.

[4] M.A.T. Ansary, G. Panda, *A sequential quadratic programming method for constrained multi-objective optimization problems*, J. Appl. Math. Comput.**64** (2020) 379-397.

[5] M.A.T. Ansary, G. Panda, *A globally convergent sqcqp method for multiobjective optimization problems*, *SIAM J. Optim.* **31** (2021) 91-113.

[6] S. Azarm, W.C. Li, *Optimality and constrained derivatives in two-level design optimization*, J. Mech. Des. **112(4)** (1990)) 563-568.

[7] H. Basirzadeh, V. Morovati, A. Sayadi, *A quick method to calculate the super-efficient point in multi-objective assignment problems*, J. Math. Comput. Sci. **10** (2014) 157-162.

[8] A. Beck, M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM J. Imaging Sci. **2** (2009) 183-202.

[9] T. Binh, *A multiobjective evolutionary algorithm: The study cases*, Technical report, Institute for Automation and Communication, Barleben, Germany, 1999.

[10] G. Cabrera-Guerrero, M. Ehrgott, A.J. Mason, A. Raith, *Biobjective optimisation over a set of convex sub-problems*, Ann. Oper. Res. **319** (2022) 1507-1532.

[11] E. Carrizosa, J.B.G. Frenk, *Dominating sets for convex functions with some applications*, J. Optim. Theory Appl. **96** (1998) 281-295.

[12] X. Chen, M.M. Kostreva, *Methods of feasible directions: A review*, in: Progress in Optimization, X. Yang, A. I. Mees, M. Fisher, and L. Jennings (eds.), Applied Optimization, vol. 39, Springer, Boston, MA, 2000.

[13] Y. Collette, P. Siarry, *Multiobjective Optimization: Principles and Case Studies*, Springer Science and Business Media, 2013.

[14] K. Deb, A. Pratap, T. Meyarivan, *Constrained test problems for multi-objective evolutionary optimization*, Lect. Notes Comput. Sci. (1993) 284-298.

[15] K. Deb, *Multi-objective genetic algorithms: Problem difficulties and construction of test problems*, Evol. Comput. **7** (1999) 205-230.

[16] K. Deb, L. Thiele, M. Laumanns, E. Zitzler, *Scalable test problems for evolutionary multiobjective optimization*, In: Abraham, A., Jain, L., Goldberg, R. (eds) Evolutionary Multiobjective Optimization. Advanced Information and Knowledge Processing, Springer, London, 2005.

[17] M. De Santis, G. Eichfelder, J. Niebling, S. Rocktäschel, *Solving multiobjective mixed integer convex optimization problems*, SIAM J. Optim. **30** (2020) 3122-3145.

[18] G. Eichfelder, *Adaptive Scalarization Methods in Multiobjective Optimization*, Springer, Berlin, 2008.

[19] G. Eichfelder, *An adaptive scalarization method in multiobjective optimization*, SIAM J. Optim. **19** (2009) 1694-1718.

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

[21] M. El Moudden, A. El Mouatasim, *Accelerated diagonal steepest descent method for unconstrained multiobjective optimization*,J. Optim. Theory Appl. **188** (2021) 220-242.

[22] M. Ehrgott, *Multicriteria Optimization*, Springer, Berlin, 2005.

[23] Z. Fan, W. Li, X. Cai, H. Li, C. Wei, Q. Zhang, K. Deb, E. Goodman, *Difficulty adjustable and scalable constrained multiobjective test problem toolkit*, Evol. Comput. **28(3)** (2020) 339–378.

[24] J. Fliege, L.M. G. Drummond, B. F. Svaiter, *Newton's method for multiobjective optimization*, SIAM J. Optim. **20** (2009) 602-626.

[25] J. Fliege, B.F. Svaiter, *Steepest descent methods for multicriteria optimization*, Math. Methods Oper. Res. **51** (2000) 479-494.

[26] J. Fliege, A.I. Vaz, *A method for constrained multiobjective optimization based on sqp techniques*, SIAM J. Optim. **26** (2016) 2091-2119.

[27] M. Fukushima, Z.Q. Luo, P. Tseng, *A sequential quadratically constrained quadratic programming method for differentiable convex minimization*, *SIAM J. Optim.* **13** (2003) 1098-1119.

[28] J. Golinski, *Investigation of a certain stray process applied to the optimum synthesis problems*, *Computing* **6** (1970) 139-160.

[29] L. Grippo, F. Lampariello, S. Lucidi, *A nonmonotone line search technique for Newton's method*, SIAM J. Numer. Anal. **23** (1986) 707-716.

[30] C.L. Hwang, A.S.M. Masud, *Multiple objective decision making methods and applications*, Lecture Notes in Economics and Mathematical Systems, vol. 164, Springer, Berlin, 1979.

[31] H. Kita, Y. Yabumoto, N. Mori, Y. Nishikawa, *Multi-objective optimization by means of the thermodynamical genetic algorithm*, in: Parallel Problem Solving from Nature—PPSN IV: International Conference on Evolutionary Computation—The 4th International Conference on Parallel Problem Solving from Nature Berlin, Germany, September 22–26, 1996 Proceedings 4, Springer, Berlin, 1996, pp. 504–512.

[32] T. Maeda, *Constraint qualifications in multiobjective optimization problems: Differentiable case*, J. Optim. Theory Appl. **80** (1994) 483-500.

[33] S. Menon, J. Karl, K. Wignaraja, *Handbook on planning, monitoring and evaluating for development results*, UNDP Evaluation Office, New York, **68** (2009) 10.

[34] K.M. Miettinen, *Nonlinear Multiobjective Optimization*, Kluwer, Boston, 1999.

[35] S.H. Mirzaie, A. Ashrafi, *Incorporating non-monotone trust region algorithm with line search method for unconstrained optimization*, J. Math. Model. **13(1)** (2025) 219-233.

[36] K. Mita, E.H. Fukuda, N. Yamashita, *Nonmonotone line searches for unconstrained multiobjective optimization problems*, J. Global Optim. **75** (2019) 63-90.

[37] V. Morovati, L. Pourkarimi, H. Basirzadeh, *Barzilai and borwein's method for multiobjective optimization problems*, Numer. Algorithms **72** (2016) 539-604.

[38] V. Morovati, L. Pourkarimi, *Extension of zoutendijk method for solving constrained multiobjective optimization problems*, European J. Oper. Res. **273** (2019) 44-57.

[39] S. Mostaghim, J. Branke, H. Schmeck, *Multi-objective particle swarm optimization on computer grids*, in: Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, ACM, New York, 2007, pp. 869–875.

[40] Y.E. Nesterov, *A method of solving a convex programming problem with convergence rate* $O(k^2)$, Dokl. Akad. Nauk **269** (1983) 543-547.

[41] J. Nocedal, S.J. Wright, *Numerical Optimization*, Springer, New Delhi, India, 2006.

[42] V.M. Panin, *Some methods of solving convex programming problems*, USSR Comput. Math. Math. Phys. **21** (1981) 57-72.

[43] J.L.J. Pereira, G.A. Oliver, M.B. Francisco, S.S. Cunha, G.F. Gomes, *A review of multi-objective optimization: Methods and algorithms in mechanical engineering problems*, Arch. Comput. Methods Eng. **29** (2022) 2285–2308.

[44] Ž. Povalej, *Quasi-Newtons method for multiobjective optimization*, J. Comput. Appl. Math. **255** (2014) 765-777.

[45] H. Salmei, M. Namjoo, *Improved feasible value constraint for multiobjective optimization problems*, J. Math. Model. **13(1)** (2025) 105-120.

[46] O. Sener, V. Koltun, *Multi-task learning as multi-objective optimization*, Adv. Neural Inf. Process. Syst. **31** (2018) 525-536.

[47] C.M. Tang, J.B. Jian, G.Y. Li, *A working set SQCQP algorithm with simple nonmonotone penalty parameters*, J. Appl. Math. Comput. Mech. **236** (2011) 1382-1398.

[48] H. Tanabe, E.H. Fukuda, N. Yamashita, *An accelerated proximal gradient method for multiobjective optimization*, Comput. Optim. Appl. (2023) 1-35.

[49] Z. Wang, J. Li, G.P. Rangaiah, Z. Wu, *Machine learning aided multi-objective optimization and multi-criteria decision making: Framework and two applications in chemical engineering*, Comput. Chem. Eng. **165** (2022) 107945.

[50] X. Wang, Y. Wang, G. Wang, *An accelerated augmented Lagrangian method for multi-criteria optimization problem*, J. Ind. Manag. Optim. **16** (2020) 1-9.

[51] H.R. Yousefzadeh, E. Zahiri, A. Heydari, *Ranking the Pareto frontiers of multi-objective optimization problems by a new quasi-Gaussian evaluation measure*, J. Math. Model. **11(1)**, (2023) 45-58.

[52] H. Zhang, W.W. Hager, *A nonmonotone line search technique and its application to unconstrained optimization*, SIAM J. Optim. **14** (2004) 1043–1056.