J. Appl. Numer. Optim. 8 (2026), No. 3, pp. 355-372
Available online at http://jano.biemdas.com
https://doi.org/10.23952/jano.8.2026.3.03

# A TRUST REGION PROXIMAL GRADIENT METHOD FOR COMPOSITE MULTI-OBJECTIVE OPTIMIZATION PROBLEMS

MD ABU TALHAMAINUDDIN ANSARY

*Department of Mathematics, Indian Institute of Technology Jodhpur, Jodhpur, India*

**Abstract.** This paper proposes a trust region proximal gradient method for composite multi-objective optimization problems. Global convergence of the proposed method is proved under some mild assumptions. Our method is validated and compared with some recent methods via a set of problems.
**Keywords.** Convex optimization; Critical point; Multi-objective optimization; Proximal gradient method; Trust region method.
**2020 Mathematics Subject Classification.** 90C25, 90C29, 65K10.

## 1. Introduction

Multiple objective functions are simultaneously minimized in a multi-objective optimization problem. Scalarization techniques, which convert the original problem to a single objective optimization problem using a set of predetermined parameters, are traditional approaches for tackling multi-objective optimization problems [1, 2]. To discover the approximate Pareto front in various engineering issues, heuristic techniques such as evolutionary algorithms [3, 4] were frequently employed. However, heuristic algorithms do not provide convergence to the solution, and scalarization techniques involve some user-dependent parameter or ordering information of objective functions. Researchers introduced gradient/subgradient-based approaches for multi-objective optimization problems as a result of the drawbacks of traditional approaches. Gradient/subgradient-based methods for single-objective optimization were extended to multi-objective cases recently. For nonsmooth multi-objective optimization problems, researchers developed various strategies. Line search techniques [5, 6, 7, 8, 9, 10] or trust region methods [11, 12, 13, 14, 15] were gradient-based approaches for multi-objective optimization issues. Proximal point methods [16, 17], proximal gradient methods [18, 19, 20], and single objective subgradient methods [21, 22, 23] have been extended to multi-objective cases by using these techniques. In [24, 25, 26, 27], descent methods were developed for uncertain multi-objective optimization problems with finite uncertainty. Proximal gradient methods is now treated as an effective way to handle composite single objective optimization problems [28, 29, 30]. This method uses a linear approximation of a smooth function to choose an appropriate descent direction at each iteration. According to Lee et al. [31], every iteration of the proximal gradient method employs a quadratic approximation of a smooth function. This method quadratically

E-mail address: mdabutalha@iitj.ac.in; md.abutalha2009@gmail.com.
Received 16 July 2025; Accepted 26 May 2026; Published online 1 August 2026.

converges under a few minor assumptions. Recently, a proximal gradient approach for multi-objective optimizations was developed by Tanabe et al. [20]. For multi-objective optimization problems, this approach combines the concepts of the proximal point method and the steepest descent technique, as developed in [5] and [17], respectively. This method has a modest convergence rate, similar to the single-objective proximal gradient method. The methodology proposed in [20] is further modified in different directions [18, 19, 32]. The steepest descent approach is replaced by the Newton method in [18] and the quasi-Newton method in [19]. The Newton-type proximal gradient approach ([18]) employs a quadratic approximation of each smooth function and solves a subproblem at each iteration. The quasi-Newton approach in [19] substitutes a positive definite matrix for each smooth function's Hessian. The subproblems employed in [18, 19] fail to produce bounded solutions if the Hessian or approximate-Hessian matrices are ill-conditioned, just like Newton and quasi-Newton methods for smooth problems. Additionally, in [18, 19], certain inexact line search methods were employed to determine an appropriate step length, which raises the computational expenses. Using the concepts of trust region approaches in composite multi-objective optimization problems is one potential strategy to avoid these restrictions. This has inspired us to make this scientific contribution. In this study, we introduce the concept of the trust region method and develop a trust region proximal gradient strategy for composite multi-objective optimization problems.

The following is the outline of this paper. Section 2 discusses definitions and lemmas. Section 3 develops a proximal gradient for the trust region. An algorithm is proposed in this section, which is a trust region variant proximal gradient method for composite multi-objective optimization problems. Global convergence of the proposed method is analyzed under some mild assumptions in Section 4. The suggested approach is validated and compared with some recent methods via a set of problems in Section 5. Section 6 ends this paper with some concluding remarks.

## 2. Preliminaries

This paper considers the following multi-objective function:

$$(MOP): \min_{x\in\mathbb{R}^n} F(x) = (F_1(x), F_2(x), ..., F_m(x)).$$

Assume that, for $j = 1,2,...,m$, $F_j : \mathbb{R}^n \to \mathbb{R}$ is defined by $F_j(x) = f_j(x) + g_j(x)$, where $f_j$ is a convex and continuously differentiable function and $g_j$ is a convex and continuous function which is not necessarily differentiable. It is assumed that $int(dom\ g_j) \neq \emptyset$ holds for every $g_j$. Denote $\Lambda_n = \{1,2,...,n\}$ for any $n \in \mathbb{N}$. Usually, inequalities in $\mathbb{R}^m$ are understood component-wise. A feasible solution of $(MOP)$ is treated as an ideal solution if it minimizes all objective functions simultaneously. However, in practice, lowering one function could lead to an increase in another function. Therefore, efficiency takes the place of optimality in the multi-objective optimization theory. It is claimed that $x^*$ is an efficient solution of $(MOP)$ if there is not any $x \in \mathbb{R}^n$ such that $F(x) \leq F(x^*)$ and $F(x) \neq F(x^*)$ hold. We declare $x^* \in \mathbb{R}^n$ as a weak efficient solution of $(MOP)$ if there is not any $x \in \mathbb{R}^n$ such that $F(x) < F(x^*)$ holds. Any efficient solution is a weak efficient solution, but the opposite is not true. However, if every objective function is strictly convex, then every weak efficient solution is efficient. If $X^*$ is the set of all efficient solutions of the $(MOP)$, then the Pareto front of the $(MOP)$ is $F(X^*)$, which falls on the boundary of $F(\mathbb{R}^n)$. Assume that $(MOP)$ has a weak efficient solution $x^*$. Then, for every $u \in \mathbb{R}^n$, $x^*$

must fulfill the following first-order necessary condition of weak efficiency, $\max_{j\in\Lambda_m} F_j^{'}(x^*;u) \geq 0$. Any feasible point satisfying the above inequality is a critical point of $(MOP)$. If $F_j$ is convex for every $j$, then every critical point of $(MOP)$ is a weak efficient solution. Further, the critical point of $(MOP)$ is an efficient solution if each $F_j$, $j \in \Lambda_m$, is a strictly or strongly convex function.

The subdifferential is used in nonsmooth optimization in place of the gradient, which is used in smooth optimization. When it comes to nonsmooth optimization, it is crucial. The definition of the subdifferential of a convex continuous function is as follows.

**Definition 2.1.** [33] Let $h : \mathbb{R}^n \to (-\infty,\infty]$ be proper function and $x \in dom(h)$. The subdifferential of $h$ at $x$, represented by $\partial h(x)$, is defined by $\partial h(x) := \{\xi \in \mathbb{R}^n | h(y) \geq h(x) + \xi^T(y-x),\ \forall\, y \in \mathbb{R}^n\}$.

The following subdifferential characteristic is also needed in our proof.

**Lemma 2.1.** [18] Let $0 \in Co \bigcup_{j\in\Lambda_m} \partial F_j(x^*)$ for some $x^* \in \mathbb{R}^n$. Then $x^*$ is a critical point of the $(MOP)$.

## 3. A Trust Region Proximal Gradient Method for $(MOP)$

The concept of the trust region approach for composite multi-objective optimization problems is presented in this section. Using the concepts of trust region approaches, a variation of the proximal gradient method is shown for $(MOP)$. The following procedures are used to perform this derivation:

- derivation of a sub-problem to determine the direction of $(MOP)$ at iterating point $x \in \mathbb{R}^n$;
- adjust the radius of the trust region;
- lastly, an algorithm for $(MOP)$ is suggested.

3.1. **Derivation of sub-problem.** We create a sub-problem to choose an appropriate descent direction. A good approximation of each objective function is used for this sub-problem. This sub-problem meets certain favorable convex optimization criteria to make it easier to solve. Our method makes use of the following gradient-based estimate of $f_j$ at $x$ with a positive definite approximation of $\nabla^2 f_j(x)$

$$TR_{x,j}(u) := \frac{1}{2}u^T C_j(x)u + g_j(x+u) - g_j(x),$$

where $C_j(x)$ is a positive definite approximation of $\nabla^2 f_j(x)$. There is no doubt that the computational cost is decreased by using $C_j(x)$ instead of $\nabla^2 f_j(x)$. Let us define

$$TR_x(u) := \max_{j\in\Lambda_m} \nabla f_j(x)^T u + \frac{1}{2}u^T C_j(x)u + g_j(x+u) - g_j(x).$$

Since $TR_{x,j}$ is continuous in $u$ for any fixed $x$, it follows that $TR$ is also continuous in $u$. So $\partial_u TR_{x,j}(u)$ is nonempty and bounded according to [18, Theorem 2.1]. Denote

$$I_x(u) := \{j \in \Lambda_m | TR_x(u) = TR_{x,j}(u)\}.$$

From [33, Theorems 2.91 and 2.96], we obtain

$$\partial_u TR_{x,j}(u) = \left\{\nabla f_j(x) + C_j(x)u\right\} + \partial_u g_j(x+u), \quad \partial_u TR_x(u) = Co \bigcup_{j\in I_x(u)} \partial_u TR_{x,j}(u).$$

Finally, we solve the sub-problem to determine an appropriate descent direction of $(MOP)$ for any $x \in \mathbb{R}^n$

$$\begin{aligned} SP(x;\Delta):\ &\min_{u\in\mathbb{R}^n} TR_x(u) \\ &s.\,t.\ \|u\|^2 \leq \Delta^2, \end{aligned}$$

where $\Delta > 0$ is the radius of the trust region. The set of feasible solutions of $SP(x,\Delta)$ is

$$X(\Delta) = \{u \in \mathbb{R}^n |\ \|u\|^2 \leq \Delta^2\}.$$

**Note 3.1.** If $g_j = 0$ for all $j \in \Lambda_m$, then $SP(x)$ coincides with the sub-problem in [11].

If $C_j(x)$ is positive definite for every $x$, then $TR_{x,j}(u)$ is strictly convex function in $u$ for every $j \in \Lambda_m$ and $x \in \mathbb{R}^n$. Hence $TR_x(u)$ is a strictly convex function in $u$ for every $x \in \mathbb{R}^n$. This implies that $SP(x;\Delta)$ has a unique finite minimizer. Denote $u(x,\Delta) = \arg\min_{u\in X(\Delta)} TR_x(u)$ and $t(x,\Delta) = TR_x(u(x,\Delta))$. Clearly, for every $x \in \mathbb{R}^n$, $t(x,\Delta) = TR_x(u(x,\Delta)) \leq TR_x(0) = 0$. Since $u(x,\Delta)$ is the solution of $SP(x,\Delta)$, from [33, Theorem 3.1], $0 \in \partial_u TR_x(u(x,\Delta)) + N_{X(\Delta)}(u(x,\Delta))$. From [33, proposition 3.3], one has

$$N_{X(\Delta)}(u(x,\Delta)) = \begin{cases} \{0\} & if\ u(x,\Delta) \in int(X(\Delta)) \\ Cone(u(x,\Delta)) & otherwise \end{cases}$$

Suppose $u(x,\Delta) \in bd(N_{X(\Delta)}(u(x,\Delta)))$. From [33, Theorem 2.96], there exists $\lambda \in \mathbb{R}_+^{|I(x,u(x,\Delta))|}$, $\xi_j \in \partial_u g_j(x,u(x,\Delta))$ $j \in I(x,u(x,\Delta))$ and $\mu \geq 0$ such that $\sum_{j\in I(x,u(x,\Delta))} \lambda_j = 1$ and

$$\sum_{j\in I(x,u(x,\Delta))} \lambda_j \left(\nabla f_j(x) + C_j(x)u(x,\Delta) + \xi_j\right) + \mu u(x,\Delta) = 0.$$

Next, we consider any $\xi_j \in \partial_u g_j(x+u(x,\Delta))$ and substitute $\lambda_j = 0$ for all $j \notin I(x,u(x,\Delta))$. Further, substituting $\mu = 0$ if $u(x,\Delta) \in int(X(\Delta))$, we have the following first order necessary conditions:

$$\sum_{j\in\Lambda_m} \lambda_j = 1, \tag{3.1}$$

$$\sum_{j\in\Lambda_m} \lambda_j \left(\nabla f_j(x) + C_j(x)u(x,\Delta) + \xi_j\right) + \mu u(x,\Delta) = 0, \tag{3.2}$$

$$\lambda_j\left(\nabla f_j(x)^T u(x,\Delta) + \frac{1}{2}u(x,\Delta)^T C_j(x)u(x,\Delta) + g_j(x+u(x,\Delta)) - g_j(x) - t(x)\right) = 0, \quad \lambda_j \geq 0, j \in \Lambda_m, \tag{3.3}$$

$$\mu \geq 0\ \mu(\|u(x,\Delta)\|^2 - \Delta^2) = 0, \tag{3.4}$$

$$\nabla f_j(x)^T u(x,\Delta) + \frac{1}{2}u(x,\Delta)^T C_j(x)u(x,\Delta) + g_j(x+u(x,\Delta)) - g_j(x) \leq t(x)\ \ j \in \Lambda_m, \tag{3.5}$$

$$\|u(x,\Delta)\|^2 - \Delta^2 \leq 0. \tag{3.6}$$

If $u(x,\Delta)$ is the solution of $SP(x,\Delta)$ and $t(x,\Delta) = TR_x(u(x,\Delta))$, then there exists $\lambda \in \mathbb{R}^m_+$, $\mu \geq 0$ such that $(u(x,\Delta), t(x,\Delta); \lambda, \mu)$ satisfies (3.1)-(3.6).

**Lemma 3.1.** *Let $C_j(x)$ be positive definite for every $j$ and $x$. Then $u(x,\Delta) = 0$ for any $\Delta > 0$ implies that $x \in \mathbb{R}^n$ is a critical point of $(MOP)$ and the vice versa.*

*Proof.* If $u(x;\Delta) = 0$ holds for any $\Delta > 0$, then the constraint in $SP(x)$ is an inactive one. Then $SP(x)$ coincides with the sub-problem $\Phi(x)$ of [19]. Thus the result follows from [19, Lemma 3.4] immediately. □

**Theorem 3.1.** *Assume that, for any $x, u \in \mathbb{R}^n$ and $j \in \Lambda_m$, there exists a $\sigma > 0$ such that $u^T C_j(x) u \geq \sigma \|u\|^2$ holds. Suppose that $u(x,\Delta)$ is the optimal solution of $SP(x,\Delta)$. Then $t(x,\Delta) \leq -(\frac{\sigma}{2} + \mu)\|u(x,\Delta)\|^2$. If $u(x,\Delta) \neq 0$ then $u(x,\Delta)$ is a descent direction for every $F_j$.*

*Proof.* Let $t(x,\Delta) = Q(x, u(x,\Delta))$, where $u(x,\Delta)$ is the solution of $SP(x,\Delta)$. Then there exist $\lambda \in \mathbb{R}^m_+$ and $\mu \geq 0$ such that $(u(x,\Delta), t(x,\Delta); \lambda, \mu)$ satisfies (3.1)-(3.6). Since $g_j$ is convex and $\xi_j \in \partial_u g_j(x + u(x,\Delta))$,

$$g_j(x + u(x,\Delta)) - g_j(x) \leq \xi_j^T u(x,\Delta). \tag{3.7}$$

Multiplying both sides of (3.2) by $u(x,\Delta)$ yields

$$\sum_{j \in \Lambda_m} \lambda_j \left[\nabla f_j(x)^T u(x,\Delta) + u(x,\Delta)^T C_j(x) u(x,\Delta) + \xi_j^T u(x,\Delta)\right] + \mu u(x,\Delta)^T u(x,\Delta) = 0$$

From (3.7), one has

$$\sum_{j \in \Lambda_m} \lambda_j \left\{\nabla f_j(x)^T u(x,\Delta) + u(x,\Delta)^T C_j(x) u(x,\Delta) + g_j(x + u(x,\Delta)) - g_j(x)\right\} + \mu u(x,\Delta)^T u(x,\Delta) \leq 0. \tag{3.8}$$

Taking sum over $j \in \Lambda_m$ in (3.3) and using (3.1) yield

$$\begin{aligned} &\sum_{j \in \Lambda_m} \lambda_j \left\{\nabla f_j(x)^T u(x,\Delta) + u(x,\Delta)^T C_j(x) u(x,\Delta) + g_j(x + u(x,\Delta)) - g_j(x)\right\} \\ = &\sum_{j \in \Lambda_m} \lambda_j \frac{1}{2} u(x,\Delta)^T C_j(x) u(x,\Delta) + t(x,\Delta). \end{aligned} \tag{3.9}$$

Substituting (3.8) into (3.9), one has

$$t(x,\Delta) \leq -\sum_{j \in \Lambda_m} \lambda_j \frac{1}{2} u(x,\Delta)^T C_j(x) u(x,\Delta) - \mu u(x,\Delta)^T u(x,\Delta). \tag{3.10}$$

Since $u(x,\Delta)^T C_j(x) u(x,\Delta) \geq \sigma \|u(x,\Delta)\|^2$ holds for every $j$, one see from (3.10) and (3.1) that $t(x,\Delta) \leq -(\frac{\sigma}{2} + \mu)\|u(x,\Delta)\|^2$. If $u(x,\Delta) \neq 0$, then $t(,\Delta) < 0$. From (3.5),

$$\begin{aligned} \nabla f_j(x)^T u(x,\Delta) + g_j(x + u(x,\Delta)) - g_j(x) &\leq t(x,\Delta) - \frac{1}{2} u(x,\Delta)^T C_j(x) u(x,\Delta) \\ &\leq -(\sigma + \mu)\|u(x,\Delta)\|^2 \\ &< 0 \end{aligned}$$

holds for every $j \in \Lambda_m$. This implies $u(x,\Delta)$ is a descent direction for every $F_j$ at $x$. □

3.2. **Trust region radius update.** Radius of the trust region ($\Delta$) is crucial to the convergence of the trust region method. This approach requires that trust region radius ($\Delta_k$) is chosen correctly because any line search technique is not used. We derive new update formulae for the trust region radius in this subsection. For the sake of simplicity, $u(x^k, \Delta_k)$ and $t(x^k, \Delta_k)$ are represented by $u^k$ and $t^k$, respectively, in the remainder of the paper, including this subsection. At $x^k$,

$$Ared(u^k) = \min_{j \in \Lambda_m} \{F_j(x^k) - F_j(x^k + u^k)\}$$

defines the actual reduction with the aid of $u^k$. The predicted reduction based on $u^k$ is defined by

$$\begin{aligned} Pred(u^k) &:= -\max_{j \in \Lambda_m} \{\nabla f_j(x^k)^T u^k + \frac{1}{2} u^{kT} C_j(x^k) u^k + g_j(x^k + u^k) - g_j(x^k)\} \\ &= -t^k. \end{aligned}$$

Define

$$\rho(u^k) := \frac{Ared(u^k)}{Pred(u^k)} = \frac{\min_{j \in \Lambda_m} \{F_j(x^k) - F_j(x^k + u^k)\}}{-t^k}.$$

With the use of scalars $0 < \sigma_3 < 1 < \sigma_1$, $0 < \sigma_0 < \sigma_2 < 1$, $\Delta_{k+1}$ is either enlarged, shrunk, or left untouched in the subsequent iteration under the following conditions

- $\rho(u^k) \geq \sigma_2$ indicates that $Ared(u^k)$ and $Pred(u^k)$ are in good agreement. The trust region radius can be expanded for the following iteration in this situation. In the proposed method, $\Delta_{k+1} = \max\{\sigma_1 \Delta_k, \Delta_{min}\}$ is used to update the trust region radius, where $\Delta_{min}$ is prespecified.
- It is implied that $u^k$ is descent for every $j$ by $\sigma_0 \leq \rho(u^k) < \sigma_2$, however there is poor agreement between $Ared(u^k)$ and $Pred(u^k)$. The trust region radius stays the same in this instance.
- As implied by $\rho(u^k) < \sigma_0$, $u^k$ does not sufficiently decrease at least one $j$. Here, the radius of the trust region is reduced by $\Delta_k = \sigma_3 \Delta_k$ trust region, and $SP(x^k, \Delta_k)$ is solved once more. Continue doing this until $\rho(u^k) \geq \sigma_0$ holds.

3.3. **Algorithm.** Now, we introduce an algorithm for $(MOP)$ in this subsection. A sequence is generated by using the update formula $x^{k+1} = x^k + u^k$, where $u^k$ is the optimal solution of $SP(x^k, \Delta_k)$, along with some initial approximation $x^0$, initial positive definite matrices $C_j(x^0)$, and initial trust region radius $\Delta_0$. The modified BFGS updated formula in [34] is used to update $C_j(x^{k+1})$ for $x^{k+1}$, and the procedures outlined in the preceding subsections are followed to update $\Delta_{k+1}$. Until we obtain an approximated critical point $(MOP)$, this process is repeated.

**Algorithm 3.1.** (Trust region proximal gradient method for $(MOP)$)

Step 1 Choose initial approximation $x^0$, initial positive definite matrix $C_j(x^0)$ for $j \in \Lambda_m$, scalars $0 < \sigma_3 < 1 < \sigma_1$, $0 < \sigma_0 < \sigma_2 < 1$, initial trust region radius $\Delta_0$ and $\Delta_{min} > 0$. Set $k := 0$.
Step 2 Solve the sub-problem $SP(x^k, \Delta_k)$ to find $u^k$. Then compute $t^k = TR_{x^k}(u^k)$.
Step 3 If $\|u^k\| < \varepsilon$, then stop. Else go to Step 4.
Step 4 Compute

$$\rho(u^k) := \frac{\min_{j \in \Lambda_m} \{F_j(x^k) - F_j(x^k + u^k)\}}{-t^k}$$

If $\rho(u^k) < \sigma_0$, we set $\Delta_k = \sigma_3\Delta_k$ and go to Step 2. If $\rho(u^k) \geq \sigma_0$, then we update

$$\begin{aligned} x^{k+1} &= x^k + u^k \\ \Delta_{k+1} &= \begin{cases} \max\{\sigma_1\Delta_k, \Delta_{min}\} & \text{if } \rho(u^k) \geq \sigma_2 \\ \Delta_k & \text{otherwise} \end{cases} \end{aligned}$$

and go to Step 5.

Step 5 Generate $C_j(x^{k+1})$ for every $j \in \Lambda_m$ via the modified BFGS update formula in [34]. Set $k := k+1$ and go to Step 2.

## 4. Convergence Analysis

Under a few minor assumptions, the global convergence of Algorithm 3.1 is justified in this section. $\theta(x;\Delta)$, defined as

$$\theta(x;\Delta) := \min_{\|u\|\leq\Delta} \left[ \max_{j\in\Lambda_m} \left\{ \nabla f_j(x)^T u + g_j(x+u) - g_j(x) \right\} \right] \tag{4.1}$$

is utilized to support the global convergence.

**Lemma 4.1.** *Consider $\theta(x;\Delta)$ defined by (4.1). Then*

*(i) $\theta(x;\Delta) \leq 0$ for every $x \in \mathbb{R}^n$ and $\Delta > 0$.*
*(ii) $\theta(x;\bar{\Delta}) \leq \theta(x;\Delta)$ for any $\bar{\Delta} \geq \Delta$.*
*(iii) $x^*$ is a critical point of $(MOP)$ if and only if $\theta(x;\Delta) = 0$ for some $\Delta > 0$.*
*(iv) $\theta(x;\alpha\Delta) \leq \alpha\theta(x;\Delta)$ for any $\alpha \in [0,1]$.*

**Lemma 4.2.** *Suppose that there exists $b > 0$ such that $u^T C_j(x^k) u \leq b\|u\|^2$ holds for every $j \in \Lambda_m$ and $u \in \mathbb{R}^n$. Then, for any $\Delta \geq \Delta_k$,*

$$t^k \leq \frac{\theta(x^k;\Delta)}{2\Delta} \min\left\{ \Delta_k, \frac{-\theta(x^k;\Delta)}{b\Delta} \right\} \tag{4.2}$$

*Proof.* Let $\bar{u}$ be the minima of $\theta(x^k,\Delta_k)$ in (4.1). Then

$$\begin{aligned} t(x^k;\Delta_k) &= \min_{\|u\|\leq\Delta_k} \max_{j\in\Lambda_m} \left\{ \nabla f_j(x^k)^T u + \frac{1}{2} u^T C_j(x^k) u + g_j(x^k+u) - g_j(x^k) \right\} \\ &\leq \min_{0\leq\alpha\leq1} \max_{j\in\Lambda_m} \left\{ \nabla f_j(x^k)^T(\alpha\bar{u}) + \frac{\alpha^2}{2} \bar{u}^T C_j(x^k)\bar{u} + g_j(x^k+\alpha\bar{u}) - g_j(x^k) \right\} \\ &\leq \min_{0\leq\alpha\leq1} \max_{j\in\Lambda_m} \left\{ \alpha \left[ \nabla f_j(x^k)^T \bar{u} + g_j(x^k+\bar{u}) - g_j(x^k) \right] + \frac{b\alpha^2}{2}\|\bar{u}\|^2 \right\} \\ &\leq \min_{0\leq\alpha\leq1} \left\{ \alpha\theta(x^k;\Delta_k) + \frac{b\alpha^2}{2}\|\bar{u}\|^2 \right\} \\ &\leq \max\left\{ \frac{\theta(x^k;\Delta_k)}{2}, \frac{-\theta(x^k;\Delta_k)^2}{2b\|\bar{u}\|^2} \right\} \end{aligned} \tag{4.3}$$

Now, for any $\Delta > \Delta_k$,

$$\theta(x^k;\Delta_k) = \theta(x^k;\frac{\Delta_k}{\Delta}\Delta) \leq \frac{\Delta_k}{\Delta}\theta(x^k;\Delta). \tag{4.4}$$

The last inequality follows from Lemma 4.1(iv) since $\frac{\Delta_k}{\Delta} < 1$. Further, both $\theta(x^k;\Delta_k)$ and $\theta(x^k;\Delta)$ are negative. From (4.4), we have

$$\theta(x^k;\Delta_k)^2 \geq \frac{\Delta_k^2}{\Delta^2}\theta(x^k;\Delta)^2 \geq \frac{\|\bar{u}\|^2}{\Delta^2}\theta(x^k;\Delta)^2 \ (\because\ \Delta_k \geq \|\bar{u}\|)$$

This implies

$$\frac{-\theta(x^k;\Delta_k)^2}{2b\|\bar{u}\|^2} \leq \frac{-\theta(x^k;\Delta)^2}{2b\Delta^2}. \tag{4.5}$$

Substituting (4.4) and (4.5) into (4.3) yields

$$t(x^k;\Delta_k) \leq \max\left\{\frac{\Delta_k}{2\Delta}\theta(x^k;\Delta), \frac{-\theta(x^k;\Delta)^2}{2b\Delta^2}\right\} = \frac{\theta(x^k;\Delta)}{2\Delta}\min\left\{\Delta_k, \frac{-\theta(x^k;\Delta)}{b\Delta}\right\}.$$

Last equality holds since $\theta(x^k;\Delta) < 0$. Hence, the lemma follows. □

In the following lemma, we show that the inner circle (Step 2- Step 4 -Step 2) is repeated finitely.

**Lemma 4.3.** *Let the assumptions of Lemma 4.2 be true. Then, for any non critical point $x^k$, $\rho_k \geq \sigma_0$ holds after finite repetition of inner circle (Step 2- Step 4 -Step 2) in Algorithm 3.1.*

*Proof.* We may suppose that the inner circle inner circle (Step 2- Step 4 -Step 2) is repeated infinitely a non critical point $x^k$. Then

$$\rho_k(u^{k,r}) < \sigma_0 \text{ and } \Delta_{k,r+1} = \sigma_3\Delta_{k,r} \tag{4.6}$$

hold for $r = 1,2,\ldots$, where index $r$ denotes the number of repetition of inner circle and $u^{k,r}$ is the solution of $SP(x^k,\Delta_{k,r})$ . Since $0 < \sigma_3 < 1$, (4.6) implies $\lim\limits_{r\to\infty}\Delta_{k,r} = 0$ and $\lim\limits_{r\to\infty}\|u^{k,r}\| = 0$. This together with (4.2) implies that there exist $\bar{r}$ and $\Delta > 0$ such that, for all $r \geq \bar{r}$ and $\Delta \geq \Delta_{k,r}$,

$$t^{k,r} \leq \frac{\theta(x^k,\Delta))\Delta_{k,r}}{\Delta} \leq -\delta\Delta_{k,r} \tag{4.7}$$

where $t^{k,r} := t(x^k,\Delta_{k,r})$. Since $C_j(x^k) \approx \nabla^2 f_j(x^k)$, there exists $\omega_j(x^k;u^{k,r})$ such that

$$f_j(x^k + u^{k,r}) = f_j(x^k) + \nabla f_j(x^k)^T u^{k,r} + \frac{1}{2}u^{k,r^T}C_j(x^k)u^{k,r} + \omega_j(x^k;u^{k,r})$$

and $\frac{\omega_j(x^k;u^{k,r})}{\|u^{k,r}\|^2} \longrightarrow 0$ as $\|u^{k,r}\|^2 \longrightarrow 0$ for all $j$. The above equality implies

$$\begin{aligned}&f_j(x^k + u^{k,r}) + g_j(x^k + u^{k,r}) - f_j(x^k) - g_j(x^k)\\ &= \nabla f_j(x^k)^T u^{k,r} + \frac{1}{2}u^{k,r^T}C_j(x^{k,r})u^{k,r} + g_j(x^k + u^{k,r}) - g_j(x^k) + \omega_j(x^k;u^{k,r})\end{aligned}$$

Define $\omega(x^k;u^{k,r}) := \max\limits_{j\in\Lambda_m}\omega_j(x^k;u^{k,r})$. Then $\frac{\omega(x^k;u^{k,r})}{\|u^{k,r}\|^2} \longrightarrow 0$ as $\|u^{k,r}\|^2 \longrightarrow 0$. Further, we have

$$\begin{aligned}\max_{j\in\Lambda_m}\left\{F_j(x^k + u^{k,r}) - F_j(x^k)\right\} \leq \max_{j\in\Lambda_m}\Big\{&\nabla f_j(x^k)^T u^{k,r} + \frac{1}{2}u^{k,r^T}C_j(x^k)u^{k,r}\\ &+ g_j(x^k + u^{k,r}) - g_j(x^k)\Big\} + \omega(x^k;u^{k,r}),\end{aligned}$$

which implies

$$-\min_{j\in\Lambda_m}\left\{F_j(x^k)-F_j(x^k+u^{k,r})\right\}-t(x^{k,r})\leq\omega(x^k;u^{k,r}) \tag{4.8}$$

Note that $\rho(u^{k,r})<1$ implies $\min\limits_{j\in\Lambda_m}\{F_j(x^k)-F_j(x^k+u^{k,r})\}+t(x^{k,r})<0$. Hence, from (4.8), one has $|\min\limits_{j\in\Lambda_m}F_j(x^k)-F_j(x^k+u^{k,r})+t(x^{k,r})|\leq\omega(x^k;u^{k,r})$. This along with (4.7) implies

$$\frac{\left|\min\limits_{j\in\Lambda_m}\ \{F_j(x^k)-F_j(x^k+u^{k,r})\}+t(x^{k,r})\right|}{-t(x^{k,r})}\leq\frac{\omega(x^k;u^{k,r})}{\delta\Delta_{k,r}}. \tag{4.9}$$

Since $t(x^{k,r})<0$, one has

$$\begin{aligned}\frac{\left|\min\limits_{j\in\Lambda_m}\ \{F_j(x^k)-F_j(x^k+u^{k,r})\}+t(x^{k,r})\right|}{-t(x^{k,r})} &= \left|\frac{\min\limits_{j\in\Lambda_m}\ \{F_j(x^k)-F_j(x^k+u^{k,r})\}+t(x^{k,r})}{-t(x^{k,r})}\right|\\ &= \left|\frac{\min\limits_{j\in\Lambda_m}\ \{F_j(x^k)-F_j(x^k+u^{k,r})\}}{-t(x^{k,r})}-1\right|\\ &= \left|\rho(u^{k,r})-1\right|,\end{aligned}$$

which together with (4.9) yields

$$\left|\rho(u^{k,r})-1\right|\leq\frac{1}{\delta}\frac{\omega(x^k;u^{k,r})}{\|u^{k,r}\|^2}\frac{\|u^{k,r}\|^2}{\Delta_{k,r}^2}\Delta_{k,r}\leq\frac{1}{\delta}\frac{\omega(x^k;u^{k,r})}{\|u^{k,r}\|^2}\Delta_{k,r}. \tag{4.10}$$

The last inequality follows from $\|u^{k,r}\|\leq\Delta_{k,r}$. Since $\lim\limits_{r\to\infty}\frac{\omega(x^k;u^{k,r})}{\|u^{k,r}\|^2}=0$ and $\lim\limits_{r\to\infty}\Delta_{k,r}=0$, there exists $0<\delta_1<1$ such that $\frac{1}{\delta}\frac{\omega(x^k;u^{k,r})}{\|u^{k,r}\|^2}\Delta_{k,r}\leq\delta_1$ holds for every $r$ sufficiently large. Then (4.10) implies that $\rho(u^{k,r})\geq 1-\delta_1$ holds for every $r$ sufficiently large. This contradicts that inner circle is repeated infinitely. □

Finally, we justify the global convergence of Algorithm 3.1 in the following theorem under some mild assumptions.

**Theorem 4.1.** *Let $\{x^k\}$ be defined by Algorithm 3.1. Let the assumptions of Lemma 4.2 hold and level set $L=\{x:F(x)\leq F(x^0)\}$ be bounded. Then either $\{x^k\}$ ends at a critical point of $(MOP)$ or any accumulation point of $\{x^k\}$ is a critical point of $(MOP)$.*

*Proof.* Since inner circle (Step 2- Step 4 -Step 2) is repeated finitely, Algorithm 3.1 terminates at $x^k$ if and only if $u^k=0$. From (3.4), $\mu_k=0$. Substituting $u^k=0$ and $\mu_k=0$ into (3.1) and (3.2), we have

$$\sum_{j\in\Lambda_m}\lambda_j^k\left(\nabla f_j^k+\xi_j^k\right)=0,$$

where $\xi_j^k\in\partial g_j(x^k)$, $\lambda_j^k\geq 0$, and $\sum\limits_{j\in\Lambda_m}\lambda_j=1$. This implies $0\in Co\big(\bigcup\limits_{j\in\Lambda_m}\partial F_j(x^k)\big)$. Hence from Lemma 2.1, $x^k$ is a critical point of $(MOP)$.

Next, we suppose that Algorithm 3.1 generates an infinite sequence. Since $\rho(u^k) \geq \sigma_0$ holds for every $k$, we have

$$-\sigma_0 t^k \leq \min_{j\in\Lambda_m} \left\{F_j^k - F_j^{k+1}\right\} \leq \min_{j\in\Lambda_m} \left\{F_j^{k-1} - F_j^{k+1}\right\} - \min_{j\in\Lambda_m} \left\{F_j^{k-1} - F_j^{k}\right\}$$
$$\leq \min_{j\in\Lambda_m} \left\{F_j^{k-1} - F_j^{k+1}\right\} + \sigma_0 t^{k-1},$$

which implies $-\sigma_0 t^k - \sigma_0 t^{k-1} \leq \min_{j\in\Lambda_m} \left\{F_j^{k-1} - F_j^{k+1}\right\}$. Proceeding in a similar way, we have

$$\sigma_0 \sum_{i=0}^{k} (-t^i) \leq \min_{j\in\Lambda_m} \left\{F_j^0 - F_j^{k+1}\right\}.$$

Note that $x^k$ is bounded. There exists $F_j^* > -\infty$ for all $j$ such that

$$\sigma_0 \sum_{i=0}^{\infty} (-t^i) \leq \min_{j\in\Lambda_m} \left\{F_j^0 - F_j^{*}\right\} < \infty,$$

which implies $t^k \to 0$ as $k \to \infty$ as $-t^k > 0$ for all $k$. Then $\|u^k\| \to 0$ as $k \to \infty$, since $\mu^k \geq 0$ for all $k$. Note that there exists at least one convergent subsequence. Let $\{x^k\}_{k\in K_0}$ be a converging subsequence of $x^k$, converging to $x^*$. Next, we show that $x^*$ is a critical point of $(MOP)$. Since $(t(x^k,\Delta_k), u(x^k,\Delta_k))$ is the solution of $SP(x^k,\Delta_k)$, there exists $(\lambda^k, \mu_k)$ satisfying (3.1)-(3.6). From (3.1), one sees that $\{\lambda^k\}$ is bounded. So there exists a subsequence of $\{\lambda^k\}$ converging to $\lambda^*$. Without loss of generality, we may assume $\lambda^k \to \lambda^*$ as $k \to \infty$. Then $\lambda_j^* \geq 0$ for all $j$ and $\sum_{j\in\Lambda_m} \lambda_j^* = 1$ (from (3.1)). Now $u(x^k) \to 0$ and $\Delta_k \geq \Delta > 0$ imply $\mu_k = 0$ for large $k$. Similarly, $\|C_j(x^k)\| \leq b$ implies $\|C_j(x^k)u^k\| \leq \|C_j(x^k)\|\|u^k\| \leq b\|u^k\|$. This along with $u^k \to \infty$ as $k \to \infty$ implies $C_j(x^k)u^k \to 0$ as $k \to \infty$. Taking limit $k \to \infty$ in (3.1) and (3.2), we have

$$\sum_{j\in\Lambda_m} \lambda_j^* = 1, \ \sum_{j\in\Lambda_m} \lambda_j^* (\nabla f_j(x^*) + \xi_j^*) = 0,$$

which implies $0 \in Co\big(\bigcup_{j\in\Lambda_m} \partial F_j(x^*)\big)$. From Lemma 2.1, one concludes that $x^*$ is a critical point of $(MOP)$. Since $\{x^k\}_{k\in K_0}$ is an arbitrary convergent subsequence of $\{x^k\}$, then any accumulation point of $\{x^k\}$ is a critical point of $(MOP)$. □

## 5. Numerical Examples

This section uses a series of problems to verify and compare the proximal gradient method for $(MOP)$ proposed in [18] (MONPG) and [20] (MOPG) with Algorithm 3.1 (MOTRPG). Below is an explanation of how these algorithms are implemented.

- For every technique, MATLAB (2024a) code is created using the '*CVX*' extension. Each technique uses '*CVX*' programming with solver '*SeDuMi*' to solve the sub-problems.
- $\sigma_0 = 0.01$, $\sigma_1 = 1.5$, $\sigma_2 = 0.5$, $\sigma_3 = 0.5$, and $\Delta_{min} = \max\{\min_{j\in\Lambda_m} \|f_j(x^0)\|, 1\}$ were utilised in (MOTRPG).
- The terminating criterion is $\|u^k\| < 10^{-5}$ or a maximum of 2000 iterations.
- Solution of a multi-objective optimization problem is not isolated optimum points but a set of efficient solutions. We have taken into consideration the multi-start technique

to produce an approximate set of efficient solutions. For each technique, the following actions are taken.

- With $lb, ub \in \mathbb{R}^n$ and $lb < ub$, a set of 100 uniformly distributed random beginning points between $lb$ and $ub$ are taken into consideration.
- The Algorithm of each method is executed individually.
- Assume that the set of approximation critical points is $\mathscr{W}\mathscr{X}^*$. An approximate set of efficient solutions is the non-dominated set of $\mathscr{W}\mathscr{X}^*$.

Next, we demonstrate the steps of Algorithm 3.1 using the following example.

**Example 5.1.** Consider the bi-objective optimization problem:

$$(E_1): \min_{x\in\mathbb{R}^2} \; (f_1(x)+g_1(x), f_2(x)+g_2(x))$$

where

$$\begin{aligned} f_1(x) &:= x_1^2+x_2^2, \; f_2(x) := (x_1-5)^2+(x_2-5)^2, \\ g_1(x) &:= \max\left\{(x_1-2)^2+(x_2+2)^2, x_1^2+8x_2\right\}, \; g_2(x) := \max\left\{5x_1+x_2, x_1^2+x_2^2\right\}. \end{aligned}$$

Here $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}$ follows from test problem BK1 [35] and $f_1, f_2 : \mathbb{R}^2 \to \mathbb{R}$ follows from the example from [22].

Considering $x^0 = (-4.5, 6.5)^T$, one has $F(x^0) = (177, 155)^T$. Using

$$\Delta_0 = \min\{\|\nabla f_1(x^0)\|^2, \|\nabla f_1(x^0)\|^2\} = 15.8114,$$

the solution of $SP(x^0;\Delta_0)$ is obtained as $u^0 = (3.4524, -1.9728)^T$ and $t^0 = -104.5165$. Then $\rho(u^0) = 0.9244 > \sigma_2$. So we calculate next iterating point as $x^1 = x^0 + u^0 = (1.1667, -1.8333)^T$ and expand the trust region for next iteration as $\Delta_1 = \max\{\sigma_3\Delta_0, \Delta_{min}\} = 18.9737$. One can observe that $F(x^1) = (73.4843, 58.3892)^T < F(x^0)$. Using the stopping criteria $\|u^k\| < 10^{-4}$, the approximate solution is obtained as $x^3 = (2.0077, 3.0015)^T \approx (2,3)^T$. We justify that $x^* = (2,3)^T$ is a critical $(MOP)$ point. One can observe that both $x_1^2+8x_2$ is active function in $g_1(x^*)$ and both $5x_1+x_2$ and $x_1^2+x_2^2$ are active functions in $g_2(x^*)$. From [33, Theorem 2.96], one sees that $\partial g_1(x^*) = \{(4,8)^T\}$ and $\partial g_2(x^*) = Co\{(5,1)^T, (4,6)^T\}$. This implies

$$\begin{aligned} \partial F_1(x^*) &= \{\nabla f_1(x^*)\} + \partial g_1(x^*) = \{(8,14)T\} \\ \partial F_2(x^*) &= \{\nabla f_2(x^*)\} + \partial g_2(x^*) = Co\{(-1,-3)^T, (-2,2)^T\}. \end{aligned}$$

Then convex hull of $\{\partial F_1(x^*), \partial F_2(x^*)\}$ is the triangle formed by $(8,14)$, $(-1,-3)$, and $(-2,2)$. From Figure 1, we can see that $(0,0)$ lies inside the triangle. Hence, $0 \in Co\{\partial F_1(x^*), \partial F_2(x^*)\}$. This implies that $x* = (2,3)^T$ is a critical point of Example 1. Approximate Pareto front of $(E_1)$ obtained by using multi-start technique with MONPG and MOPG is provided in Figure 2. One can observe that $F(x^*)$ belongs to this approximate Pareto front.

In many areas of data science, least squares problems are crucial.

**Example 5.2.** **$l_1$-regularized multi-objective least square problem (MOLS):**

$$\min_{x\in\mathbb{R}^n} \left(\frac{1}{2}\|A_1x-b_1\|^2, \frac{1}{2}\|A_2x-b_2\|^2, \ldots, \frac{1}{2}\|A_mx-b_m\|^2\right)$$

is the form of a multi-objective least square problem, where, for every $j \in \Lambda_m$, $A_j \in \mathbb{R}^{m\times n}$ and $C_j \in \mathbb{R}^m$. Nonetheless, a sparse solution is sought in many least squares situations. In these situations, the main objective is increased by a multiple of the $l_1$ norm of $x$ to solve a regularised

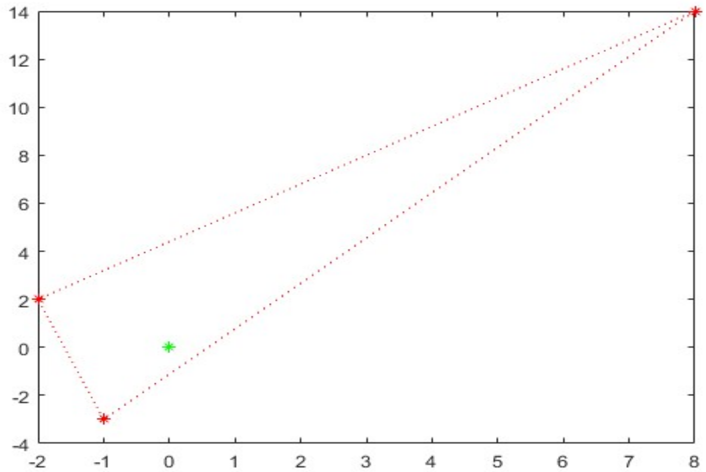



FIGURE 1. Convex hull of $\{\partial F_1(x^*), \partial F_2(x^*)\}$

.

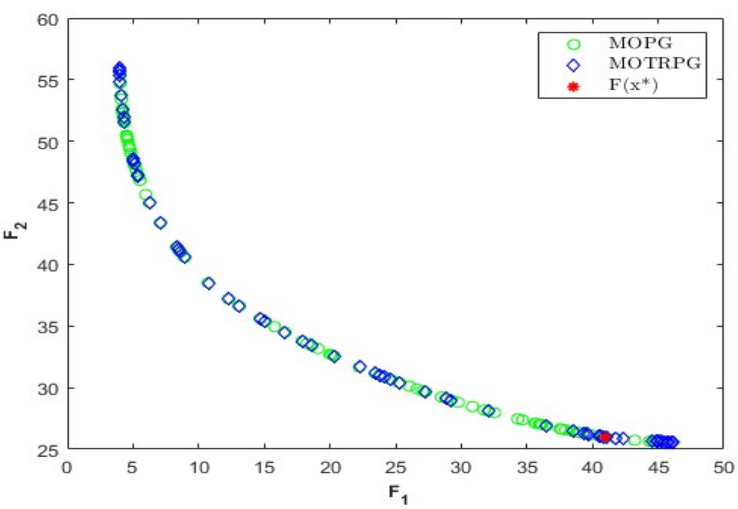



FIGURE 2. Approximate Pareto fronts of $P_1$

form of the least squares problem. The regularized multi-objective least squares problem may be expressed as follows:

$$\min_{x\in\mathbb{R}^n} \left( \frac{1}{2}\|A_1x-b_1\|^2 + \frac{\nu_1}{2}\|x\|_1, \frac{1}{2}\|A_2x-b_2\|^2 + \frac{\nu_2}{2}\|x\|_1, \dots, \frac{1}{2}\|A_mx-b_m\|^2 + \frac{\nu_m}{2}\|x\|_1 \right),$$

where regularized parameters are $\mu_j > 0$; see [36] for more information. Multi-objective least square problems can be solved via Algorithm 3.1 using

$$f_j(x) = \frac{1}{2}\|A_jx-b_j\|^2 \tag{5.1}$$

$$g_j(x) = \frac{\nu_j}{2}\|x\|_1. \tag{5.2}$$

We have constructed a three objective optimization problem

$$\min_{x\in\mathbb{R}^3} \left(\frac{1}{2}\|A_1x-b_1\|^2+\frac{\nu_1}{2}\|x\|_1, \frac{1}{2}\|A_2x-b_2\|^2+\frac{\nu_2}{2}\|x\|_1, \frac{1}{2}\|A_2x-b_2\|^2+\frac{\nu_3}{2}\|x\|_1\right),$$

where entries of $(A_j) \in \mathbb{R}^{10\times 3}$ and $b_j \in \mathbb{R}^{10}$ are generated using uniform distribution in $[0,5]$ and $[0,10]$ respectively. For nonsmooth functions, we consider $\nu_1 = 0.30$, $\nu_2 = 1.06$, and $\nu_3 = 1.84$. A set of 100 initial approximation uniformly distributed in $(-1,-1,-1)^T$ and $(1,1,1)^T$ is used to find approximate Pareto front. Approximate Pareto front generated by $(MOTRPG)$ and $(MOPG)$ if provided in Figure 3.

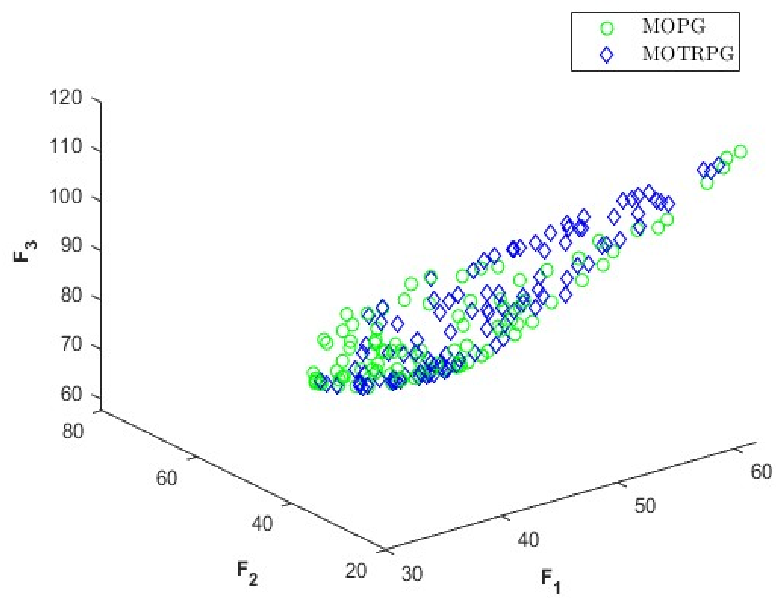



FIGURE 3. Approximate Pareto fronts of (MOLS)

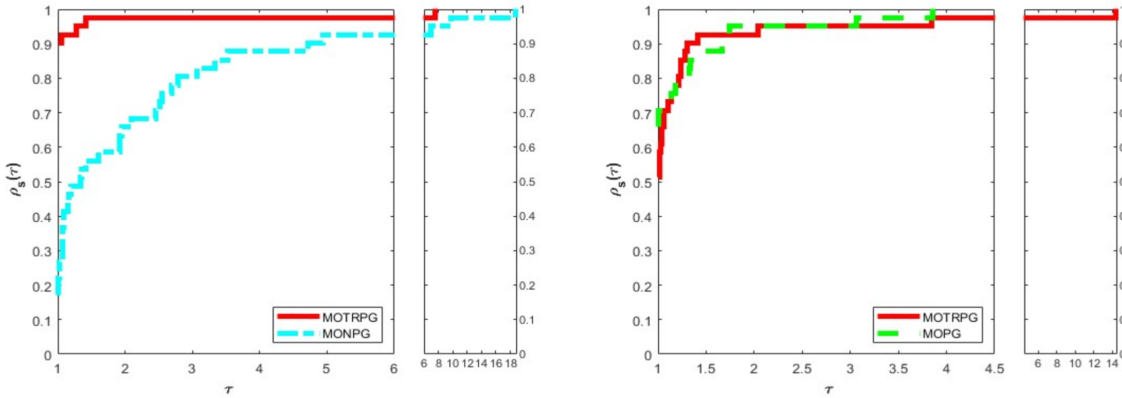
MOTRPG is compared with MONPG and MOPG. Each algorithm is executed on a set of test problems. Smooth functions $(f_j)$ of the test problems are gathered from different sources. In some problems, nonsmooth functions $(g_j)$ are gathered from [18] ($gA$, $gB$, $gC$, and $gF$ are from [18, Section 5]). Rest problems use $l_1$ regularization in (5.2) with $\nu \in \mathbb{R}^m$ uniformly generated in $0^m$ and $2^m$. Details of test problems are provided in Table 1. In this table, $m$ and $n$ denote the number of objective functions and the dimension of $x$, respectively. A set of 100 initial points uniformly distributed between $lb$ and $ub$ is considered to generate an approximate Pareto front.

Performance profiles using 'purity metric', '$\Delta$ and $\Gamma$ spread metrics', 'hypervolome metric', and 'total function evaluations' are used to compare different methods. Detailed explanation of performance profile as well as 'purity metric', '$\Delta$ and $\Gamma$ spread metrics', and 'hypervolome metric' are provided in [8, 9, 10, 18]. Total function evaluation of an algorithm for a test problem is based on total function, gradient, and Hessian (if required) evaluations. Let $\#f$, $\#\nabla f$, and $\#\nabla^2 f$ denote the number of functions, gradient, and Hessian evaluations required to solve a test problem. We use the forward difference formula to compute the approximate $\nabla f(x)$

| Sl. No | $(m,n)$ | $f$ | $g$ | $lb^T$ | $ub^T$ |
|---|---|---|---|---|---|
| 1 | (2,2) | Ansary1 ([18]) | $gA$ | $(-3,-3)$ | $(7,7)$ |
| 2 | (2,2) | Ansary1 ([18]) | $gB$ | $(-3,-3)$ | $(7,7)$ |
| 3 | (2,2) | AP2 ([7]) | gA | $(-5,-5)$ | $(5,5)$ |
| 4 | (2,2) | AP2 ([7]) | gB | $(-5,-5)$ | $(5,5)$ |
| 5 | (2,2) | BK1 ([35]) | gA | $(-5,-5)$ | $(7.5,7.5)$ |
| 6 | (2,2) | BK1 ([35]) | gB | $(-5,-5)$ | $(7.5,7.5)$ |
| 7 | (3,3) | FDS ([6]) | gC | $(-2,-2,-2)$ | $(4,4,4)$ |
| 8 | (3,5) | FDS ([6]) | (5.2) | $(-2,\dots,-2)$ | $(2,\dots,2)$ |
| 9 | (3,8) | FDS ([6]) | (5.2) | $(-2,\dots,-2)$ | $(2,\dots,2)$ |
| 10 | (3,2) | IKK1 ([35]) | (5.2) | $(-2,-2)$ | $(3,3)$ |
| 11 | (2,2) | Jin1 ([37]) | gA | $(-3,-3)$ | $(5,5)$ |
| 12 | (2,2) | Jin1 ([37]) | gB | $(-3,-3)$ | $(5,5)$ |
| 13 | (2,4) | Jin1 ([37]) | gF | $(-5,\dots,-5)$ | $(10,\dots,10)$ |
| 14 | (2,10) | Jin1 ([37]) | (5.2) | $(-5,\dots,-5)^T$ | $(5,\dots,5)^T$ |
| 15 | (2,2) | Lovison1 ([38]) | gA | $(-3,-3)$ | $(5,5)$ |
| 16 | (2,2) | Lovison1 ([38]) | gB | $(-3,-3)$ | $(5,5)$ |
| 17 | (2,2) | Lovison4 ([38]) | gB | $(-10,-5)$ | $(5,5)$ |
| 18 | (2,2) | LRS1 ([35]) | gB | $(-50,-50)$ | $(50,50)$ |
| 19 | (2,2) | LRS1 ([35]) | gA | $(-50,-50)$ | $(50,50)$ |
| 20 | (3,3) | MOLS1 (5.1) | (5.2) | $(-1,-1,-1)$ | $(1,1,1)$ |
| 21 | (3,1) | MHHM1 ([35]) | (5.2) | $-4$ | $4$ |
| 22 | (3,2) | MHHM2 ([35]) | (5.2) | $(-4,-4)$ | $(4,4)$ |
| 23 | (2,1) | MOP1 ([35]) | (5.2) | $-100$ | $100$ |
| 24 | (3,2) | MOP7 ([35]) | (5.2) | $(-4,-4)$ | $(4,4)$ |
| 25 | (2,2) | MS1([39]) | gA | $(-2,-2)$ | $(2,2)$ |
| 26 | (2,2) | MS1 ([39]) | gB | $(-2,-2)$ | $(2,2)$ |
| 27 | (2,4) | MS2([39]) | gG | $(-2,\dots,-2)$ | $(2,\dots,2)$ |
| 28 | (2,10) | MS2 ([39]) | (5.2) | $(-2,\dots,-2)$ | $(2,\dots,2)$ |
| 29 | (3,3) | SDD1 ([40]) | gC | $(-2,-2,-2)$ | $(2,2,2)$ |
| 30 | (3,10) | SDD1 ([40]) | (5.2) | $(-2,\dots,-2)$ | $(2,\dots,2)$ |
| 31 | (2,2) | SP1 ([35]) | gA | $(-1,-1)$ | $(5,5)$ |
| 32 | (2,2) | SP1 ([35]) | gB | $(-1,-1)$ | $(5,5)$ |
| 33 | (2,2) | SSFY1 ([35]) | gA | $(-50,-50)$ | $(50,50)$ |
| 34 | (2,2) | SSFY1 ([35]) | gB | $(-50,-50)$ | $(50,50)$ |
| 35 | (3,2) | VFM1 ([35]) | (5.2) | $(-2,-2)$ | $(2,2)$ |
| 36 | (2,2) | VU1 ([35]) | gA | $(-3,-3)$ | $(3,3)$ |
| 37 | (2,2) | VU1 ([35]) | gB | $(-3,-3)$ | $(3,3)$ |
| 38 | (2,2) | VU2 ([35]) | gA | $(-3,-3)$ | $(3,3)$ |
| 39 | (2,2) | VU2([35]) | gB | $(-3,-3)$ | $(3,3)$ |
| 40 | (3,3) | ZLT1 ([35]) | gC | $(-100,-100,-100)$ | $(100,100,100)$ |
| 41 | (3,10) | ZLT1 ([35]) | (5.2) | $(-5,\dots,-5)$ | $(5,\dots,5)$ |

TABLE 1. Details of test problems

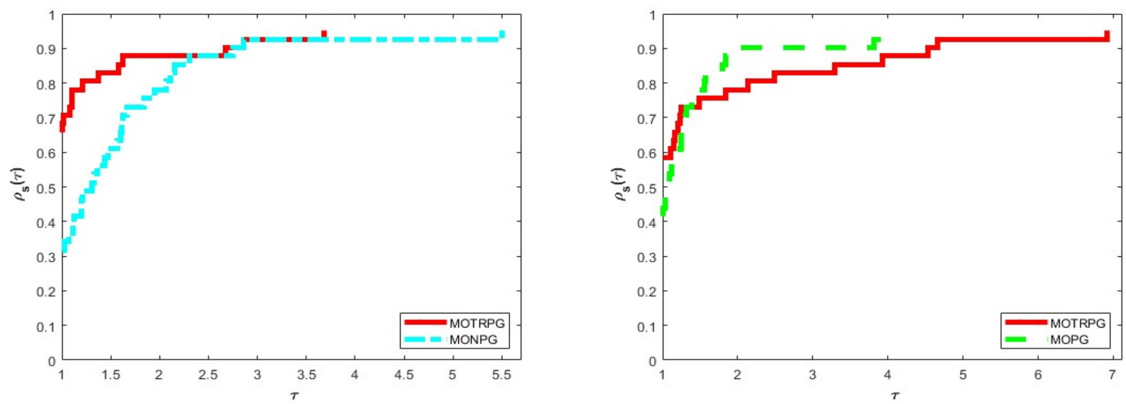
and $\nabla^2 f(x)$. For MOTRPG and MOPG, total function evaluations are $\#fun = \#f + n * \#\nabla f$. For MONPG, total function evaluation is $\#fun = \#f + n * \#\nabla f + \frac{n*(n+1)}{2} * \nabla^2 f$. Performance profile between MOTRPG and MONPG using purity matrix is provided in Figure 4a. Performance profile between MOTRPG and MOPG using purity matrix is provided in Figure 4b. Figure 5a and 5b represent the performance profiles using the $\Gamma$-spread metric. Similarly, performance profiles using $\Delta$-spread metric are provided in Figure 6a and 6b. Figure 7a and 7b represent the performance profiles using the hypervolume metric. Performance profiles using function evaluations are provided in Figures 8a and 8b.




(A) Between MOTRPG and MONPG

(B) Between MOTRPG and MOPG

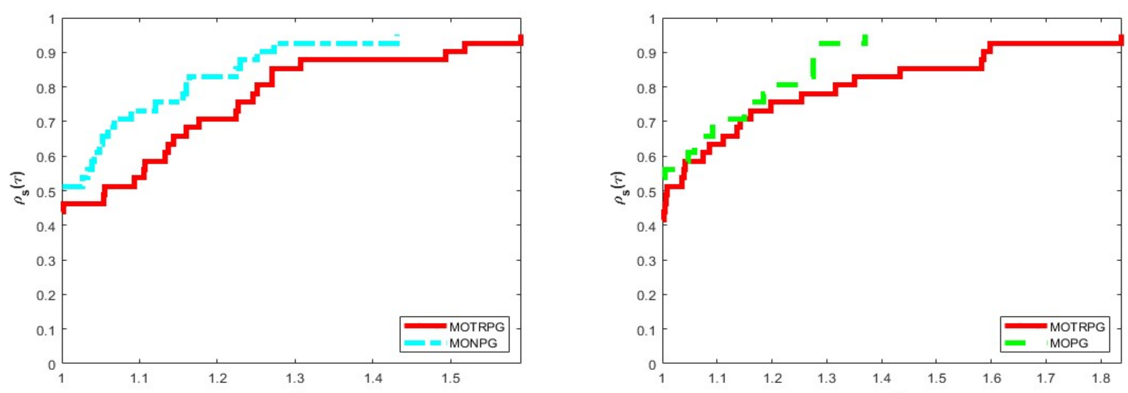
FIGURE 4. Performance profiles using purity metric




(A) Between MOTRPG and MONPG

(B) Between MOTRPG and MOPG

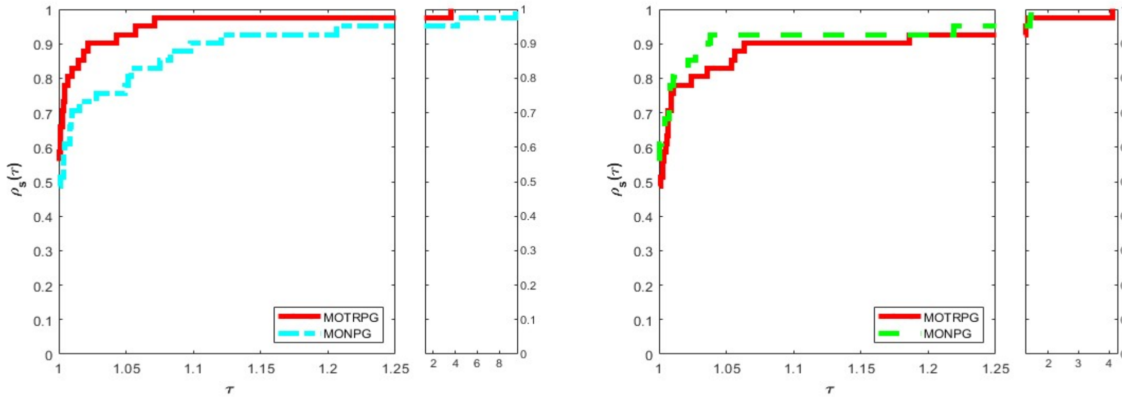
FIGURE 5. Performance profiles using Γ-spread metric




(A) Between MOTRPG and MONPG

(B) Between MOTRPG and MOPG

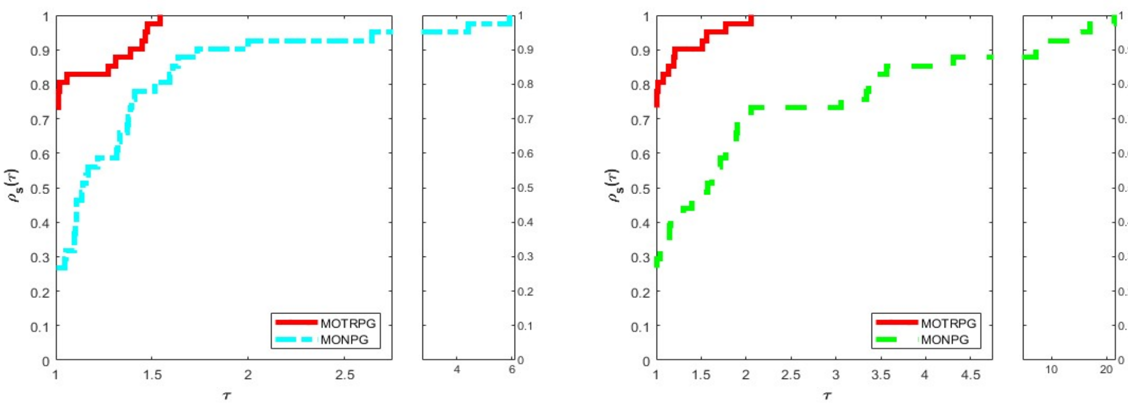
FIGURE 6. Performance profiles using Δ-spread metric




(A) Between MOTRPG and MONPG

(B) Between MOTRPG and MOPG

FIGURE 7. Performance profiles using hypervolume metric




(A) Between MOTRPG and MONPG

(B) Between MOTRPG and MOPG

FIGURE 8. Performance profiles using # fun

From the performance profile figures (Figures 4a-8b), MONPG typically outperforms MOPG and MONPG in terms of purity, Γ-spread metric, and total function evaluations. In terms of other measures, MOTRPG's performance is mediocre.

## 6. CONCLUSION

This work develops a trust region proximal gradient approach for multi-objective optimization problems that are nonlinear and convex. There are no predetermined parameters or objective function ordering information in this approach. Under a few minor assumptions, the global convergence is justified. Nevertheless, only convex multi-objective optimization problems can be solved by using our approach. We intend to apply our findings to nonconvex multi-objective optimization issues in the future research. To create approximate Pareto fronts, we employed the multi-start technique, which occasionally fails to produce a well-distributed approximate Pareto front. As a future extension of our approach, a well-distributed approximate Pareto front can be generated by developing some beginning point selection techniques.

**Acknowledgements**
The author is thankful to the reviewer for his/her detailed comments and suggestions that have significantly improved the content as well as the presentation of the results in this paper.

## REFERENCES

[1] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, Wiley, New Delhi, 2003.
[2] K. Miettinen, Nonlinear Multiobjective Optimization, Kluwer, Boston, 1999.
[3] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist nondominated sorting genetic algorithm for multi-objective optimization: NSGA II, IEEE Trans. Evol. Comput. 6 (2002), 182-197.
[4] K. Deb, H. Jain, An evolutionary many-objective optimization algorithm using reference-point-based non-dominated sorting approach, part I: solving problems with box constraints, IEEE Trans. Evol. Comput. 18 (2013), 577–601.
[5] J. Fliege, B. F. Svaiter, Steepest descent methods for multicriteria optimization, Math. Methods Oper. Res., 51 (2000) 479–494.
[6] J. Fliege, L. M. G. Drummond, B. F. Svaiter, Newton's method for multiobjective optimization, SIAM J. Optim, 20 (2009) 602–626.
[7] M. A. T. Ansary, G. Panda, A modified quasi-Newton method for vector optimization problem, Optimization, 64 (2015), 2289–2306.
[8] M. A. T. Ansary, G. Panda, A sequential quadratically constrained quadratic programming technique for a multi-objective optimization problem, Eng. Optim. 51 (2019), 22–41.
[9] 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.
[10] M. A. T. Ansary, G. Panda, A globally convergent SQCQP method for multiobjective optimization problems, SIAM J. Optim. 31 (2021), 91–113.
[11] S. Qu, M. Goh, B. Liang, Trust region methods for solving multiobjective optimisation, Optim. Methods Softw. 28 (2013) 796–811.
[12] N. K. Bisui, G. Panda, Adaptive trust region scheme for multi-objective optimization problem using Geršgorin circle theorem, J. Appl. Math. Comput. 68 (2022), 2151–2172.
[13] N. K. Bisui, G. Panda, A trust-region scheme for constrained multiobjective optimization problems with superlinear convergence property, Optim. Methods Softw. 24 (2024), 24–64.
[14] A. Mohammadi, A. L. Custódio, A trust-region approach for computing Pareto fronts in multiobjective optimization, Comput. Optim. Appl. 87(2024), 149-179.
[15] V. A. Ramirez, G. N. Sottosanto, Nonmonotone trust region algorithm for solving the unconstrained multi-objective optimization problems, Comput. Optim. Appl., 81(2022), 769-788.
[16] G. D. C. Bento, J. X. Cruz-Neto, G. Lopez, A. Soubeyran, J. C. O. Souza, The proximal point method for locally Lipschitz functions in multiobjective optimization with application to the compromise problem, SIAM J. Optim. 28 (2018), 1104–1120.
[17] H. Bonnel, A. N. Iusem, B. F. Svaiter, Proximal methods in vector optimization, SIAM J. Optim. 15 (2005), 953–970.
[18] M. A. T. Ansary, A Newton-type proximal gradient method for nonlinear multi-objective optimization problems, Optim. Methods Softw. 38 (2023), 570–590.
[19] J. W. Peng, J. C. Yao, The quasi-Newton method for the composite multiobjective optimization problems, arXiv preprint arXiv:2309.04966, 2023.
[20] H. Tanabe, E. H. Fukuda, N. Yamashita, Proximal gradient methods for multiobjective optimization and their applications, Comput. Optim. Appl. 72 (2019), 339–361.
[21] J. Y. Bello Cruz, A subgradient method for vector optimization problems, SIAM J. Optim. 23 (2013), 2169–2182.
[22] O. Montonen, N. Karmitsa, M. M. Mäkelä, Multiple subgradient descent bundle method for convex non-smooth multiobjective optimization, Optimization, 67 (2018), 139–158.
[23] J. X. D. C. Neto, G. J. P. Da Silva, O. P. Ferreira, J. O. Lopes, A subgradient method for multiobjective optimization, Comput. Optim. Appl. 54 (2013), 461–472.

[24] S. Kumar, M. A. T. Ansary, N. K. Mahato, D. Ghosh, Steepest descent method for uncertain multiobjective optimization problems under finite uncertainty set, Appl. Anal. 104 (2024), 1163–1184.

[25] S. Kumar, M. A. T. Ansary, N. K. Mahato, D. Ghosh, Y. Shehu, Newton's method for uncertain multiobjective optimization problems under finite uncertainty sets, J. Nonlinear Var. Anal. 7 (2023), 785-809.

[26] S. Kumar, N. K. Mahato, M. A. T. Ansary, D. Ghosh, S. Treanţă, A modified quasi-Newton method for uncertain multiobjective optimization problems under a finite uncertainty set, Eng. Optim. 57 (2024), 1392–1421.

[27] S. Kumar, N. K. Mahato, D. Ghosh, A trust region method for uncertain multiobjective optimization: comparative analysis with existing descent methods, Optim. Eng. (2026), 1-45.

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

[29] A. Beck, M. Teboulle, A fast dual proximal gradient algorithm for convex minimization and applications, Oper. Res. Lett. 42 (2014), 1–6.

[30] Y. Nesterov, Gradient methods for minimizing composite functions, Math. Program. 140 (2013), 125–161.

[31] J. D. Lee, Y. Sun, M.A. Saunders, Proximal Newton-type methods for minimizing composite functions, SIAM J. Optim. 24 (2014), 1420–1443.

[32] M. A. T. Ansary, J. Dutta, "A proximal gradient method for multiobjective optimization problems using Bregman functions, Optimization Online, 2022.

[33] A. Dhara, J. Dutta, Optimality Conditions in Convex Optimization: A Finite-dimensional View, CRC Press, London, 2011.

[34] M. J. D. Powell, A fast algorithm for nonlinearly constrained optimization calculations, In: G.A. Watson (ed.), Numerical analysis, pp. 144–157, Springer Berlin, Heidelberg, 1978.

[35] S. Huband, P. Hingston, L. Barone, L. While, A review of multiobjective test problems and a scalable test problem toolkit, IEEE Trans. Evol. Comput. 10 (2006), 477–506.

[36] S. Boyd, L. Vandenberghe, Introduction to Applied Linear Algebra: Vectors, Matrices, and Least squares, Cambridge University Press, Cambridge, 2018.

[37] Y. Jin, M. Olhofer, B. Sendhoff, Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how?, In. Proceedings of the $3^{rd}$ Annual Conference on Genetic and Evolutionary Computation, pp. 1042–1049, San Francisco, 2001.

[38] A. Lovison, Singular continuation: Generating piecewise linear approximations to Pareto sets via global analysis, SIAM J. Optim. 21 (2011), 463–490.

[39] A. Martín, O. Schütze, Pareto tracer: A predictor-corrector method for multi-objective optimization problems, Eng. Optim. 50 (2018), 516–536.

[40] O. Schütze, A. Dell'Aere, M. Dellnitz, On continuation methods for the numerical treatment of multiobjective optimization problems, In. Dagstuhl Seminar Proceedings, Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2005.