

Chapter 2

A Nonmonotone Polak-Ribière-Polyak Conjugate Gradient Method for MOPs

2.1 Introduction

Optimization problems are ubiquitous in various fields of science, engineering, and economics. Among the numerous methods developed to solve these problems, the conjugate gradient method stands out for its efficiency and simplicity. Introduced by Fletcher and Reeves [47] in the 1960s, conjugate gradient methods have become powerful tools for solving unconstrained optimization problems of the form

$$\text{minimize } f(x), \quad x \in \mathbb{R}^n, \quad (2.1)$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is assumed to be continuously differentiable. The fundamental idea behind conjugate gradient methods is to generate a sequence of iterates that converge to the optimal solution of the problem.

The conjugate gradient method starts with an initial point $x^0 \in \mathbb{R}^n$ and generates a

sequence of iterates according to the following update rule:

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, 2, \dots, \quad (2.2)$$

where α_k is the step size, typically determined by a line search technique, and d^k is the search direction. The search direction is updated using the formula:

$$d^0 = -\nabla f(x^0) \quad \text{and} \quad d^k = -\nabla f(x^k) + \beta_k d^{k-1}, \quad k = 1, 2, 3, \dots \quad (2.3)$$

The parameter β_k is crucial in determining the behavior and efficiency of the conjugate gradient method. Different choices of β_k result in various conjugate gradient methods, each with its own characteristics and performance profiles. Some of the most well-known conjugate gradient methods include

- Hestenes-Stiefel (HS) [73]
- Polak-Ribière-Polyak (PRP) [122]
- Liu-Storey (LS) [96]
- Hager-Zhang (HZ) [68]
- Dai-Yuan (DY) [29]
- Fletcher-Reeves (FR) [47]
- Conjugate Descent (CD) [4]

When $f(x)$ is a strictly convex quadratic function with Hessian Q , the parameter β_k is chosen such that the directions d^k are Q -conjugate, i.e., $(d^k)^\top Q d^{k+1} = 0$ for all $k = 0, 1, 2, \dots$. In this case, the step size α_k is given by the exact line search

$$\alpha_k = \operatorname{argmin}\{f(x^k + \alpha d^k) : \alpha > 0\}. \quad (2.4)$$

For quadratic functions, this algorithm, known as the *linear conjugate gradient algorithm*, converges to the minimizer of f over \mathbb{R}^n within n iterations. For nonquadratic functions, the choice of β_k becomes more critical and leads to different conjugate gradient methods with varying convergence properties and practical performance. The effectiveness of these methods often depends on the specific problem structure and characteristics.

2.2 Motivation

While conjugate gradient methods have proven highly effective for singleobjective optimization problems, many real-world scenarios involve multiple, often conflicting objectives that need to be optimized simultaneously. These problems, known as MOPs, present unique challenges and require specialized approaches.

In MOPs, the concept of a single optimal solution is replaced by the notion of *Pareto optimality or efficiency*. A point is considered Pareto optimal if there exists no other feasible point that improves all objective functions simultaneously. The set of all Pareto optimal solutions forms the Pareto front, which represents the trade-offs between different objectives.

The increasing importance of MOPs in various fields has led to intensive research over the past two decades to extend conventional optimization methods to handle multiple objectives. Some of the key developments in this area include

- Steepest descent methods for MOPs [49]
- Newton and quasi-Newton methods for vector optimization [50,123]
- Projected gradient methods for vector-valued functions [38]
- Proximal methods for multiobjective convex optimization [15]
- Extensions of various conjugate gradient methods to MOPs [63,64,98]

These approaches share a common advantage: they solve MOPs directly, without transforming them into a set of parameterized scalar problems. This makes them attractive alternatives to traditional scalarization techniques [145] and heuristic approaches [93].

Many of these methods, particularly those introduced in [8, 38, 49, 79, 123], are iterative and use Armijo-type conditions for step size selection. While this ensures a decrease in objective function values at each iteration, it can lead to potential issues:

- (i) As the number of objectives increases, the Armijo-type condition becomes stricter, potentially resulting in smaller step sizes.
- (ii) Smaller step sizes can slow down convergence, especially when the iterate lies in a narrow curved valley of the objective function landscape.

To address these challenges, researchers in singleobjective optimization have explored nonmonotone line search strategies. These strategies allow occasional increases in function values during the optimization process, which can help escape from local minima and improve convergence speed, particularly for complex problems with multiple local optima. Two notable nonmonotone line search approaches are

- (i) by Grippo et al. [66], which uses the maximum of recent objective function values as a benchmark, and
- (ii) by Zhang and Hager [156], which employs an average-type nonmonotone procedure, which has shown to be more effective in practice.

Despite their success in singleobjective optimization, nonmonotone line searches have not been extensively studied in the context of multiobjective descent methods, particularly in combination with conjugate gradient methods for MOPs.

Recent work by Lucambio et al. [98] has demonstrated the extension of several conjugate gradient methods (DY, FR, PRP, HS, and CD) to continuously differentiable vector-valued optimization problems. Gonçalves and Prudente [63] extended the HZ

conjugate gradient method to unconstrained MOPs, while Fukuda et al. [57] studied a conjugate gradient method for unconstrained quadratic MOPs. More recently, Gonçalves et al. [64] presented a study of LS conjugate gradient methods for vector optimization problems. These developments in conjugate gradient methods for MOPs, combined with the potential benefits of nonmonotone line search strategies, motivate the current study. There is a clear need to investigate the joint combination of conjugate gradient methods and nonmonotone schemes for MOPs, analyzing their convergence properties and practical performance. Among the various conjugate gradient methods, the PRP method stands out due to its unique characteristics. Its built-in restart feature automatically adjusts to avoid jamming [69], making it particularly suitable for the complex landscapes often encountered in MOPs. Furthermore, the PRP method offers flexibility with different line search techniques [69], allowing for adaptation to various problem structures. This balanced approach, which often demonstrates superior practical performance despite some theoretical convergence uncertainties [69], makes the PRP method an attractive choice for tackling the challenges posed by MOPs.

2.3 Contributions

This chapter makes several significant contributions to the field of multiobjective optimization, particularly in the development and analysis of conjugate gradient methods. The main contributions are as follows:

1. **Novel Algorithm:** We propose a Polak-Ribière-Polyak (PRP) conjugate gradient algorithm with an average-type nonmonotone (NM) Armijo-type line search for unconstrained MOPs. This algorithm combines the efficiency of the PRP method, which is generally considered the most effective conjugate gradient method for scalar problems [152], with the flexibility of nonmonotone line search techniques. To the best of our knowledge, this is the first study to explore the combination of conjugate gradient methods and nonmonotone schemes in the context

of MOPs.

2. **Convergence Analysis:** We provide a comprehensive convergence analysis of the proposed method. This includes:
 - Asymptotic convergence: We prove that any limit point generated by the sequence of iterates is a Pareto critical point.
 - Global convergence: We establish two key lemmas - one that provides an upper bound on the norm of the search directions for each iteration, and another that gives a lower bound of an algorithmic parameter.
3. **Complexity Analysis:** We derive a complexity bound for the proposed method. This bound indicates the maximum number of iterations required to achieve an ϵ -precise Pareto critical point, providing insights into the algorithm's efficiency and scalability.
4. **Numerical Experiments:** We conduct extensive numerical experiments on both convex and non-convex MOPs to demonstrate the practical efficiency of the proposed algorithm. These experiments provide empirical evidence of the algorithm's performance across a range of problem types and sizes.
5. **Comparative Analysis:** We perform a detailed comparative analysis of the proposed method against state-of-the-art algorithms in the field. Specifically:
 - We compare our method with the HZ conjugate gradient method [63] and the LS conjugate gradient method [64] for MOPs.
 - We use performance profiles [37] and relative efficiency measures [27, 152] to provide a comprehensive evaluation of the algorithm's performance.
6. **Pareto Front Approximation:** We evaluate the algorithm's capability to generate complete approximations of the Pareto front. This is done by comparing our method with existing popular solvers using two key indicators:

- Inverted Generational Distance (IGD): Measures how well the generated solutions represent the true Pareto front.
- Hypervolume (HV): Quantifies the volume of the objective space dominated by the generated solutions.

This analysis provides insights into the algorithm's effectiveness in exploring the entire Pareto front, a crucial aspect in multiobjective optimization.

These contributions collectively advance the state of the art in multiobjective optimization, particularly in the development of efficient and robust conjugate gradient methods for MOPs. By combining the strengths of the PRP conjugate gradient method with nonmonotone line search strategies, our work opens new avenues for solving complex multiobjective problems more effectively.

The theoretical analyses provided in this thesis, including convergence proofs and complexity bounds, establish a solid foundation for the proposed method. These results not only guarantee the method's correctness but also provide insights into its behavior and performance characteristics.

Furthermore, the extensive numerical experiments and comparative analyses offer practical validation of the theoretical results. They demonstrate the proposed method's effectiveness across a range of problem types and sizes, and its competitiveness with existing state-of-the-art methods.

Finally, the evaluation of the method's ability to approximate the entire Pareto front addresses a crucial aspect of multiobjective optimization. This analysis provides valuable insights into the method's capability to generate diverse solutions that effectively represent the trade-offs between different objectives.

In summary, this thesis presents a comprehensive study of a novel conjugate gradient method for MOPs, encompassing theoretical developments, algorithmic innovations, and practical evaluations. The results and insights presented here have the potential

to significantly impact the field of multiobjective optimization and pave the way for future research in this important area.

In this work, we consider an unconstrained MOP which is given by:

$$\min_{x \in \mathbb{R}^n} F(x) = (f_1(x), f_2(x), \dots, f_m(x))^\top, \quad (2.5)$$

where $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is assumed to be continuously differentiable and f_i is bounded from below for each $i \in \{1, 2, 3, \dots, m\}$.

2.4 Nonmonotone Line Search

This section briefly explain classical nonmonotone line search procedures and the proposed nonmonotone line search for the MOP (2.5). Several studies have shown that nonmonotone technique enhances the performance of descent methods in scalar cases (see [156] and the references therein).

In the classical monotone line search technique, the choice of α_k is such that $F(x^{k+1}) \prec F(x^k)$. It means that each iteration results in a lower objective function value. In nonmonotone line search, a certain amount of growth is allowed in the objective function values, i.e., we choose $\alpha_k > 0$ satisfying

$$F(x^k + \alpha_k d^k) \preceq C^k + \rho \alpha_k \mathcal{M}(x^k, d^k) e, \quad (2.6)$$

with $\rho \in (0, 1)$ and $C^k \succeq F(x^k)$. Two different rules for updating C^k leads to two nonmonotone line search methods: *max-type nonmonotone line search* and *average-type nonmonotone line search*. Grippo et al. [66] proposed a max-type nonmonotone line search technique which is based on the maximum of recent objective function values of the previous iterations. The idea of average-type nonmonotone procedure was first proposed by Zhang and Hager [156]. Rather than using the maximum of recent objective

function values, it takes the average of the recent objective function values.

An improved version of the classical Armijo condition with a simple backtracking procedure has been proposed in [138] to achieve the sufficient descent condition in scalar minimization. At each iteration, the starting trial step-size is determined by an estimate L_k of the L (Lipschitz constant). Several estimation scheme for finding the Lipschitz constant for the scalar case can be found in [139]. Our study utilizes the estimate of L for Jacobian of F [64] which is given as

$$L \cong \frac{|\mathcal{M}(x^k, d_{SD}(x^k)) - \mathcal{M}(x^{k-1}, d_{SD}(x^k))|}{\|t^{k-1}\|_2},$$

where $t^{k-1} = x^k - x^{k-1}$. For each $k = 1, 2, 3, \dots$ we take into account the following estimate L_k of L [64]:

$$L_k = \max \left\{ L_{k-1}, \min \left\{ \frac{|\mathcal{M}(x^k, d_{SD}(x^k)) - \mathcal{M}(x^{k-1}, d_{SD}(x^k))|}{\|t^{k-1}\|_2}, \bar{S} \right\} \right\}, \quad (2.7)$$

where L_0 and \bar{S} are given real constants such that $L_0 > 0$, $\bar{S} > 0$ and $L_0 < \bar{S}$. It becomes apparent from (2.7) that it is an extension to the vector case of a Lipschitz estimate as given in [138].

The nonmonotone algorithms have proven to be effective in solving highly nonlinear and possibly ill-conditioned problems. Therefore, in our work we have considered an average-type nonmonotone line search technique since it is known to be more effective in practice, at least in the scalar case. We have used an average-type nonmonotone line search analogous to the line search used in the third algorithm of [64]. Therefore, in the proposed method, the required step-size α_k is the largest value in $\{\tau_k, \tau_k v, \tau_k v^2, \dots\}$ such that

$$F(x^{k+1}) \preceq C^k + \alpha_k \rho \mathcal{M}(x^k, d^k) e \text{ and} \quad (2.8)$$

$$\mathcal{M}(x^{k+1}, d^{k+1}) \leq \delta \mathcal{M}(x^{k+1}, d_{SD}(x^{k+1})) \quad (2.9)$$

where $\tau_k = -\frac{(1-\delta)\mathcal{M}(x^k, d_{SD}(x^k))}{L_k \|d^k\|_2^2}$, $\delta \in (0, 1)$, $x^{k+1} = x^k + \alpha_k d^k$, $d^0 = d_{SD}(x^0)$, $d^k = d_{SD}(x^k) + \beta_k d^{k-1}$, $k = 1, 2, 3, \dots$, β_k is the scalar-valued conjugate parameter, and C^k is updated according to the following rule:

$$C^{k+1} = \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} F(x^{k+1}), \text{ and } m_{k+1} = \sigma m_k + 1. \quad (2.10)$$

Note that conjugate gradient methods are classified on the basis of different expressions of β_k . The interested reader can see [4] for a detailed explanation of the classifications.

2.5 A PRP Conjugate Gradient Method for MOPs

In this part, we derive a PRP conjugate gradient method to solve (2.5). We first discuss the extension of the classical PRP conjugate gradient method. From now on, we assume the following hypothesis.

- (A) The level set $\mathcal{L} = \{x \mid F(x) \preceq F(x^0)\}$ is bounded, where $x^0 \in \mathbb{R}^n$ (initial point), and JF is L -Lipschitz continuous on an open set containing \mathcal{L} (i.e., there exists an open set $\mathcal{P} \supset \mathcal{L}$ such that for all $y, z \in \mathcal{P}$, it follows that $\|JF(y) - JF(z)\|_2 \leq L\|y - z\|_2$).

The general conjugate gradient method [4] for finding a solution of (2.5) is depicted by the following recurrence relation

$$x^{k+1} = x^k + \alpha_k d^k, \quad k = 0, 1, 2, \dots,$$

where the step-size α_k is attained by a line search method, and d^k (search direction) is given by

$$d^0 = d_{SD}(x^0) \text{ and } d^k = d_{SD}(x^k) + \beta_k d^{k-1}, \quad k = 1, 2, 3, \dots, \quad (2.11)$$

where β_k is the scalar-valued conjugate parameter.

There has been some significant improvement in the field of the conjugate gradient method for multiobjective optimization. For continuously differentiable vector-valued optimization, Lucambio et al. [98] demonstrated the extension of the CD, FR, PRP, DY, and HS conjugate gradient methods. In the following years, Gonçalves and Prudente [63] extended the HZ conjugate gradient method to solve unconstrained MOPs. Moreover, Fukuda et al. [57] studied the conjugate gradient for unconstrained quadratic MOPs. In a recent publication, Gonçalves et al. [64] studied LS conjugate gradient methods for vector optimization problems. Three types of the LS conjugate gradient method have been adapted for multiobjective problems in [64]. It can be seen from all the above-mentioned works that the conjugate gradient method has been extensively extended for multiobjective optimization. Although both LS and PRP conjugate gradient method have been extended in [98] for vector-valued optimization, but still a study on extension of different variants of LS method has been recently done for vector-valued optimization. Hence, following up on the recent trend, we consider the PRP conjugate gradient method with a nonmonotone line search for MOPs.

As given in [98] the vector extensions of the PRP conjugate parameter is given by

$$\beta_k^{\text{PRP}} = \frac{-\mathcal{M}(x^k, d_{SD}(x^k)) + \mathcal{M}(x^{k-1}, d_{SD}(x^k))}{-\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))}.$$

$\mathcal{M}(x, d)$ represents a measure of the descent condition at the point x along the direction d , as define in (1.4). Specifically, $\mathcal{M}(x^k, d_{SD}(x^k))$ quantifies the magnitude of descent achieved using the steepest descent direction $d_{SD}(x^k)$, as defined in (1.7). The function \mathcal{M} is used to ensure sufficient decrease in the objective function F at each iteration and serves as a basis for defining the step-size α_k . The following Algorithm 1 formalizes an extention on PRP conjugate gradient method with an *average-type Armijo-like nonmonotone* line search for MOP (2.5).

Algorithm 1 A nonmonotone PRP conjugate gradient method to generate a discrete approximation of the complete set of Pareto critical points of (2.5)

Aim: To generate a discrete approximation to the complete set of Pareto critical points of the problem (2.5)

- 1: Provide $F = (f_1, f_2, \dots, f_m)^\top$ as in problem (2.5)
- 2: Choose arbitrarily $\sigma \in (0, 1)$, $\rho \in (0, 1)$, $\delta \in (0, 1)$, $v \in (0, 1)$, $\bar{S} \in (0, \infty)$ and $L_0 \in (0, \bar{S})$
- 3: Provide \mathcal{N} , the number of initial points to be randomly chosen
- 4: Provide the tolerance level $\epsilon > 0$ for the optimum solution of the problem (2.5)
- 5: Set Pareto set $\mathcal{S} \leftarrow \emptyset$
- 6: **for** $n = 1 : 1 : \mathcal{N}$ **do**
- 7: Choose a random point $x^0 \in \mathbb{R}^n$
- 8: Set $C^0 \leftarrow F(x^0)$, $m_0 \leftarrow 1$, $k \leftarrow 0$
- 9: Compute $d_{SD}(x^0)$ as in (1.7) and set $d^0 \leftarrow d_{SD}(x^0)$
- 10: Compute $\Theta(x^k)$ using (1.8).
- 11: Compute the search direction by (2.11)
- 12: **while** $|\Theta(x^k)| > \epsilon$ **do**
- 13: Set $\tau_k = -\frac{(1-\delta) \mathcal{M}(x^k, d_{SD}(x^k))}{L_k \|d^k\|_2^2}$, where L_k is taken from (2.7)
- 14: Set $\alpha \leftarrow \tau_k$
- 15: Set $x^+ \leftarrow x^k + \alpha d^k$
- 16: Compute $d_{SD}(x^+)$ and $d(x^+)$ using (1.7) and (2.11)
- 17: **while** $F(x^+) \succ C^k + \alpha_k \rho \mathcal{M}(x^k, d^k)e$ or $\mathcal{M}(x^+, d(x^+)) > \delta \mathcal{M}(x^+, d_{SD}(x^+))$ **do**
- 18: $\alpha \leftarrow \alpha v$
- 19: **end while**
- 20: Set $x^{k+1} \leftarrow x^k + \alpha d^k$, $d_{SD}(x^{k+1}) \leftarrow d_{SD}(x^k + \alpha d^k)$ and $d(x^{k+1}) \leftarrow d(x^k + \alpha d^k)$
- 21: Set $k \leftarrow k + 1$
- 22: Update m_{k+1} and C^{k+1} by:

$$m_{k+1} \leftarrow \sigma m_k + 1 \quad \text{and} \quad C^{k+1} \leftarrow \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} F(x^{k+1})$$
- 23: Update $k \leftarrow k + 1$
- 24: Compute $\Theta_{SD}(x^k)$ and L_k using (1.8) and (2.7), respectively
- 25: **end while**
- 26: **return** $\bar{x} = x^k$ as a Pareto critical point of (2.5)
- 27: Update the set $\mathcal{S} \leftarrow \mathcal{S} \cup \{F(\bar{x})\}$
- 28: **end for**
- 29: **return** \mathcal{S} as a discrete approximation of the whole Pareto set of the problem (2.5)

In average-type nonmonotone line searches, C^{k+1} depicted as a convex combination of $F(x^{k+1})$ and C^k . Since $C^0 = F(x^0)$, C^k acts as a convex combination of all the objective values $F(x^0), F(x^1), \dots, F(x^k)$. The parameter σ is responsible for controlling the degree of nonmonotonicity in C^k . When $\sigma = 0$, we get $C^k = F(x^k)$, and as a result, the

nonmonotone line search (2.6) reduces to the monotone line search. When $\sigma = 1$, (2.10) implies that C^k is the average of objective function values $F(x^0), F(x^1), \dots, F(x^k)$.

It is important to note that Algorithm 1 either terminates at a Pareto critical point or produces an infinite sequence x^k of non Pareto critical points. We proceed further assuming that the algorithm iterates infinitely. For the global convergence of the proposed Algorithm 1, we first recall a useful property of the vector C^k , defined in (2.10). The following result shows that for each i , $f_i(x^k)$ and $S_i^k = \frac{1}{k+1} \sum_{j=0}^k f_i(x^j)$ represent lower and upper bounds of C_i^k , respectively.

Lemma 2.1 *Let $x^k \in \mathbb{R}^n$ be a non Pareto critical point of (2.5) generated by Algorithm 1. Then, we have $F(x^k) \preceq C^k \preceq S^k$, where $S^k = \frac{1}{k+1} \sum_{j=0}^k F(x^j)$.*

Proof: Let us define a function $\mathcal{B}^k : \mathbb{R} \rightarrow \mathbb{R}^m$ by

$$\mathcal{B}^k(t) = \frac{1}{t+1} (tC^{k-1} + F(x^k)).$$

The derivative of $\mathcal{B}^k(t)$ is given by

$$\frac{d\mathcal{B}^k(t)}{dt} = \frac{1}{(t+1)^2} (C^{k-1} - F(x^k)).$$

As x^k is non Pareto critical point, we have $\mathcal{M}(x^{k-1}, d^{k-1}) < 0$. It follows from the condition in (2.8) that $F(x^k) \preceq C^{k-1}$. This implies $\frac{d\mathcal{B}^k(t)}{dt} \succeq 0 \forall t \neq -1$. Thus, \mathcal{B}^k is nondecreasing $\forall t \geq 0$. The facts that $\sigma \in [0, 1]$ and $m_0 = 1$ imply that $m_k \geq 1 \forall k = 0, 1, 2, \dots$. Hence, $\sigma m_{k-1} \geq 0$, and we have

$$F(x^k) = \mathcal{B}^k(0) \preceq \mathcal{B}^k(\sigma m_{k-1}) = C^k.$$

Next, we show $C^k \preceq S^k$ by mathematical induction. For $k = 0$, the inequality trivially holds due to $C^0 = S^0 = F(x^0)$. Let $C^j \preceq S^j \forall 0 \leq j \leq k-1$. As $\sigma \in [0, 1]$ and $m_0 = 1$,

from (2.10) we have the following inequality

$$m_k = 1 + \sum_{\mathcal{J}=1}^k \sigma^{\mathcal{J}} \leq 1 + k \quad \text{for all } k = 1, 2, 3, \dots$$

Thus, $0 \leq m_k - 1 \leq k$ holds. Since \mathcal{B}^k is nondecreasing $\forall t \geq 0$ and $m_k = \sigma m_{k-1} + 1$, we get

$$C^k = \mathcal{B}^k(\sigma m_{k-1}) = \mathcal{B}^k(m_k - 1) \preceq \mathcal{B}^k(k).$$

Through the mathematical induction, we additionally have

$$\mathcal{B}^k(k) = \frac{1}{(k+1)}(kC^{k-1} + F(x^k)) \preceq \frac{1}{(k+1)}(kS^{k-1} + F(x^k)) = S^k,$$

and the result follows. \square

The following proposition shows that Algorithm 1 is well-defined. The purpose is to show that there always exists a stepsize α (for line 20, of Algorithm 1) which follow the conditions (2.8) and (2.9) so that the iterates x^k 's can be generated.

Lemma 2.2 *Assume that d^k is a descent direction for F at x^k . Let $\rho \in (0, 1)$, $\delta \in (0, 1)$, and $e = (1, 1, \dots, 1)^\top \in \mathbb{R}^m$. Then, $\exists \bar{\alpha} \in (0, 1]$ such that $\forall \alpha \in (0, \bar{\alpha}]$,*

$$F(x^k + \alpha d^k) \preceq C^k + \alpha \rho \mathcal{M}(x^k, d^k) e \quad \text{and}$$

$$\mathcal{M}(x^k + \alpha d^k, d(x^k + \alpha d^k)) \leq \delta \mathcal{M}(x^k + \alpha d^k, d_{SD}(x^k + \alpha d^k)),$$

where

$$d(x^k + \alpha d^k) = d_{SD}(x^k + \alpha d^k) + \left(\frac{-\mathcal{M}(x^k + \alpha d^k, d_{SD}(x^k + \alpha d^k)) + \mathcal{M}(x^k, d_{SD}(x^k + \alpha d^k))}{-\mathcal{M}(x^k, d_{SD}(x^k))} \right) d^k.$$

Proof: Since d^k is a descent direction of F , we obtain $\mathcal{M}(x^k, d^k) < 0$. Moreover,

$\mathcal{M}(x^k, d^k) < \rho\mathcal{M}(x^k, d^k)$ because $\rho \in (0, 1)$. Now, since F is differentiable, we obtain

$$f_i(x^k + \alpha d^k) = f_i(x^k) + \langle \nabla f_i(x^k), \alpha d^k \rangle + o(\alpha) \quad \forall i = 1, 2, \dots, m. \quad (2.12)$$

As $\langle \nabla f_i(x^k), d^k \rangle \leq \mathcal{M}(x^k, d^k) < \rho\mathcal{M}(x^k, d^k) < 0$ for all $i = 1, 2, 3, \dots$, thus there exists $\alpha' \in (0, 1)$ such that for all $i = 1, 2, \dots, m$, (2.12) can be rewritten as

$$f_i(x^k + \alpha d^k) \leq f_i(x^k) + \rho\alpha\mathcal{M}(x^k, d^k) \quad \text{for all } \alpha \in (0, \alpha'].$$

Since $f_i(x^k) \leq C_i^k$ for all $i = 1, 2, \dots, m$ from Lemma 2.1, the above inequality results in

$$f_i(x^k + \alpha d^k) \leq C_i^k + \rho\alpha\mathcal{M}(x^k, d^k) \quad \text{for all } \alpha \in (0, \alpha'].$$

Now, we define two functions $\eta, \xi : \mathbb{R}_+ \rightarrow \mathbb{R}$ by

$$\eta(\alpha) = \mathcal{M}(x^k + \alpha d^k, d(x^k + \alpha d^k)) \quad \text{and} \quad \xi(\alpha) = \delta\mathcal{M}(x^k + \alpha d^k, d_{SD}(x^k + \alpha d^k)).$$

Since d^k is a descent direction of F , thus from the equation (2.9), we can write $\eta(0) < \xi(0)$. Thus, due to the continuity of \mathcal{M} , $\exists \alpha'' \in (0, 1)$ such that $\eta(\alpha) \leq \xi(\alpha) \quad \forall \alpha \in (0, \alpha'')$, i.e.,

$$\mathcal{M}(x^k + \alpha d^k, d(x^k + \alpha d^k)) \leq \delta\mathcal{M}(x^k + \alpha d^k, d_{SD}(x^k + \alpha d^k)) \quad \forall \alpha \in (0, \alpha'').$$

Therefore, by taking $\bar{\alpha} = \min\{\alpha', \alpha''\}$, we conclude that for all $\alpha \in (0, \bar{\alpha}]$,

$$F(x^k + \alpha d^k) \preceq C^k + \alpha\rho\mathcal{M}(x^k, d^k)e \quad \text{and}$$

$$\mathcal{M}(x^k + \alpha d^k, d(x^k + \alpha d^k)) \leq \delta\mathcal{M}(x^k + \alpha d^k, d_{SD}(x^k + \alpha d^k)).$$

□

Before proving the global convergence for Algorithm 1, we establish two significant results which are shown in Lemmas 2.3 and 2.4.

Lemma 2.3 *Let x^k, d^k, L_k (estimate of L), L_0 and δ be given as in Algorithm 1. Then,*

$$\|d^k\|_2 \leq \left(1 + \frac{L(1-\delta)}{L_0}\right) \|d_{SD}(x^k)\|_2 \quad \forall k = 0, 1, 2, \dots \quad (2.13)$$

Proof: For $k = 0$, we have $d^0 = d_{SD}(x^0)$, and hence (2.13) trivially holds. For $k \geq 1$, since $L_{k-1} \geq L_0$, it follows from Steps 13 and 18 of Algorithm 1 that $\alpha_{k-1} \leq \tau_{k-1}$. Thus, we can write

$$\alpha_{k-1} \leq -\frac{(1-\delta) \mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))}{L_{k-1} \|d^{k-1}\|_2^2} \leq -\frac{(1-\delta) \mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))}{L_0 \|d^{k-1}\|_2^2} \quad (2.14)$$

due to the fact $\mathcal{M}(x^{k-1}, d^{k-1}) < 0$.

Now, by (2.11), Assumption (A), Lemma 1.1(iii) and because of $x^k - x^{k-1} = \alpha_{k-1} d^{k-1}$, we obtain from (2.11) that

$$\begin{aligned} \|d^k\|_2 &\leq \|d_{SD}(x^k)\|_2 + \frac{|-\mathcal{M}(x^k, d_{SD}(x^k)) + \mathcal{M}(x^{k-1}, d_{SD}(x^k))|}{-\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))} \|d^{k-1}\|_2 \\ &\leq \|d_{SD}(x^k)\|_2 + \frac{L \|d_{SD}(x^k)\|_2 \|x^k - x^{k-1}\|_2}{-\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))} \|d^{k-1}\|_2 \\ &= \|d_{SD}(x^k)\|_2 \left(1 + \frac{L \|\alpha_{k-1}\|_2 \|d^{k-1}\|_2^2}{-\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))}\right). \end{aligned} \quad (2.15)$$

Using (2.14) and (2.15) we get

$$\|d^k\|_2 \leq \left(1 + \frac{L(1-\delta)}{L_0}\right) \|d_{SD}(x^k)\|_2 \quad \forall k = 0, 1, 2, \dots$$

This proves the required inequality (2.13). \square

Lemma 2.4 *Let L_k (estimate of L), L_0 , δ and \bar{S} as in Algorithm 1. We define*

$$\tau_{\min} = \frac{(1-\delta)}{2\bar{S}} \left(1 + \frac{L(1-\delta)}{L_0}\right)^{-2} > 0.$$

Then, $\tau_k > \tau_{\min}$ for all $k = 0, 1, 2, \dots$.

Proof: By (2.7), clearly we have $L_k \leq \bar{S}$ for each $k = 0, 1, 2, \dots$. Thus, from the definition of τ_k , we have

$$\tau_k = -\frac{(1-\delta)}{L_k} \frac{\mathcal{M}(x^k, d_{SD}(x^k))}{\|d^k\|_2^2} \geq -\frac{(1-\delta)}{\bar{S}} \frac{\mathcal{M}(x^k, d_{SD}(x^k))}{\|d^k\|_2^2} \quad (2.16)$$

because $\mathcal{M}(x^k, d_{SD}(x^k)) < 0$, by Lemma 1.2(ii).

By using (2.16) and Lemma 1.2(ii), we can write

$$\tau_k = -\frac{(1-\delta)}{L_k} \frac{\mathcal{M}(x^k, d_{SD}(x^k))}{\|d^k\|_2^2} > \frac{(1-\delta)}{2\bar{S}} \frac{\|d_{SD}(x^k)\|^2}{\|d^k\|_2^2}.$$

Then, incorporating (2.13), we get

$$\tau_k > \frac{(1-\delta)}{2\bar{S}} \left(1 + \frac{L(1-\delta)}{L_0}\right)^{-2},$$

which proves the required result. □

2.6 Global Convergence

This section demonstrates global convergence of Algorithm 1. First, we prove some supporting results for global convergence. In the next lemma, we depict a significant characteristic of the sequence $\{C^k\}$, which is obtained by the average-type nonmonotone line search procedure.

Lemma 2.5 *Let $\sigma \in (0, 1)$ and $\{x^k\}$ be a sequence obtained by Algorithm 1. Then, for every $i = 1, 2, \dots, m$, $\{C_i^k\}$ is a nonincreasing sequence and convergent.*

Proof: By (2.8) and (2.10), for each $i = 1, 2, \dots, m$ and for all $k = 0, 1, 2, \dots$, we obtain

$$\begin{aligned} C_i^{k+1} &= \frac{\sigma m_k}{m_{k+1}} C_i^k + \frac{1}{m_{k+1}} f_i(x^{k+1}) \leq \frac{\sigma m_k}{m_{k+1}} C_i^k + \frac{1}{m_{k+1}} [C_i^k + \rho \alpha_k \mathcal{M}(x^k, d^k)] \\ &= C_i^k + \frac{\rho \alpha_k}{m_{k+1}} \mathcal{M}(x^k, d^k) \leq C_i^k, \end{aligned} \quad (2.17)$$

where the last inequality is because of the fact that $\mathcal{M}(x^k, d^k) < 0$. Therefore, the sequence $\{C_i^k\}$ is non-increasing. Since f_i is bounded from below, and Lemma 2.1 gives $f_i(x^k) \leq C_i^k \forall i$ and k , we conclude that $\{C_i^k\}$ converges for every $i = 1, 2, \dots, m$. \square

Let $\{x^k\}$ be a sequence obtained by Algorithm 1. As $F(x^k) \preceq F(x^0) \forall k = 0, 1, 2, \dots$, we get $\{x^k\} \subset \mathcal{L}$. Thus, by Assumption (A), we conclude that $\{x^k\}$ is bounded which guarantees existence of a subsequential limit of the sequence $\{x^k\}$. Here \mathcal{L} is compact, and since F is continuous, so due to Lemma 1.2(iii) and from the continuity of JF there exist \bar{p} and \bar{q} such that

$$\|d_{SD}(x^k)\|_2 \leq \bar{p}, \quad \|JF(x^k)\|_2 \leq \bar{q} \quad \forall k = 0, 1, 2, \dots \quad (2.18)$$

Now we prove the global convergence of the sequence $\{x^k\}$ obtained by Algorithm 1. In particular, we prove that every subsequential limit of $\{x^k\}$ is Pareto critical.

Theorem 2.1 *Let $\{(x^k, d^k)\}$ be the sequence obtained by Algorithm 1. If \hat{x} is a subsequential limit of $\{x^k\}$, then \hat{x} is Pareto critical of F .*

Proof: Assume that $K = \{k_0, k_1, k_2, \dots\} \subseteq \mathbb{N}$ be a sequence of indices such that $\lim_{j \rightarrow \infty} x^{k_j} = \hat{x}$. Since $k_{j+1} \geq k_j + 1$, due to Lemmas 2.1 and 2.5 and the inequalities in (2.17), we obtain, for all $j \in \mathbb{N}$,

$$f_i(x^{k_{j+1}}) \leq C_i^{k_{j+1}} \leq C_i^{k_j+1} \leq C_i^{k_j} + \frac{\rho \alpha_{k_j}}{m_{k_{j+1}}} \mathcal{M}(x^{k_j}, d^{k_j}) \leq C_i^{k_j} \quad \forall i = 1, \dots, m. \quad (2.19)$$

Since the objective function is continuous (see (2.5)), we have

$$\lim_{j \rightarrow \infty} F(x^{k_j}) = F(\hat{x}),$$

which indicates that $\{F(x^{k_j})\}$ is bounded. As a consequence, by (2.19), we have, for every $i = 1, 2, \dots, m$, that $\{C_i^{k_j}\}$ is a monotone bounded sequence, and therefore its limit exists. From (2.19), we have

$$C_i^{k_{j+1}} \leq C_i^{k_j} + \frac{\rho \alpha_{k_j}}{m_{k_{j+1}}} \mathcal{M}(x^{k_j}, d^{k_j}) \quad \forall i = 1, 2, \dots, m. \quad (2.20)$$

By (2.20), we can write

$$0 \leq \lim_{j \rightarrow \infty} \frac{-\rho \alpha_{k_j}}{m_{k_{j+1}}} \mathcal{M}(x^{k_j}, d^{k_j}) \leq \lim_{j \rightarrow \infty} C_i^{k_j} - C_i^{k_{j+1}} = 0 \quad (2.21)$$

because $\mathcal{M}(x^{k_j}, d^{k_j}) < 0$. Hence,

$$\lim_{j \rightarrow \infty} \left\{ \left(\frac{\alpha_{k_j}}{m_{k_{j+1}}} \right) \mathcal{M}(x^{k_j}, d^{k_j}) \right\} = 0.$$

We claim that

$$\lim_{j \rightarrow \infty} \{ \alpha_{k_j} \mathcal{M}(x^{k_j}, d^{k_j}) \} = 0. \quad (2.22)$$

As $\sigma \in (0, 1)$, by (2.10), we have

$$m_{k_{j+1}} = 1 + \sum_{\ell=0}^{k_j} \sigma^{\ell+1} \leq \sum_{\ell=0}^{\infty} \sigma^{\ell} = \frac{1}{1-\sigma}. \quad (2.23)$$

Therefore, $\left\{ \frac{1}{m_{k_{j+1}}} \right\}$ is bounded from below.

By (2.21) and (2.23), we get

$$0 \leq \lim_{j \rightarrow \infty} \left\{ -(1-\sigma) \rho \alpha_{k_j} \mathcal{M}(x^{k_j}, d^{k_j}) \right\} \leq \lim_{j \rightarrow \infty} \left\{ \frac{-\rho \alpha_{k_j}}{m_{k_{j+1}}} \mathcal{M}(x^{k_j}, d^{k_j}) \right\} \leq 0.$$

Thus, (2.22) follows.

Now, by (2.22), $\exists K' \subset K$ for which at least one the following two cases hold:

1. $\lim_{k \in K'} \mathcal{M}(x^k, d^k) = 0$ or
2. $\lim_{k \in K'} \alpha_k = 0$.

For both of the above cases, we show that the subsequential limit point \hat{x} of the sequence $\{x^k\}$ is a Pareto critical point of F .

Case 1: By (2.9) and Lemma 1.2(ii), we have

$$\mathcal{M}(x^k, d^k) \leq \delta \mathcal{M}(x^k, d_{SD}(x^k)) \leq -\frac{\delta}{2} \|d_{SD}(x^k)\|_2^2 \leq 0. \quad (2.24)$$

Thus, applying limits in (2.24) for $k \in K'$, we get

$$\lim_{k \in K'} d_{SD}(x^k) = d_{SD}(\hat{x}) = 0.$$

Hence, by Lemma 1.2(i), \hat{x} is a Pareto critical point of F .

Case 2: Without loss of generality, assume that $\alpha_k < \tau_{\min} \quad \forall k \in K'$, where $\tau_{\min} > 0$ is as given in Lemma 2.4. Therefore, by Lemma 2.4 and Steps 13 and 18 of Algorithm 1, there exists $\bar{\alpha}_k = \frac{\alpha_k}{v}$ for each $k \in K'$ such that

$$F(\bar{x}^k) \succ C^k + \rho \bar{\alpha}_k \mathcal{M}(x^k, d^k) e, \quad (2.25)$$

or

$$\mathcal{M}(\bar{x}^k, d(\bar{x}^k)) > \delta \mathcal{M}(\bar{x}^k, d_{sd}(\bar{x}^k)), \quad (2.26)$$

where $\bar{x}^k = x^k + \bar{\alpha} d^k$ and v is taken from Algorithm 1.

Case 2.1: Assume that (2.25) holds for infinitely many k 's, say in $K'' \subset K'$. Thus,

there exists at least one $i \in \{1, 2, \dots, m\}$ such that $\forall k \in K''$, we can write

$$f_i(x^k + \bar{\alpha}d^k) > C_i^k + \rho\bar{\alpha}_k\mathcal{M}(x^k, d^k) \geq f_i(x^k) + \rho\bar{\alpha}_k\mathcal{M}(x^k, d^k), \quad (2.27)$$

here the second inequality is due to Lemma 2.1.

By the mean value theorem, $\forall k \in K'' \exists \gamma_k \in [0, 1]$ such that

$$\langle \nabla f_i(x^k + \gamma_k\bar{\alpha}_k d^k), \bar{\alpha}_k d^k \rangle = f_i(x^k + \bar{\alpha}_k d^k) - f_i(x^k) \quad \forall i \in \{1, 2, \dots, m\}. \quad (2.28)$$

Thus, by (2.27), (2.28), and from the definition of \mathcal{M} , we get

$$\bar{\alpha}_k\mathcal{M}(x^k + \gamma_k\bar{\alpha}_k d^k, d^k) \geq \langle \nabla f_i(x^k + \gamma_k\bar{\alpha}_k d^k), \bar{\alpha}_k d^k \rangle > \rho\bar{\alpha}_k\mathcal{M}(x^k, d^k) \quad \forall k \in K''. \quad (2.29)$$

Due to Lemma 2.3 and (2.18), $\{d^k\}$ is bounded. As a result, $\exists K''' \subset K''$ and $\hat{d} \in \mathbb{R}^n$ for which we have $\lim_{k \in K'''} d^k = \hat{d}$. Additionally, by the definition of $\bar{\alpha}_k$, we get $\lim_{k \in K'''} \bar{\alpha}_k = 0$. Thus, by dividing both sides of (2.29), by $\bar{\alpha}_k > 0$ and applying limits for $k \in K'''$, we have

$$\mathcal{M}(\hat{x}, \hat{d}) \geq \rho\mathcal{M}(\hat{x}, \hat{d}).$$

As $\rho \in (0, 1)$, and $\mathcal{M} < 0$ for all $k = 0, 1, 2, \dots$, we get $\mathcal{M}(\hat{x}, \hat{d}) = 0$. By (2.9) and Lemma 1.2(ii), it follows that

$$\mathcal{M}(x^k, d^k) \leq \delta\mathcal{M}(x^k, d_{SD}(x^k)) \leq -\frac{\delta}{2}\|d_{SD}(x^k)\|_2^2 \leq 0.$$

Thus, applying limits for $k \in K'''$ in the above inequality, we get

$$\lim_{k \in k'} d_{SD}(x^k) = d_{SD}(\hat{x}) = 0.$$

Hence, by Lemma 1.2(i), \hat{x} is a Pareto critical point of F .

Case 2.2: Assume that without loss of generality (2.26) follows and (2.25) does not hold $\forall k \in K'$. Thus, $\{\bar{x}^k\}_{k \in K'} \subset \mathcal{L}$, and incorporating (2.18), $\exists \bar{q} > 0$ such that

$$\|JF(\bar{x}^k)\|_2 \leq \bar{q} \quad \forall k \in K'. \quad (2.30)$$

Define $\xi = \text{sgn}(-\mathcal{M}(x^k, d_{SD}(x^k)) + \mathcal{M}(x^{k-1}, d_{SD}(x^k)))$, where $\text{sgn}(z)$ denotes the sign function, defined as:

$$\text{sgn}(z) = \begin{cases} 1 & \text{if } z > 0, \\ 0 & \text{if } z = 0, \\ -1 & \text{if } z < 0. \end{cases}$$

From the definition of \bar{x}^k and $d(\bar{x}^k)$, and Lemma 1.1(i), we have

$$\mathcal{M}(\bar{x}^k, d(\bar{x}^k)) \leq \mathcal{M}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \frac{|-\mathcal{M}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \mathcal{M}(x^k, d_{SD}(\bar{x}^k))|}{-\mathcal{M}(x^k, d^k)} \mathcal{M}(\bar{x}^k, \xi d^k). \quad (2.31)$$

By using (2.31), Lemma 1.1(iii), Assumption (A) and due to the fact that $\bar{x}^k - x^k = \bar{\alpha}_k d^k$, we get

$$\mathcal{M}(\bar{x}^k, d(\bar{x}^k)) \leq \mathcal{M}(\bar{x}^k, d_{SD}(\bar{x}^k)) + \frac{L\bar{\alpha}_k \|d_{SD}(\bar{x}^k)\|_2 \|d^k\|_2}{-\mathcal{M}(x^k, d_{SD}^k)} |\mathcal{M}(\bar{x}^k, \xi d^k)|. \quad (2.32)$$

Thus, incorporating (2.26) and (2.32), we can write

$$-(1 - \delta)\mathcal{M}(\bar{x}^k, d_{SD}(\bar{x}^k)) < \frac{L\bar{\alpha}_k \|d_{SD}(\bar{x}^k)\|_2 \|d^k\|_2}{-\mathcal{M}(x^k, d_{SD}^k)} |\mathcal{M}(\bar{x}^k, \xi d^k)|, \quad \forall k \in K'. \quad (2.33)$$

By the definition of \mathcal{M} as given in (1.4), there exists $\bar{w}^k \in \Omega$ such that

$$|\mathcal{M}(\bar{x}^k, \xi d^k)| = |\langle JF(\bar{x}^k)(\xi d^k), \bar{w}^k \rangle| \leq \|JF(\bar{x}^k)\|_2 \|d^k\|_2 \leq \bar{q} \|d^k\|_2, \quad (2.34)$$

where the inequality is due to $\|\bar{w}\|_2 = 1$ and (2.30). Here Ω is the canonical basis of

\mathbb{R}^m (see (1.5)). By (2.33), (2.34) and Lemma 1.2(ii), we obtain

$$(1 - \delta) \frac{\|d_{SD}(\bar{x}^k)\|_2^2}{2} < \frac{\bar{q}L\bar{\alpha}_k \|d_{SD}(x^k)\|_2 \|d^k\|_2^2}{-\mathcal{M}(x^k, d_{SD}^k)} \quad \forall k \in K',$$

or, equivalently,

$$\bar{\alpha}_k > -\frac{(1 - \delta) \mathcal{M}(x^k, d_{SD}^k)}{2\bar{q}L} \frac{\|d_{SD}(x^k)\|_2}{\|d^k\|_2^2} \quad \forall k \in K'.$$

By the definition of $\bar{\alpha}_k$ and Lemma 1.2(ii), we have

$$\alpha_k > \frac{v(1 - \delta)}{4\bar{q}L} \frac{\|d_{SD}(x^k)\|_2^2}{\|d^k\|_2^2} \|d_{SD}(\bar{x}^k)\|_2.$$

Hence, from Lemma 2.3, for all $k \in K'$, we have

$$\alpha_k > \frac{v(1 - \delta)}{4\bar{q}L} \left(1 + \frac{L(1 - \delta)}{L_0}\right)^{-2} \|d_{SD}(\bar{x}^k)\|_2. \quad (2.35)$$

From Lemma 2.3 and (2.18), $\{d^k\}$ is bounded. Since $\lim_{k \in K'} \alpha_k = 0$, it holds due the definition of \bar{x}^k that $\lim_{k \in K'} \bar{x}^k = \hat{x}$. Therefore, by applying limits in (2.35) for $k \in K'$, we obtain $d_{SD}(\hat{x}) = 0$. Thus, \hat{x} is a Pareto critical point of F , which completes the proof. \square

The next lemma depicts the iteration-complexity bounds of the proposed method.

Lemma 2.6 *Let $\rho \in (0, 1)$ be as in Algorithm 1. Then, for any $\mathcal{N} \in \mathbb{N}$, we have*

$$\min\{|\mathcal{M}(x^k, d^k)| : k = 0, 1, \dots, \mathcal{N} - 1\} \leq \frac{1}{\rho\mathcal{N}} (f_i(x^0) - f_i^{\inf}),$$

where $f_i^{\inf} = \inf\{f_i(x^k) : k = 0, 1, 2, \dots\}$

Proof: By (2.10) and the inequality (2.8), we have

$$\begin{aligned} C^{k+1} &= \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} F(x^{k+1}) \preceq \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} (C^k + \rho \alpha_k \mathcal{M}(x^k, d^k) e) \\ &= C^k + \frac{\rho \alpha_k}{m_{k+1}} \mathcal{M}(x^k, d^k) e. \end{aligned}$$

By the above inequality, the following inequality follows

$$C^{k+1} - C^k \preceq \frac{\rho \alpha_k}{m_{k+1}} \mathcal{M}(x^k, d^k) e. \quad (2.36)$$

By (2.36), and the fact that $\alpha_k \leq 1$ and $\mathcal{M}(x^k, d^k) < 0$, we have

$$C^{k+1} - C^k \prec \frac{\rho \alpha_k}{m_{k+1}} (\mathcal{M}(x^k, d^k)) e \preceq \frac{\rho}{m_{k+1}} (\mathcal{M}(x^k, d^k)) e \prec 0.$$

Thus, by the above inequality, we can write

$$C^{k+1} - C^k \preceq \frac{\rho}{m_{k+1}} (\mathcal{M}(x^k, d^k)) e \prec 0. \quad (2.37)$$

We take summation on both side of (2.37), for $k = 0, 1, 2, \dots, \mathcal{N} - 1$, and we get

$$C_i^{\mathcal{N}} - C_i^0 \leq \sum_{k=0}^{\mathcal{N}-1} \frac{\rho}{m_{k+1}} (\mathcal{M}(x^k, d^k)) \quad \forall i = 1, 2, \dots, m.$$

By (2.23), $\{\frac{1}{m_{k+1}}\}$ is bounded above by 1, thus we get

$$- \sum_{k=0}^{\mathcal{N}-1} (\mathcal{M}(x^k, d^k)) \leq \frac{1}{\rho} (C_i^0 - C_i^{\mathcal{N}}).$$

Since $C_i^0 = f_i(x^0)$, by Lemma 2.1, we can write

$$\sum_{k=0}^{\mathcal{N}-1} |\mathcal{M}(x^k, d^k)| \leq \frac{1}{\rho} (f_i(x^0) - f_i(x^{\mathcal{N}})). \quad (2.38)$$

As per the assumption that $f_i(x)$ is bounded below $\forall i = 1, \dots, m$ (see (2.5)), we get

$$\min\{|\mathcal{M}(x^k, d^k)| : k = 0, 1, \dots, \mathcal{N} - 1\} \leq \frac{1}{\rho \mathcal{N}}(f_i(x^0) - f_i^{\text{inf}}),$$

which completes the proof. \square

Theorem 2.2 *Let $\rho \in (0, 1)$ be as in Algorithm 1 and assume $\epsilon > 0$. If for every $\mathcal{N} \in \mathbb{N}$, we define the set $B(\epsilon) = \{k : |\mathcal{M}(x^k, d^k)| > \epsilon, k = 0, 1, 2, \dots, \mathcal{N} - 1\}$, then*

$$|B(\epsilon)| \leq \frac{(f_i(x^0) - f_i^{\text{inf}}) 1}{\rho \epsilon},$$

where $|B(\epsilon)|$ denotes the cardinality of the set $B(\epsilon)$.

Proof: By (2.38), we can write

$$\epsilon |B(\epsilon)| < \sum_{k=0}^{\mathcal{N}-1} |\mathcal{M}(x^k, d^k)| \leq \frac{1}{\rho} (f_i(x^0) - f_i(x^{\mathcal{N}})).$$

Because of the assumption that $f_i(x)$ is bounded below for each $i \in \{1, 2, 3, \dots, m\}$, we get

$$|B(\epsilon)| \leq \frac{(f_i(x^0) - f_i^{\text{inf}}) 1}{\rho \epsilon}.$$

\square

2.7 Numerical Experiments

An algorithm is a logical iterative process for finding a solution of a problem. The count of iterations a method takes determines the time to converge to a solution, which may vary from problem to problem. A method can find a solution to some problems very fast and some others in a longer time. As a result, the question arises, how one can

measure the efficiency of any method in order to choose the best method from several competing ones. Three basic approaches [152] which have been widely adopted in the literatures for measuring performance of any algorithm are:

Empirical analysis: In this analysis, we aim to determine how a method behaves in practice. As a part of this analysis, we design a code for the algorithm on computer and examine its performance on several typical test problems.

Average-case analysis: In this analysis, we estimate the expected number of steps that an algorithm takes. We also select a probability distribution for the test problems and determine the expected asymptotic running time.

Worst-case analysis: In this analysis, identifies an upper bound on the total count of iteration that a given algorithm can take for any given problem.

In the current work, we employ empirical analysis to evaluate the performance of the proposed method. To rank a given algorithm, one can evaluate the total count of function and gradient evaluations using the following formula [152]:

$$NT = NF + \bar{m} \cdot NG, \quad (2.39)$$

where NF and NG represents the count of function evaluations and gradient evaluations, respectively, and \bar{m} is an integer. As per the results on automatic differentiation [65], the value of \bar{m} can be set to $\bar{m} = 5$, i.e., one gradient evaluation is equivalent to \bar{m} number of function evaluations in the case of automatic differentiation. For a detailed explanation on the value $m = 5$, we refer to Subsection 3.3 of the reference [65].

This section shows the effectiveness and ability to generate Pareto frontiers for some benchmark multiobjective test problems. We perform our experiments using MATLAB software (version R2018b). For the experiments, we restrict the PRP conjugate parameter to be nonnegative. Thus, the value of β^k used in Algorithm 1 leads to the following

expression:

$$\beta_k = \max \left\{ \beta_k^{\text{PRP}}, 0 \right\}, \text{ where } \beta_k^{\text{PRP}} = \frac{-\mathcal{M}(x^k, d_{SD}(x^k)) + \mathcal{M}(x^{k-1}, d_{SD}(x^k))}{-\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1}))}.$$

As far as the algorithmic parameters are concerned, we used $\sigma = 0.4$, $\delta = 0.3$, $\rho = 0.05$, $v = 0.4$, $L_0 = 2$, and $\bar{S} = 10$. For each starting point, the iteration stops at x^k , satisfying $|\Theta(x^k)| \leq 5 \times \text{eps}^{1/2}$, where $\text{eps} = 2^{-52} \approx 2.22 \times 10^{-16}$.

In Table 2.1, we have listed 32 multiobjective test problems, which includes convex as well as non-convex. The randomly generated starting points is taken from a box $\{x \in \mathbb{R}^n : lb \leq x \leq ub\}$. Table 2.1 contains the significant informations of the test problems. The columns of the Table 2.1 depict the following:

- (i) Problem: the name of the test problem,
- (ii) Source: source reference of the test problem,
- (iii) n : number of decision variables in the test problem,
- (iv) m : number of objectives in the test problem,
- (v) Convex: whether the problem is convex or not (Y for convex and N for nonconvex),
- (vi) lb^\top : vector of the lower bounds of the box from which we choose the starting points, and
- (vii) ub^\top : vector of the upper bounds of the box from which we choose the starting points.

We compare Algorithm 1 with two other methods for MOPs. For comparison, we have considered the following methods:

- (i) HZ: the Hager Zhang method [63] equipped with standard Wolfe line search condition,

- (ii) LS: the Liu-Storey conjugate gradient method [64] under an Armijo-type line search condition, and
- (iii) NMGRP: the proposed PRP conjugate gradient method under an average-type nonmonotone line search conditions given in (2.8) and (2.9), i.e., Algorithm 1.

For each considered method, each test problems were solved for 100 randomly chosen initial points from the box of lower and upper bounds as shown in columns “ lb^\top ” and “ ub^\top ”, respectively. Table 2.2 shows the comparison of Algorithm 1 with HZ and LS methods for MOPs. The columns of Table 2.2 depict the following:

- (i) Problem: the name of test problem,
- (ii) NI : average number of iteration,
- (iii) NF : average number of function evaluation,
- (iv) NG : average number of gradient evaluation, and
- (v) NT : the total number of average function evaluations and gradient evaluations (see (2.39)).

In the performance evaluation, we compare the HZ and LS methods with the NMGRP method as follows: for each i^{th} test problem, we evaluate the total number of average function evaluations and gradient evaluations by the j^{th} solver using the equation (2.39) with $m = 5$, and denote it by $NT(i, j)$. Then, we calculate the ratio $r(i, j) = \frac{NT(i, j)}{NT(i, NMGRP)}$. Geometric mean of these ratio for j^{th} solver over all the test problems is given by

$$r(j) = \left(\prod_{i \in P} r(i, j) \right)^{\frac{1}{|P|}} \quad (2.40)$$

which is also known as *relative efficiency*, where P represents the set of the test problems under consideration and $|P|$ is the cardinality of P .

Problem	n	m	Convex	lb^\top	ub^\top	Source
FON	2	2	N	$(-1, -1)$	$(1, 1)$	[164]
Hill	2	2	N	$(0, 0)$	$(1, 1)$	[74]
SK2	4	2	N	$(-10, \dots, -10)$	$(10, \dots, 10)$	[75]
PNR	2	2	Y	$(-1, -1)$	$(1, 1)$	[126]
MOP3	2	2	N	$(-\pi, -\pi)$	(π, π)	[75]
KW2	2	2	N	$(-3, -3)$	$(3, 3)$	[87]
FAR1	2	2	N	$(-1, -1)$	$(1, 1)$	[98]
FF1	2	2	N	$(-1, -1)$	$(1, 1)$	[75]
VU1	2	2	N	$(-3, -3)$	$(3, 3)$	[75]
VU2	2	2	Y	$(-3, -3)$	$(3, 3)$	[75]
SP1	2	2	Y	$(-10, -10)$	$(10, 10)$	[75]
JOS1	1000	2	Y	$(-10^4, \dots, -10^4)$	$(10^4, \dots, 10^4)$	[75]
AP1	2	3	Y	$(-100, -100)$	$(100, 100)$	[63]
AP3	2	2	N	$(-100, -100)$	$(100, 100)$	[63]
AP4	3	3	Y	$(-100, -100, -100)$	$(100, 100, 100)$	[63]
SLCDT1	2	2	N	$(-5, -5)$	$(5, 5)$	[63]
SLCDT2	10	3	Y	$(-100, -100, \dots, -100)$	$(100, 100, \dots, 100)$	[63]
DD1	5	2	N	$(-20, \dots, -20)$	$(20, \dots, 20)$	[63]
Comet	2	3	N	$(1, -2)$	$(3.5, 2)$	[75]
Viennet	2	3	N	$(-3, -3)$	$(3, 3)$	[75]
DLTZ2	10	3	N	$(0, 0, \dots, 0)$	$(1, 1, \dots, 1)$	[164]
VFM1	2	3	Y	$(-2, -2)$	$(2, 2)$	[75]
IKK1	2	3	Y	$(-50, -50)$	$(50, 50)$	[75]
BK1	2	2	Y	$(-5, -5)$	(1010)	[75]
LRS1	2	2	Y	$(-50, -50)$	$(50, 50)$	[75]
SSFYY1	2	2	Y	$(-100, -100)$	$(100, 100)$	[75]
FDS#1	2000	3	Y	$(-2, -2, \dots, -2)$	$(2, 2, \dots, 2)$	[50]
FDS#2	3000	3	Y	$(-2, -2, \dots, -2)$	$(2, 2, \dots, 2)$	[50]
FDS#3	5000	3	Y	$(-2, -2, \dots, -2)$	$(2, 2, \dots, 2)$	[50]
MMR5#1	2000	2	N	$(-5, -5, \dots, -5)$	$(5, 5, \dots, 5)$	[63]
MMR5#2	3000	2	N	$(-5, -5, \dots, -5)$	$(5, 5, \dots, 5)$	[63]
MMR5#3	5000	2	N	$(-5, -5, \dots, -5)$	$(5, 5, \dots, 5)$	[63]

Table 2.1: List of test problems taken for numerical comparison

Problem	HZ				LS				NMPRP			
	<i>NI</i>	<i>NF</i>	<i>NG</i>	<i>NT</i>	<i>NI</i>	<i>NF</i>	<i>NG</i>	<i>NT</i>	<i>NI</i>	<i>NF</i>	<i>NG</i>	<i>NT</i>
FON	7	54	48	294	6.1	36	28	176	4.3	9.3	13.7	77.8
HILL	9	62	54	332	7	47.1	32.8	211.1	5.8	12.4	18.2	103.7
SK2	29	93	951	568	22.9	87.7	110	637.7	17.3	39.9	52.4	301.9
PNR	3	13.5	13.5	81	3.9	12.6	28.3	154.1	2.9	6.5	9.5	54.2
MOP3	9	35.5	35	210.5	8.5	31.5	31	186.5	12.3	28.3	35.2	204.3
KW2	12	87.5	75	462.5	9.3	66.2	55.9	345.7	17.2	47.5	51.1	303
FAR1	36	208.5	182.5	1121	32.6	145.2	122.5	757.7	30.1	71.3	106.7	605.1
FF1	7.5	53	48	293	7.1	36.5	27.4	173.5	6	12.6	18.6	106.1
VU1	784	2394	2396	14374	312.3	950	988.2	5891	110.7	221.4	115.9	800.9
VU2	6.9	16.2	8.9	60.7	6.9	14.1	7.8	53.1	6.1	11.2	5.1	36.7
SP1	3	6	8	46	9.2	19	16.1	99.5	8.1	19	13.5	86.5
JOS1	1	2	4	22	1	2	4	22	1	2	4	22
AP1	10.5	104	85	529	10.1	52	38.4	244	9	41.8	31.3	198.3
AP3	13.5	129.5	118	719.5	12.3	110	99.2	606	11.2	71.6	63.9	391.1
AP4	11	108	101	613	10.2	97	100	597	9.1	81.7	90	531.7
SLCDT1	4	22.5	22	132.5	4	12.5	18.5	105	4	12.5	18.5	105
SLCDT2	18	190.5	173.5	1058	17.3	168.1	154.5	940.6	14.8	155	128.3	796.5
DD1	74.5	230	232	1390	40.2	51.1	67	386.1	33.9	41.2	51.2	297.2
Comet	74.4	74.4	443.4	2291.4	28.1	35	66	365	11.2	21.8	19.3	118.3
Viennet	127.1	145	762.6	3958	66	101.6	150.4	853.6	31.5	57.1	29.6	205.1
DLTZ2	41.7	161.5	190.3	1113	41.1	157.9	184	1077.9	34	136.2	148.3	877.7
VFM1	5.9	16.2	29	161.2	11	13.6	31	168.6	4.7	9.6	16.5	92.1
IKK1	15.7	33	45.9	262.5	15	32.8	34	202.8	16.9	30.1	21.6	138.1
BK1	6.1	12	8.9	56.5	5.3	11.3	8.1	51.8	5.1	11	7.7	49.5
LRS1	17.7	25	29.8	174	16	25	31	180	12.4	17.1	10.4	69.1
SSFYY1	22.1	36.5	29	181.5	16.2	28	17.9	117.5	8.9	15.2	9	60.2
FDS#1	33	257	234	1427	27.7	251.9	226	1381.9	24.9	243	211	1298
FDS#2	35	262	241	1467	30.2	256.8	238.3	1448.3	32.4	254	241	1459
FDS#3	36	271	248.5	1515.5	32	261	237	1446	40	268.1	260.7	1571.6
MMR5#1	25.5	237.5	218	1327.5	18.7	235	212.1	1295.5	15.5	211	198	1201
MMR5#2	38	370	339	2065	38.6	365.4	300.2	1866.4	31	331	268	1671
MMR5#3	25	244	236	1424	22	231.5	236.4	1413.5	18.7	210.2	213	1275.2

Table 2.2: Performance of HZ, LS and NMPRP methods on the test problems given in Table 2.1

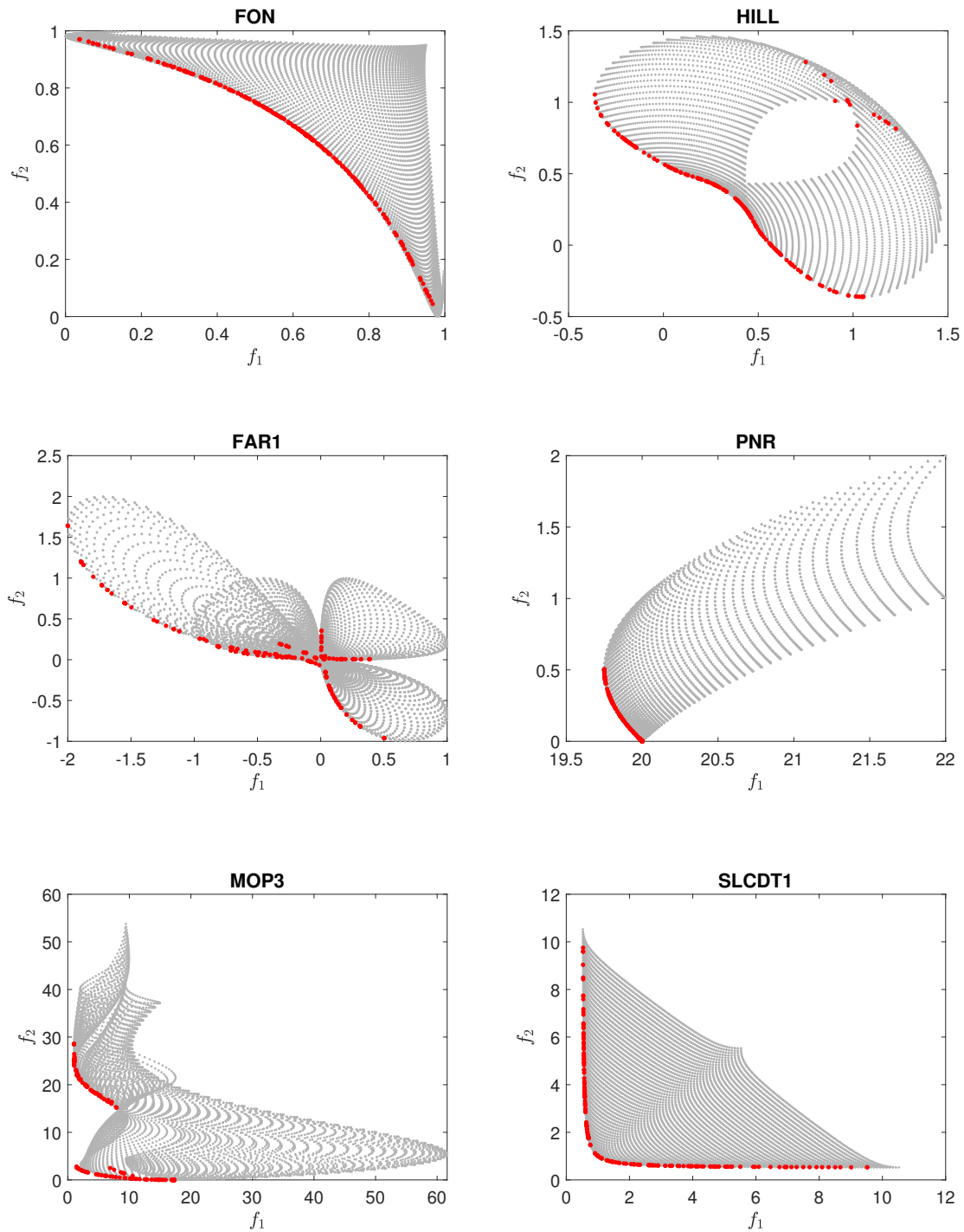


Figure 2.1: Gray points denote the objective feasible region of the problems FON, HILL, FAR1, PNR, MOP3, SLCDT1, FF1, BK1, VU2, DLTZ2, VFM1, IKK1, and Comet. The red points represent the final iterates obtained by the Algorithm 1 using 150 randomly generated starting points.

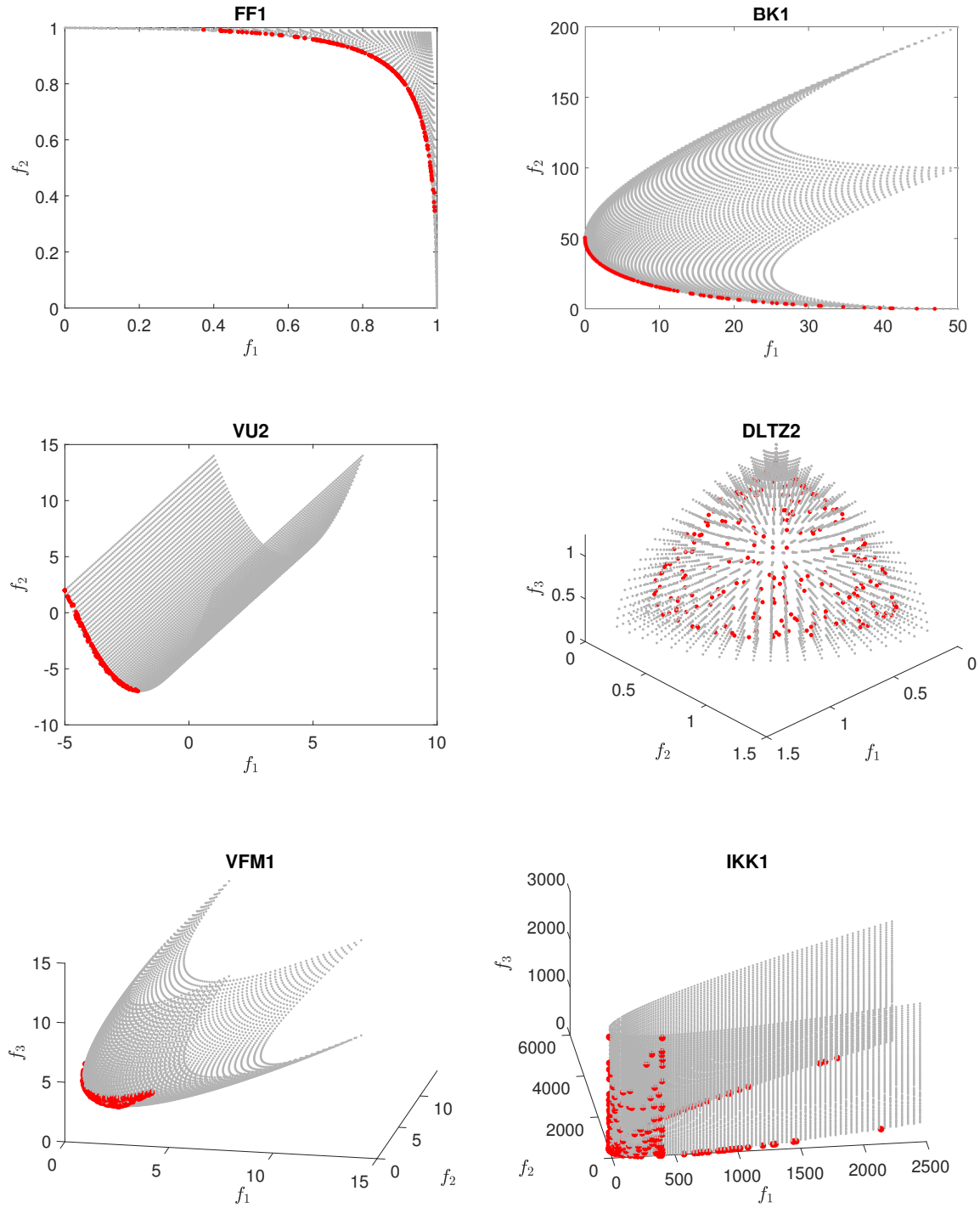


Figure 2.1: Continued from Figure 2.1

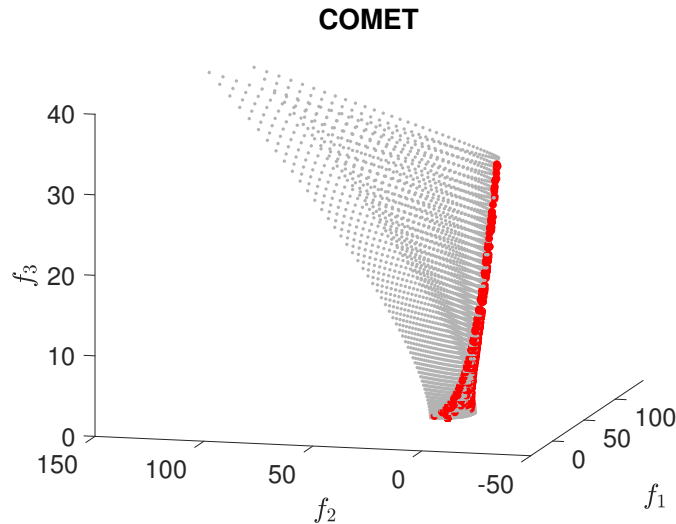


Figure 2.1: Continued from Figure 2.1

HZ	LS	NMPRP
1.990	1.521	1

Table 2.3: Relative efficiency of HZ, LS and NMPRP methods

The comparison of the considered methods is relative and is not dominated by a few problems for which the method takes a large number of function and gradient evaluations. According to (2.40), it is evident that $r(\text{NMPRP}) = 1$. Also, the lesser r , the better in the performance. The values of $r(\text{HZ})$ and $r(\text{LS})$ are mentioned in Table 2.3. Table 2.3 indicates that the NMPRP method produces the best average performances, and the LS method produces better average performances than the HZ method.

Figure 2.1 displays the output of the test results obtained by the Algorithm 1. To check the ability of generating the Pareto critical points by the proposed method, we have considered thirteen multiobjective test problems of both convex and non-convex type. For each test problems, the gray points in Figure 2.1 represent the points of feasible criterion space, which is achieved by discretizing the respective boxes $[lb^\top, ub^\top]$

through fine grid and plotting all the image points. These plots give us with accurate idea of the image spaces of F and give us a geometric location of the Pareto front of F . The red points were obtained by executing the proposed algorithm for each considered problem using 150 times using randomly obtained initial points from the respective box $[lb^\top, ub^\top]$. Figure 2.1 depicts that for a given test problems, considering a pertinent number of starting points, the Algorithm 1 can generate a satisfying outline of the Pareto front. From Figure 2.1, we can notice that for each test problem, a starting point has reached a critical point for almost all the iterations, which depicts their robustness. Even then, failures may occur. For instance, in problems like HILL, FAR1, and MOP3, in a few runs, the iteration terminated at a local (nonglobal) Pareto optimal point which is due to the error occurs while calculating the SD direction $d_{SD}(x)$.

Based on the test results reported in Table 2.2, we show the performance profile given by Dolan and Moré [37] to compare the performance of the considered methods. If S and P denote the set of solvers and problems, respectively, then the performance profile $\mathcal{P} : \mathbb{R} \rightarrow [0, 1]$ is measure by the following rule: for a given problem $p \in P$ and a given solver $s \in S$, we define $NI_{p,s}$ by average number of iterations need to solve the problem p by solver s . The performance ratio measure for the average number of iterations is given by the following formula:

$$r_{p,s} = \frac{NI_{p,s}}{\min_{s \in S} NI_{p,s}}.$$

Then, the performance profile measure for the average number of iterations is defined by the following formula:

$$\mathcal{P}(\varrho) = \frac{1}{|P|} \text{size}\{p \in P : r_{p,s} \leq \varrho\} \text{ for all } \varrho \in \mathbb{R},$$

where $\text{size}\{B\}$ denotes the number of elements in the set B . Similarly, we can also define the performance profile measure by NF , NG and NT . Note that $\mathcal{P}(\varrho)$ is a probability

for a solver $s \in S$ that a performance ratio $r_{p,s}$ is within a factor $\varrho \in \mathbb{R}$ of the best possible ratio.

Figures 2.2a-2.2d show that NMPRP method outperform the HZ and LS method for all the four scenarios, i.e., NI , NF , NG and NT .

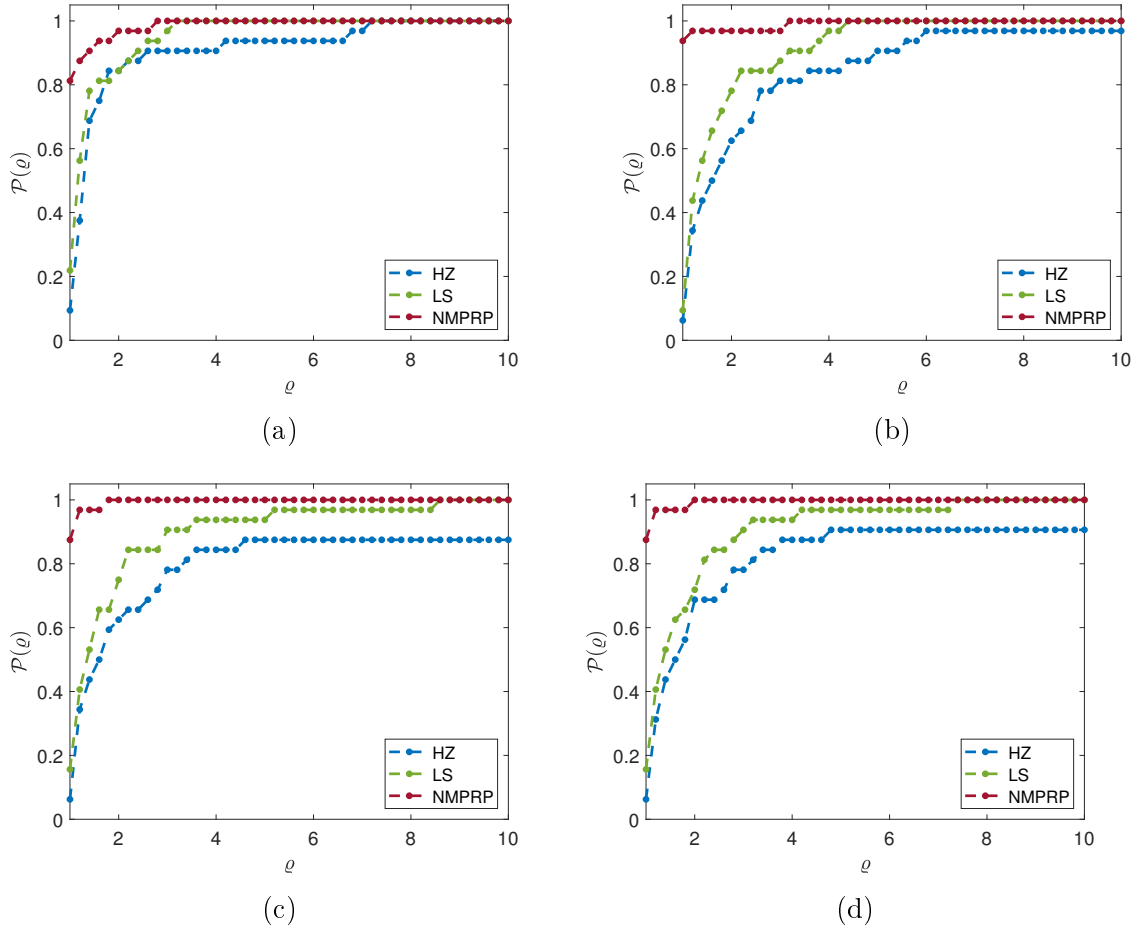


Figure 2.2: Performance profile measured by (a) NI , (b) NF , (c) NG , and (d) NT of 32 test problems listed in Table 2.1

2.7.1 Performance Metrics

For Algorithm 1, two performance metrics, namely, HV and IGD [9] are used to assess the quality of the obtained solution set in terms of optimality and diversity. For the IGD value, the lesser, the better. For the HV value, the bigger, the better.

Table 2.4 reports the considered test problems. In Table 2.4, the parameters m and

n is the number of objectives and decision variables, respectively. “ lb^\top ” represents the lower bound vector of the decision variable $x \in \mathbb{R}^n$, and “ ub^\top ” denotes the upper bound vector of the decision variable $x \in \mathbb{R}^n$. The column “nondominated set type” indicates whether the nondominated set of the test problem is convex or not. The column “nondominated set location” provides the information whether the nondominated solution is a priori known or not. The column “Source” provides the source of the test problems.

Problem	m	n	lb^\top	ub^\top	Nondominated set type	Nondominated set location	Source
ZDT1	2	10	(0,0)	(1,1)	Convex	Known	[164]
ZDT2	2	10	(0,0)	(1,1)	Non-convex	Known	[164]
ZDT3	2	10	(0,0)	(1,1)	Nonconvex	Known	[164]
ZDT4	2	10	(0,-5)	(1,5)	Convex	Known	[164]
DTLZ1	3	5	(0,0,0)	(1,1,1)	Convex	Known	[164]
DTLZ2	3	5	(0,0,0)	(1,1,1)	Nonconvex	Known	[164]
DTLZ3	3	5	(0,0,0)	(1,1,1)	Nonconvex	Known	[164]
DTLZ5	3	5	(0,0,0)	(1,1,1)	Nonconvex	Known	[61]

Table 2.4: Data for the ZDT and DTLZ benchmark suite to check the performance of NMPRP method

Problem	MOEA/D (WS)	MOEA/D (TE)	MOEA/D (PBI)	NSGA-II	$pa\lambda$ -MOEA/D	NMPRP
ZDT1	$5.42E-4$	$6.84E-4$	$1.14E-3$	$7.94E-4$	$5.80E-4$	$1.12E-4$
ZDT2	$1.30E-2$	$5.84E-4$	$7.03E-4$	$8.16E-4$	$6.02E-4$	$3.97E-6$
ZDT3	$4.93E-3$	$2.01E-3$	$2.06E-3$	$1.20E-3$	$1.97E-3$	$4.16E-4$
ZDT4	$7.02E-3$	$6.54E-4$	$7.86E-4$	$8.14E-4$	$5.94E-4$	$2.01E-4$
DTLZ1	$4.03E-3$	$6.93E-4$	$4.21E-4$	$7.91E-4$	$4.52E-4$	$3.97E-6$
DTLZ2	$5.35E-3$	$7.42E-4$	$6.15E-4$	$7.58E-4$	$5.78E-4$	$5.12E-6$
DTLZ3	$1.42E-2$	$1.24E-3$	$1.90E-3$	$3.59E-3$	$1.18E-3$	$1.12E-3$
DTLZ5	$1.38E-3$	$5.22E-5$	$1.11E-4$	$1.89E-5$	$2.79E-5$	$1.08E-5$

Table 2.5: Median of IGD values for ZDT and DTLZ benchmark suite obtained by different algorithms

For ZDT and DTLZ test suits (listed in Table 2.4), the median of the IGD values and HV values of the generated solution sets by Algorithm 1 and other existing efficient solvers are given in Table 2.5 and Table 2.6, respectively. From Table 2.5, we see that NMPRP method has least values for all the test problems. From Table 2.6, we observe that the NMPRP has greatest values for all the test problems. Hence, NMPRP method

Problem	MOEA/D (WS)	MOEA/D (TE)	MOEA/D (PBI)	NSGA-II	$pa\lambda$ -MOEA/D	NMPPR
ZDT1	0.6521	0.6392	0.6057	0.6381	0.6412	0.6901
ZDT2	0.0000	0.3097	0.2957	0.3060	0.3107	0.3177
ZDT3	0.4863	0.4807	0.4642	0.5066	0.4873	0.6004
ZDT4	0.3534	0.6360	0.6244	0.6359	0.6391	0.7101
DTLZ1	0.2259	0.7434	0.7835	0.7262	0.7814	0.7921
DTLZ2	0.0000	0.3777	0.3812	0.3766	0.4074	0.5109
DTLZ3	0.0000	0.3605	0.2633	0.1901	0.3817	0.4021
DTLZ5	0.0000	0.0894	0.0779	0.0930	0.0916	0.1921

Table 2.6: Median of HV values for ZDT and DTLZ benchmark suite obtained by different algorithms

outperforms the existing efficient methods.

As per the results obtained in Table 2.5, we also measure the performance of the Algorithm 1 through relative efficiency. We compare the MOEA/D(WS), MOEA/D(TE), MOEA/D(PBI), NSGA-II and $pa\lambda$ -MOEA/D methods with NMPPR method as follows: for each i^{th} test problem, we compute the median of IGD values by the j^{th} solver and denote it by $\text{MIGD}(i, j)$, then we evaluate the ratio by

$$r(i, j) = \frac{\text{MIGD}(i, j)}{\text{MIGD}(i, \text{NMPPR})}.$$

Geometric mean of these ratio for j^{th} solver for all the problems of Table 2.1 is evaluated by (2.40) also known as relative efficiency.

MOEA/D(WS)	MOEA/D(TE)	MOEA/D(PBI)	NSGA-II	$pa\lambda$ -MOEA/D	NMPPR
101.504	14.459	17.224	15.176	11.831	1

Table 2.7: Relative efficiency of MOEA/D(WS), MOEA/D(TE), MOEA/D(PBI), NSGA-II, $pa\lambda$ -MOEA/D and NMPPR methods with respect to median of IGD values

According to the rule given in (2.40), it is evident that $r(\text{NMPPR}) = 1$. The

values of $r(\text{MOEA/D(WS)})$, $r(\text{MOEA/D(TE)})$, $r(\text{MOEA/D(PBI)})$, $r(\text{NSGA-II})$ and $r(\text{pa}\lambda\text{-MOEA/D})$ are mentioned in Table 2.7. Table 2.7 indicates that the NMPRP method produces the best median of IGD values among all the considered methods.

On the basis of median HV values as mentioned in Table 2.6, we also calculate the relative efficiency with respect to median of HV values. Table 2.8 indicates that the NMPRP method produces the best median of HV values among all the considered methods.

MOEA/D(WS)	MOEA/D(TE)	MOEA/D(PBI)	NSGA-II	pa λ -MOEA/D	NMPRP
0	0.810	0.757	0.753	0.834	1

Table 2.8: Relative efficiency of MOEA/D(WS), MOEA/D(TE), MOEA/D(PBI), NSGA-II, pa λ -MOEA/D and NMPRP methods with respect to median of HV values

2.8 Conclusion

This chapter has presented a nonmonotone PRP conjugate gradient algorithm which is useful in solving MOPs without depending on the convexity assumption on the objectives. The novelty of Algorithm 1 is that an average-type Armijo-like nonmonotone line search procedure has been implemented instead of the monotone line search strategies. We have shown in Theorem 2.1 that under a few common assumptions, Pareto critical point is the limit of a sequence of iterates obtained through the proposed method. The complexity bounds of Algorithm 1 has been established in Theorem 2.2. To check the efficiency of Algorithm 1 over other methods, we have considered the HZ and LS methods. We have computed the relative efficiency (see Table 2.2 and 2.3) of the proposed method through an empirical analysis. We have also shown the performance profile measure given by Dolan and Moré [37] to compare the performance of the considered methods (see Figures 2.2a-2.2d). Algorithm 1 has been compared with some existing popular solvers based on IGD and HV indicators. The numerical results depict that the

average-type nonmonotone PRP conjugate gradient method outperform the HZ and LS methods for the considered multiobjective optimization test problems.

As part of our further research, we will be focusing on studying the theoretical properties and experimental performance of different conjugate gradient methods under a nonmonotone line search setting for solving MOPs.

2.8.1 Pros of Algorithm 1

- The proposed method (Algorithm 1) can generate the convex as well as nonconvex portion of the nondominated set.
- The method does not require prior information about the location of the nondominated set.
- Algorithm 1 is capable to solve the convex and non-convex MOPs.
- The asymptotic convergence of Algorithm 1 has been proved without the convexity assumption of the objectives.
- NMPRP method satisfactorily estimates the Pareto front of the problems when there are a sufficient number of starting points.

2.8.2 Cons of Algorithm 1

- The method require the objectives and constraint functions to be continuously differentiable. When the objective functions and constrains do not fulfill this requirement of continuous differentiability then the proposed algorithm will not be applicable.
- For a few problems like HILL, FAR1, and MOP3, the proposed method may also generate the local (nonglobal) Pareto optimal points.
