Chapter 4

Nonmonotone Wolfe-Type Quasi-Newton Methods for MOPs

4.1 Introduction

Quasi-Newton algorithms are widely regarded as one of the most effective methods for solving unconstrained singleobjective optimization problems. Their development dates back to 1959 with the work of Davidon [34], and they gained widespread recognition through the contributions of Fletcher and Powell in 1963 [48]. Since then, these algorithms have garnered significant attention due to their practical efficiency and the fact that they bypass the need for calculating second derivatives. Numerous papers have been published on this topic, authored by leading figures in the optimization field. In the quasi-Newton approach, the search direction is determined using a quadratic approximation of the objective function, where the Hessian matrix is approximated and updated iteratively. The BFGS update formula, introduced in 1970 by Broyden, Fletcher, Goldfarb, and Shanno, remains the most commonly used quasi-Newton scheme. With appropriate assumptions and a suitable line search, the BFGS algorithm achieves global and superlinear convergence for strongly convex problems [17, 127], though it may fail to converge when applied to general nonconvex problems [26, 28].

The BFGS method was first introduced for multiobjective optimization in [123] and further explored in works such as [90, 102, 112, 131]. Much like in singleobjective optimization, the search direction is determined by solving a problem that uses quadratic models of the objective functions, with BFGS updates serving as approximations of the true Hessians.

4.2 Motivation

The Wolfe conditions are essential in the quasi-Newton method as they guide the selection of the step size during each iteration, balancing the reduction of the objective function and the satisfaction of the curvature condition. This balance ensures that the method progresses effectively toward a stationary point, even in nonmonotone scenarios where the objective function does not consistently decrease. By satisfying the Wolfe conditions, the method avoids overly large steps that could cause divergence and overly small steps that might slow convergence, thus enhancing the overall stability and efficiency of the optimization process [116].

4.3 Contribution

The present chapter can be summarized by the following key points:

- This study presents a novel BFGS algorithm equipped with two types of nonmonotone Wolfe line search strategy specifically designed for addressing unconstrained strongly convex MOPs.

- In this work, we incorporate step sizes that adhere to the nonmonotone Wolfe line search conditions for MOPs. For the scalar case of nonmonotone Wolfe line search, a reference worth considering is [156]. Additionally, to explore its vector extension, [98] serves as an authoritative source.

- This study extensively examines the utilization of both max-type and average-type nonmonotone line searches, resulting in the identification of two approaches discussed within the study.

- We rigorously establish the convergence properties of both approaches, taking into account some common assumptions.

- For both approaches, we show that any limit point generated by a sequence of iterates is a Pareto critical.

- Our focus in the theoretical findings revolves around strongly convex problems, but for numerical experiments, we have tested strongly convex and nonconvex problems as well using an appropriate modification in the Hessian approximation matrix.

- To evaluate the comparative efficacy of the proposed methods, which employ two nonmonotone line searches, we conduct a comparison with a monotone line search algorithm through relative efficiency.

- In order to conduct a thorough comparison of Algorithms 5 and 6, we employ Dolan and Moré performance profiles. These profiles allow us to assess and compare the performance of various algorithms when applied to a specific set of test problems [37].

Suppose we have a set $K = \{k_1, k_2, k_3, \ldots\} \subseteq \mathbb{N}$, where $k_j < k_{j+1}$ for all $j \in \mathbb{N}$. In this case, we use the notation $K \underset{\infty}{\subset} \mathbb{N}$ to indicate that K is an infinite subset of $\mathbb{N}$. If $\mathscr{B} \in \mathbb{R}^{n \times n}$, $\mathscr{B} \succ 0$ (or $\mathscr{B} \prec 0$) indicates that $\mathscr{B}$ is positive (or negative) definite. The Hessian of a function $F : \mathbb{R}^n \to \mathbb{R}$ with respect to x at a point $\bar{x} \in \mathbb{R}^n$ is an $n \times n$ matrix denoted by $\nabla^2 F(\bar{x})$.

This study focuses on analyzing an unconstrained MOP, as given by

$$\min_{z\in\mathbb{R}^n} F(z) = (f_1(x), f_2(x), \ldots, f_r(x))^\top, \tag{4.1}$$

where $F : \mathbb{R}^n \to \mathbb{R}^r$ is a stongly convex continuously differentiable function.
The function $F : \mathbb{R}^n \to \mathbb{R}^r$ is regarded as convex (strongly convex) if its components $f_j : \mathbb{R}^n \to \mathbb{R}$ are convex (strongly convex), $\forall j = 1, 2, \ldots, r$.

In the context of quasi-Newton methods, the search direction $w(x)$ for a given $x \in \mathbb{R}^n$ is defined by the solution of the following problem

$$\min_{w\in\mathbb{R}^n} \max_{j=1,2,\ldots,r} \langle \nabla f_j(x), w\rangle + \tfrac{1}{2} w^\top \mathscr{B}_j w, \tag{4.2}$$

where $\mathscr{B}_j \in \mathbb{R}^{n\times n}$ is an approximation of the Hessian, $\nabla^2 f_j(x)$, $j = 1, 2, \ldots, r$ (see [123]). If $\mathscr{B}_j \succ 0$ $\forall j = 1, 2, \ldots, r$, then (4.2) is a strongly convex optimization problem. It follows that (4.2) has a unique optimal solution. We denote the solution of (4.2) by $w(x)$ and its optimal value by $\Psi(x)$, i.e.,

$$w(x) = \arg\min_{w\in\mathbb{R}^n} \max_{j=1,2,\ldots,r} \langle \nabla f_j(x), w\rangle + \tfrac{1}{2} w^\top \mathscr{B}_j w, \tag{4.3}$$

and

$$\Psi(x) = \max_{j=1,2,\ldots,r} \langle \nabla f_j(x), w(x)\rangle + \tfrac{1}{2} w(x)^\top \mathscr{B}_j w(x). \tag{4.4}$$

Notice that if $\mathscr{B}_j = I_n$ $\forall j = 1, 2, \ldots, r$, then the direction $w(x)$ is a steepest descent direction [49]. Also, if $\mathscr{B}_j = \nabla^2 f_j(x)$ $\forall j = 1, 2, \ldots, r$, then $w(x)$ turns out to be the Newton direction, see [50].

If we assume that $\mathscr{B}_j \succ 0$ $\forall j = 1, 2, \ldots, r$, then the problem (4.2) can be reformulated as the following convex quadratic optimization problem:

$$\begin{aligned}
&\min_{(t,w)\in\mathbb{R}\times\mathbb{R}^n} \ t \\
&\text{subject to } \nabla f_j(x)^\top w + \tfrac{1}{2} w^\top \mathscr{B}_j w \le t \ \ \forall j = 1, 2, \ldots, r.
\end{aligned} \tag{4.5}$$

Clearly, the unique solution of (4.5) is $(t, w) = (\Psi(x), w(x))$. As the problem (4.5) is convex with a slater point (e.g., $(1, 0) \in \mathbb{R} \times \mathbb{R}^n$), there exists a multiplier $\lambda(x) \in \mathbb{R}^r$ such that the triple $(t, w, \lambda) = (\Psi(x), w(x), \lambda(x)) \in \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^r$ satisfies the Karush-Kuhn-Tucker conditions of the problem (4.5):

$$\sum_{j=1}^{r} \lambda_j \left[\nabla f_j(x) + \mathscr{B}_j w\right] = 0, \quad \sum_{j=1}^{r} \lambda_j = 1,$$

and

$$\lambda_j \ge 0,\ \nabla f_j(x)^\top w + \tfrac{1}{2} w^\top \mathscr{B}_j w \le t,\ \lambda_j \left[\nabla f_j(x)^\top w + \tfrac{1}{2} w^\top \mathscr{B}_j w - t\right] = 0, \forall j = 1, 2, \ldots, r.$$

Therefore, in particular, we have

$$\begin{aligned}
w(x) &= -\left[\sum_{j=1}^{r} \lambda_j(x) \mathscr{B}_j\right]^{-1} \sum_{j=1}^{r} \lambda_j(x) \nabla f_j(x), \\
&\sum_{j=1}^{r} \lambda_j(x) = 1, \quad \lambda_j(x) \ge 0, \quad \forall j = 1, 2, \ldots, r, \\
\text{and } \Psi(x) &= -\tfrac{1}{2} w(x)^\top \left[\sum_{j=1}^{r} \lambda_j(x) \mathscr{B}_j\right] w(x).
\end{aligned}$$

The following lemma demonstrates that the direction $w(x)$ and the optimal value $\Psi(x)$ can serve as important factors for identifying Pareto critical points of (4.1).

Lemma 4.1 (See [49, Lemma 3.3]). Let $w : \mathbb{R}^n \to \mathbb{R}^n$ and $\Psi : \mathbb{R}^n \to \mathbb{R}$ be given by (4.3) and (4.4), respectively. Assume that $\mathscr{B}_j \succ 0\ \forall j = 1, 2, \ldots, r$. Then, the following hold:

(i) $\hat{x}$ is Pareto critical if and only if $w(\hat{x}) = 0$ and $\Psi(\hat{x}) = 0$,

(ii) if $\hat{x}$ is not Pareto critical, then $w(\hat{x}) \neq 0$, $\Psi(\hat{x}) < 0$, $\mathcal{M}(\hat{x}, w(\hat{x})) < \Psi(\hat{x}) < 0$, and $w(\hat{x})$ is a descent direction for F at $\hat{x}$.

The BFGS method [116] is a popular quasi-Newton method for singleobjective minimization problems with the objective function $f(x)$, where f is a mapping from $\mathbb{R}^n$ to $\mathbb{R}$. The BFGS method combines a descent direction with a line search strategy, where the descent direction, at $k-$th iteration, is denoted as

$$w(x^k) = -\mathscr{B}(x^k)^{-1}\nabla f(x^k).$$

In this expression, $\mathscr{B}(x^k)$ represents an $n \times n$ symmetric positive definite matrix that is updated at each iteration. For convenience, we denote $\mathscr{B}(x^k), w(x^k), f(x^k)$ to $\mathscr{B}^k, w^k, f^k$, respectively. The new iterate in BFGS method for the singleobjective optimization problem is given by

$$x^{k+1} = x^k + \alpha_k w^k,$$

where $\alpha_k \geq 0$ is the line search step length, and the BFGS update formula is expressed as

$$\mathscr{B}^{k+1} = \mathscr{B}^k - \frac{\mathscr{B}^k l^k l^{k\top} \mathscr{B}^k}{l^{k\top} \mathscr{B}^k l^k} + \frac{y^k y^{k\top}}{l^{k\top} y^k},$$

where

- $l^k = x^{k+1} - x^k$ is the step vector, and
- $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ is the gradient difference vector.

According to [116], it has been demonstrated that $\mathscr{B}^{k+1}$ remains positive definite under the condition that $\mathscr{B}^k$ is positive definite and $l_k^\top y^k > 0$. The satisfaction of the latter condition relies on the strong convexity of the function f, which implies that for all $x, y \in \mathbb{R}^n$, the following holds

$$(\nabla f(x) - \nabla f(y))^\top (x - y) \geq a\|x - y\|_2^2,$$

for some $a > 0$.

4.4 Nonmonotone Wolfe Line Search

This section gives a brief explanation of classical nonmonotone line search procedures and the proposed nonmonotone line search for the MOP (4.1). Several studies have shown that the 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 result has a lesser objective function value. In nonmonotone line search, a certain increase in the objective function values is permitted. Specifically, we select $\alpha_k > 0$ such that it satisfies the following conditions:

$$F(x^k + \alpha_k w^k) \preceq C^k + b_1 \alpha_k \mathcal{M}(x^k, w^k) e,$$

where $e = (1, 1, \ldots, 1)^\top \in \mathbb{R}^r$, $b_1 \in (0, 1)$ and $C^k \succeq F(x^k)$. Two different rules for updating C^k lead 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 that is based on the maximum of recent objective function values of the previous iterations. The idea of an 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 previous objective function values.

Now we recall the line search techniques for vector-valued functions as discussed in [98]. Let $w \in \mathbb{R}^n$ be a descent direction for F at x. We say that $\alpha > 0$ is obtained by means of *exact line search* if

$$\mathcal{M}(x + \alpha w, w) = 0.$$

The Wolfe-like *inexact line search* conditions for vector case are classified into two categories:

(i)*Standard Wolfe line search* and

(ii)*Strong Wolfe line search.*

Let $w \in \mathbb{R}^n$ be a descent direction for F at x and $0 < b_1 < b_2 < 1$. We say that $\alpha > 0$ satisfies the *standard Wolfe condition* if

$$F(x + \alpha w) \preceq F(x) + b_1 \alpha \mathcal{M}(x, w) e \quad \text{and} \quad \mathcal{M}(x + \alpha w, w) \geq b_2 \mathcal{M}(x, w), \tag{4.6}$$

and $\alpha > 0$ satisfies the *strong Wolfe condition* if

$$F(x + \alpha w) \preceq F(x) + b_1 \alpha \mathcal{M}(x, w) e \quad \text{and} \quad \left|\mathcal{M}(x + \alpha w, w)\right| \leq b_2 \left|\mathcal{M}(x, w)\right|. \tag{4.7}$$

The nonmonotone algorithms have been proven to be effective in solving highly nonlinear and possibly ill-conditioned problems. Therefore, in our work, we have considered both the average-type nonmonotone line search technique and the max-type nonmonotone line search version of the standard Wolfe line search.

An average-type nonmonotone version of the standard Wolfe line search (4.6) requires step size α that satisfies

$$F(x^k + \alpha w^k) \preceq C^k + b_1 \alpha \mathcal{M}(x^k, w^k) e \text{ and} \tag{4.8}$$

$$\mathcal{M}(x^k + \alpha w^k, w^k) \geq b_2 \mathcal{M}(x^k, w^k) \tag{4.9}$$

with $0 < b_1 < b_2 < 1$, and $C^0 = F(x^0)$. C^k is updated according to the following rule (see [111]):

$$\left.\begin{aligned} C^{k+1} &= \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} F(x^{k+1}) \text{ and} \\ m_{k+1} &= \sigma m_k + 1, \text{ where } m_0 = 1 \text{ and } \sigma \in [0,1]. \end{aligned}\right\} \tag{4.10}$$

A max-type nonmonotone version of the standard Wolfe line search (4.6) requires step size α that satisfies (4.8) and (4.9) with $0 < b_1 < b_2 < 1$, and $C^0 = F(x^0)$. In order to give the update of C^k for max-type, we first introduce a nonnegative integer A. Then, the update of C^k is given by

$$C_j^k = \max_{0 \le i \le a(k)} f_j(x^{k-i}), \quad j = 1, 2, \ldots, r, \tag{4.11}$$

where $a(k) = \min\{k, A\}$ (see [111]).

The following Algorithm 5 formalizes a quasi-Newton method equipped with *average-type nonmonotone* line search for MOP (4.1).

Algorithm 5 An average-type nonmonotone Wolfe-type quasi-Newton method for MOP (4.1)

Aim: To generate a discrete approximation to the complete set of Pareto critical points of problem (4.1)

1: Provide $F = \big(f_1, f_2, \ldots, f_r\big)$, where each f_j for $j = 1, 2, \ldots, r$ is strongly convex, twice continuously differentiable and bounded below function

2: Choose arbitrarily $0 < b_1 < b_2 < 1$, $\sigma \in [0, 1], x^0 \in \mathbb{R}^n, \Gamma_1, \Gamma_2 \in (0, 1)$ and a positive definite initial matrix $\mathscr{B}_j(x^0) = I_{n\times n}$ for $j = 1, 2, \ldots, r$

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 (4.1)

5: Set Pareto set $\mathcal{S} \leftarrow \emptyset$

6: **for** $n = 1 : \mathcal{N}$ **do**

7: Choose a random point $x^0 \in \mathbb{R}^n$

8: Set $C^0 \leftarrow F(x^0)$, $s_0 \leftarrow 1$, $k \leftarrow 0$

9: (Generation of search direction $w(x^k)$ and $\Psi(x^k)$) Calculate

$$w(x^k) = \underset{w\in\mathbb{R}^n}{\arg\min} \max_{j=1,2,\ldots,r} \langle \nabla f_j(x^k), w\rangle + \tfrac{1}{2} w^\top \mathscr{B}_j(x^k) w \text{ and}$$
$$\Psi(x^k) = \max_{j=1,2,\ldots,r} \langle \nabla f_j(x^k), w(x^k)\rangle + \tfrac{1}{2} w(x^k)^\top \mathscr{B}_j(x^k) w(x^k)$$

10: **while** $|\Psi(x^k)| > \epsilon$ **do**

11: Set $\alpha \leftarrow 1$

12: **while** $F(x^k + \alpha w(x^k)) > C^k + b_1 \alpha \mathcal{M}(x^k, w(x^k)) e$, or $\mathcal{M}(x^k + \alpha w(x^k), w(x^k)) < b_2 \mathcal{M}(x^k, w(x^k))$ **do**

13: $\alpha \in [\Gamma_1 \alpha, \Gamma_2 \alpha]$

14: **end while**

15: Output α

16: Set $\alpha_k \leftarrow \alpha$

17: Set $x^{k+1} \leftarrow x^k + \alpha_k w^k$

18: Update the Hessian approximation matrices $\mathscr{B}_j(x^k)$ $\forall j = 1, 2, \ldots, r$ by

$$\mathscr{B}_j(x^{k+1}) \leftarrow \mathscr{B}_j(x^k) - \frac{\mathscr{B}_j(x^k) l^k l^{k\top} \mathscr{B}_j(x^k)}{l^{k\top} \mathscr{B}_k(x^k) l^k} + \frac{y_j^k y_j^{k\top}}{l^{k\top} y_j^k}, \tag{4.12}$$

where $l^k \leftarrow x^{k+1} - x^k$ and $y_j^k \leftarrow \nabla f_j(x^{k+1}) - \nabla f_j(x^k), \forall j = 1, 2, \ldots, r$

19: Update m_{k+1} and C^{k+1} as follows:

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

20: Set $k \leftarrow k + 1$, and go to 9

21: **end while**

22: **return** $\bar{x} = x^k$ as a Pareto critical point

23: Update set $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{F(\bar{x})\}$

24: **end for**

25: **return** $\mathcal{S}$ as a discrete approximation of the complete Pareto set of problem (4.1)

An average-type nonmonotone version of Wolfe line search condition has been incorporated to generate the step length. In average-type nonmonotone line searches, C^{k+1} is a convex combination of C^k and $F(x^{k+1})$. Since $C^0 = F(x^0)$, C^k acts as a convex combination of all the function values $F(x^0), F(x^1), F(x^2), \ldots, F(x^k)$. The degree of nonmonotonicity in C^k is controlled by the parameter σ. When $\sigma = 0$, we get C^k $=F(x^k)$, resulting in the nonmonotone line search being reduced to the monotone one.

The following Algorithm 6 is a max-type version of Algorithm 5, which is obtained by incorporating slight modifications to Step 19 in Algorithm 5 and by introducing a new nonnegative integer A. Thus, we present only Step 19 in Algorithm 6.

Algorithm 6 A max-type nonmonotone Wolfe-type quasi-Newton method for MOP (4.1)

19: Let $a(k) = \min(k, A)$ and set

$$C_j^k = \max_{0 \le i \le a(k)} f_j(x^{k-i}), \quad j = 1, 2, \ldots, r.$$

It is important to note that Algorithms 5 and 6 either stop at a Pareto critical point or produce an infinite sequence x^k of non Pareto critical points. We proceed further, assuming that the algorithms iterates infinitely.

We now recall a useful property of the vector C^k, defined in (4.10). The following result shows that for each j, $f_j(x^k)$ and $P_j^k = \frac{1}{k+1}\sum_{i=0}^{k} f_j(x^i)$ represent lower and upper bounds of C_j^k, respectively, where C_j^k are the components of C^k $\forall j = 1, 2, \ldots, r$.

Lemma 4.2 *Assume that $x^k \in \Omega$ is a non Pareto critical point generated by Algorithm 5. Then, we have $F(x^k) \preceq C^k \preceq P^k$, where $P^k = \frac{1}{k+1}\sum_{i=0}^{k} F(x^i)$.*

Proof: Let us define a function $\mathcal{L}^k : \mathbb{R} \setminus \{-1\} \to \mathbb{R}^r$ by

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

Its derivative is given by

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

As x^k is not a Pareto critical point, we have $\mathcal{M}(x^{k-1}, w^{k-1}) < 0$. It follows from the condition (4.8) that $F(x^k) \preceq C^{k-1}$. This implies $\frac{d\mathcal{L}^k(t)}{dt} \succeq 0$ for all $t \neq -1$. Thus, $\mathcal{L}^k$ is nondecreasing for all $t \geq 0$. Using the facts that $\sigma \in [0,1]$ and $s_0 = 1$, we have $m_k \geq 1$ for all $k = 0, 1, 2, \ldots$. Hence, $\sigma s_{k-1} \geq 0$ holds, and we obtain

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

Now, we show $C^k \preceq P^k$ by induction. For $k = 0$, the inequality trivially holds because $C^0 = P^0 = F(x^0)$. Let $C^i \preceq P^i$ for all $0 \leq i \leq k-1$. As $\sigma \in [0,1]$ and $s_0 = 1$, from (4.10) we can write

$$m_k = 1 + \sum_{\mathcal{I}=1}^{k} \sigma^{\mathcal{I}} \leq 1 + k \quad \text{for all } k > 0. \tag{4.13}$$

Thus, $0 \leq m_k - 1 \leq k$ holds. Since $\mathcal{L}^k$ is nondecreasing for all $t \geq 0$ and $m_k = \sigma s_{k-1} + 1$, we obtain

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

Through the induction, we also have

$$\mathcal{L}^k(k) = \frac{1}{(k+1)}\big(kC^{k-1} + F(x^k)\big) \preceq \frac{1}{(k+1)}\big(kP^{k-1} + F(x^k)\big) = P^k,$$

and the conclusion follows. $\square$

The next lemma demonstrates an important inequality of vector C^k defined in (4.11).

Lemma 4.3 *At each iteration k of Algorithm 6, it holds that $f_j(x^k) \leq C_j^k$.*

Proof: This result can be readily derived from the definition of C^k in equation (4.11). $\square$

Next, we show that Algorithm 5 and Algorithm 6 are well-defined. The sense is that there is always a step size satisfying the line search conditions (4.8) and (4.9) so that the iterates x^k can be generated and $\mathscr{B}_j^{k+1}$ is positive definite whenever $\mathscr{B}_j^k$ is positive definite for each $j = 1, 2, \dots r$.

From now on, we assume that the level set $\mathcal{T} = \{x : F(x) \preceq F(x^0)\}$ is bounded, where $x^0 \in \mathbb{R}^n$ (initial point).

Before showing the well-definedness, we first establish a lemma that states that there is always a step size satisfying the line search conditions (4.8) and (4.9).

Lemma 4.4 *Assume that w^k is a descent direction for F at x^k. Let $0 < b_1 < b_2 < 1$, and $e = (1, 1, \dots, 1)^\top \in \mathbb{R}^r$. Then, $\exists\ \bar{\alpha} \in (0, 1]$ such that $\forall\ \alpha \in (0, \bar{\alpha}]$*

$$F(x^k + \alpha w^k) \preceq C^k + b_1 \alpha \mathcal{M}(x^k, w^k) e \quad \text{and} \quad \mathcal{M}(x^k + \alpha w^k, w^k) \geq b_2 \mathcal{M}(x^k, w^k).$$

Proof: Let us first define two functions λ_j and χ_j from $\mathbb{R}$ to $\mathbb{R}$ such that

$$\lambda_j(\alpha) = f_j(x^k + \alpha w^k) \text{ and}$$
$$\chi_j(\alpha) = f_j(x^k) + b_1 \alpha \mathcal{M}(x^k, w^k) \quad \text{for all } j = 1, 2, \dots, r.$$

As $\lambda_j(0) = \chi_j(0)$ for all $j = 1, 2, \dots, r$. Given that F is bounded below in $\mathcal{T}$, thus $\lambda_j(\alpha)$ is bounded below for all $\alpha > 0$. By the definition of $\mathcal{M}$, we have $\mathcal{M}(x^k, w^k) < 0$, because w^k is a descent direction for F at x^k. Given that $b_1 \in (0, 1)$, the function $\chi_j(\alpha)$ is unbounded below and must intersect the graph of λ_j at least once for a positive α (use Lemma 3.1, [116]). Let $\alpha' > 0$ represent the smallest value of α at which the intersection occurs, that is,

$$f_j(x^k + \alpha' w^k) = f_j(x^k) + b_1 \alpha' \mathcal{M}(x^k, w^k). \tag{4.14}$$

Thus, we have

$$f_j(x^k + \alpha w^k) \leq f_j(x^k) + b_1 \alpha \mathcal{M}(x^k, w^k) \text{ for all } \alpha \in (0, \alpha']. \tag{4.15}$$

Thus, by Lemmas 4.3 and 5.3, using the inequality (4.15), we get

$$f_j(x^k + \alpha w^k) \leq C_j^k + b_1 \alpha \mathcal{M}(x^k, w^k) \text{ for all } \alpha \in (0, \alpha'].$$

By the Mean Value Theorem, there exists $\alpha'' \in (0, \alpha')$ such that

$$f_j(x^k + \alpha' w^k) = f_j(x^k) + \langle \nabla f_j(x^k + \alpha'' w^k), \alpha' w^k \rangle. \tag{4.16}$$

As $\langle \nabla f_j(x^k), w^k \rangle \leq \mathcal{M}(x^k, w^k) < 0$ for all $j = 1, 2, \ldots, r$, thus using (4.16), we can write

$$f_j(x^k + \alpha' w^k) \leq f_j(x^k) + \alpha' \mathcal{M}(x^k + \alpha'' w^k, w^k). \tag{4.17}$$

By combining (4.14) and (4.17), we get

$$\mathcal{M}(x^k + \alpha'' w^k, w^k) \geq b_1 \mathcal{M}(x^k, w^k) \geq b_2 \mathcal{M}(x^k, w^k),$$

since $b_2 > b_1$ and $\mathcal{M}(x^k, w^k) < 0$.

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

$$F(x^k + \alpha w^k) \preceq C^k + b_1 \alpha \mathcal{M}(x^k, w^k) e \text{ and } \mathcal{M}(x^k + \alpha w^k, w^k) \geq b_2 \mathcal{M}(x^k, w^k).$$

$\square$

The following theorem demonstrates that both algorithms are well-defined.

Theorem 4.1 *Assume that F is bounded below in $\mathcal{T}$. Then, Algorithm 5 and Algorithm 6 are well-defined.*

Proof: We prove by method of induction. Let us assume that $\mathscr{B}_j^k$ is positive definite for all $j = 1, 2, \ldots, r$ (which is trivially true for $k = 0$). Consequently, the subproblem presented in Step 9 of Algorithm 5 can be solved (see Section 3 of [129]). If x^k is Pareto critical, as indicated by Lemma 4.1(i), Algorithm 5 will terminate as mentioned in Step 9. Therefore, let us consider the scenario where x^k is not Pareto critical. In such a case, according to Lemma 4.1(ii), w^k serves as a descent direction of F at x^k. Given that F is bounded below in $\mathcal{T}$, there exist intervals of positive step sizes that fulfill conditions (4.8) and (4.9) (see Lemma 4.4). Consequently, x^{k+1} is well-defined. Now we will show that $\mathscr{B}_j^{k+1}$ in (4.12) remains positive definite for all $j = 1, 2, \ldots, r$. If $\mathscr{B}^k$ is positive definite and $l_k^\top y^k > 0$, it has been demonstrated in [116] that $\mathscr{B}_j^{k+1}$ will also be positive definite. The fulfillment of the latter requirement can be achieved when f_j exhibits strong convexity. In other words, for all x and y in $\mathbb{R}^n$, the following condition holds:

$$(\nabla f_j(x) - \nabla f_j(y))^\top (x - y) \geq \hat{b} \|x - y\|_2^2,$$

for each $j = 1, 2, \ldots, r$ and for some $\hat{b} > 0$. Towards this goal, we have assumed our objective function is strongly convex. Hence, Algorithm 5 and 6 are well-defined. □

4.5 Global Convergence

Throughout this section, we demonstrate the global convergence of Algorithm 5 and Algorithm 6. First, we prove some auxiliary results for global convergence. In the following lemma, we show a property about the sequence $\{C^k\}$, which is produced by the average-type nonmonotone line search and max-type nonmonotone line search procedures.

Lemma 4.5 (See [111, Lemma 7]). Let $\sigma \in (0, 1)$ and $\{x^k\}$ be a sequence generated by Algorithm 5 and Algorithm 6. Then, for each $j = 1, 2, \ldots, r$, $\{C_j^k\}$ is a nonincreasing sequence and admits a limit when $k \to \infty$.

We begin by estimating a lower bound for the step size α_k, generated by Algorithms 5 and 6.

Lemma 4.6 *Let x^k and α_k satisfy the nonmonotone Wolfe line search conditions* (4.8) *and* (4.9). *Let JF satisfy the following Lipschitz condition with the Lipschitz constant $L > 0$*

$$\|JF(x^{k+1}) - JF(x^k)\|_2 \leq L\|x^{k+1} - x^k\|_2. \tag{4.18}$$

Then,

$$\alpha_k \geq \left(\frac{1-b_2}{L}\right)\frac{|\mathcal{M}(x^k, w^k)|}{\|w^k\|_2^2}. \tag{4.19}$$

Proof: As α_k satisfies the Wolfe conditions, by (4.9), we have

$$\mathcal{M}(x^k + \alpha_k w^k, w^k) - \mathcal{M}(x^k, w^k) \geq (b_2 - 1)\mathcal{M}(x^k, w^k). \tag{4.20}$$

Since $\mathcal{M}(x^k, w^k) < 0$ and $b_2 \in (b_1, 1)$, we have $(b_2 - 1)\mathcal{M}(x^k, w^k) \geq 0$.
As JF is L-Lipschitz, thus by (4.20), (4.18) and Proposition 1.1(iii), we can write

$$\alpha_k L\|w^k\|_2^2 \geq (b_2 - 1)\mathcal{M}(x^k, w^k),$$

which gives

$$\alpha_k \geq \left(\frac{b_2 - 1}{L}\right)\frac{\mathcal{M}(x^k, w^k)}{\|w^k\|_2^2}. \tag{4.21}$$

Using (4.21) and the fact that $\mathcal{M}(x^k, w^k) < 0$, we can write

$$\alpha_k \geq \left(\frac{1-b_2}{L}\right)\frac{|\mathcal{M}(x^k, w^k)|}{\|w^k\|_2^2}.$$

This proves the desired inequality. □

Next, we demonstrate the global convergence of the sequence $\{x^k\}$ produced by Algorithm 5. In particular, we prove that every limit point of $\{x^k\}$ is Pareto critical.

Theorem 4.2 *Let x^k and α_k satisfy the nonmonotone Wolfe line search conditions (4.8) and (4.9). Assume that f_j for each $j \in \{1, 2, \dots, r\}$ are bounded below functions, $\sigma < 1$ and there exists c_1 such that*

$$|\mathcal{M}(x^k, w^k)| \geq c_1 \|w^k\|_2^2 \quad \textit{for all} \quad k = 0, 1, 2, \dots. \tag{4.22}$$

Moreover, let the assumption of Lemma 4.6 holds. Then, for any sequence $\{(x^k, w^k)\}$ obtained by Algorithm 5, if $\hat{x}$ is a limit point of $\{x^k\}$, then $\hat{x}$ is Pareto critical of F.

Proof: We first show that

$$F(x^{k+1}) \preceq C^k - \eta |\mathcal{M}(x^k, w^k)| e, \tag{4.23}$$

for all $j \in \{1, 2, \dots, r\}$, where $\eta = \frac{(1-b_2)c_1}{L}$. By (4.19), we have

$$\alpha_k \geq \left(\frac{1 - b_2}{L} \right) \frac{|\mathcal{M}(x^k, w^k)|}{\|w^k\|_2^2}.$$

Thus, by (4.8) and using the above inequality, we can write

$$F(x^{k+1}) \preceq C^k - \left(\frac{1 - b_2}{L} \right) \left(\frac{|\mathcal{M}(x^k, w^k)|}{\|w^k\|_2} \right)^2 e. \tag{4.24}$$

Now, by (4.22) and (4.24), we have

$$F(x^{k+1}) \preceq C^k - \left(\frac{(1 - b_2)c_1}{L} \right) |\mathcal{M}(x^k, w^k)| e.$$

which implies (4.23). Combining the expression of update of C^k given in (4.10) with (4.23), we obtain

$$C^{k+1} = \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} F(x^{k+1}) \preceq \frac{\sigma m_k}{m_{k+1}} C^k + \frac{C^k - \eta |\mathcal{M}(x^k, w^k)| e}{m_{k+1}} = C^k - \frac{\eta |\mathcal{M}(x^k, w^k)| e}{m_{k+1}}. \tag{4.25}$$

Since $f_j,\ j=1,2,\ldots,r$, is bounded from below, therefore by Lemma 5.3, we have $F(x^k) \preceq C^k$, for all $k=0,1,2,\ldots$. We conclude that $\{C_j^k\}$ is bounded from below. It follows from (4.25) that

$$\sum_{k=0}^{+\infty} \frac{|\mathcal{M}(x^k,w^k)|}{m_{k+1}} < \infty. \tag{4.26}$$

From Lemma 4.1(ii) using (4.26), we have

$$\sum_{k=0}^{+\infty} \frac{|\Psi(x^k)|}{m_{k+1}} < \sum_{k=0}^{+\infty} \frac{|\mathcal{M}(x^k,w^k)|}{m_{k+1}} < \infty.$$

Let $\hat{x}$ be a limit point of the sequence $\{x^k\}$. Given this, assume that the subsequence $\{x^k\}$, where $k \in K \underset{\infty}{\subset} \mathbb{N}$, converges to $\hat{x}$. We prove $\Psi(\hat{x})=0$ using by the method of contradiction. Assume that $\Psi(\hat{x})<0$. This assumption implies the existence of two positive values $\xi>0$ and $\gamma_0>0$ such that for any $0<\gamma\leq\gamma_0$ and for any $k\in K$ satisfying $\|x^k-\hat{x}\|\leq\gamma$, the following condition holds:

$$\Big|\Psi(x^k)\Big| \geq \xi > 0.$$

This means that

$$\sum_{k=0}^{+\infty} \frac{\Psi(x^k)}{m_{k+1}} \geq \sum_{k\in\{k\in K:\|x^k-\hat{x}\|\leq\gamma\}} \frac{\xi}{m_{k+1}}. \tag{4.27}$$

From (4.13) and $\sigma<1$, we have

$$m_{k+1} = 1 + \sum_{i=0}^{k}\prod_{p=0}^{i}\sigma_{k-p} \leq 1 + \sum_{i=0}^{k}\sigma^{i+1} \leq \sum_{i=0}^{\infty}\sigma^i = \frac{1}{1-\sigma}. \tag{4.28}$$

Based on (4.27) and (4.28) , we obtain

$$\sum_{k=0}^{+\infty} \frac{\Psi(x^k)}{m_{k+1}} \geq \sum_{k\in\{k\in K:\|x^k-\hat{x}\|\leq\gamma\}} \frac{\xi}{m_{k+1}} \geq \sum_{k\in\{k\in K:\|x^k-\hat{x}\|\leq\gamma\}} (1-\sigma)\xi = +\infty.$$

This contradicts (4.26). As a result, we can conclude that $\Psi(\hat{x})=0$ and according to

Lemma 4.1(i) $\hat{x}$ is a critical point for F. □

Subsequently, we demonstrate the global convergence of the sequence $\{x^k\}$ obtained by Algorithm 6. In particular, we prove that every limit point of $\{x^k\}$ is Pareto critical.

Theorem 4.3 *Let x^k and α_k satisfy the nonmonotone Wolfe line search conditions* (4.8) *and* (4.9)*. Assume that f_j for each $j \in \{1, 2, \ldots, r\}$ are bounded below functions, and there exists c_2 such that*

$$|\mathcal{M}(x^k, w^k)| \geq c_2 \|w^k\|_2^2, \quad \textit{for all} \quad k = 0, 1, 2, 3, \ldots. \tag{4.29}$$

Moreover, let the assumption of Lemma 4.6 hold. Then, for a sequence $\{(x^k, w^k)\}$ obtained by Algorithm 6, if $\hat{x}$ is a limit point of $\{x^k\}$, then $\hat{x}$ is Pareto critical of F.

Proof: We first show that

$$F(x^{k+1}) \preceq C^k - \delta |\mathcal{M}(x^k, w^k)| e, \tag{4.30}$$

for all $j \in \{1, 2, \ldots, r\}$, where

$$\delta = \frac{(1 - b_2) c_2}{L}. \tag{4.31}$$

By (4.19), we have

$$\alpha_k \geq \left(\frac{1 - b_2}{L} \right) \frac{|\mathcal{M}(x^k, w^k)|}{\|w^k\|_2^2}.$$

Thus by (4.8) and using the above inequality, we can write

$$F(x^{k+1}) \preceq C^k - \left(\frac{1 - b_2}{L} \right) \left(\frac{|\mathcal{M}(x^k, w^k)|}{\|w^k\|_2} \right)^2 e. \tag{4.32}$$

Now, by (4.29) and (4.32), we have

$$F(x^{k+1}) \preceq C^k - \left(\frac{(1 - b_2) c_2}{L} \right) |\mathcal{M}(x^k, w^k)| e,$$

which implies (4.30). For an index $j = 1, 2, \ldots, r$ and an iteration k, let $n_j(k)$ be an integer such that

$$k - a(k) \leq n_j(k) \leq k, \text{ and } C_j^k = f_j(x^{n_j(k)}).$$

Fix an index j. From (4.29) for $k > A$, we obtain

$$\begin{aligned} C_j^k &= f_j(x^{n_j(k)}) \\ &\leq C_j^{n_j(k)-1} - \delta|\mathcal{M}(x^{n_j(k)-1}, w^{n_j(k)-1})| \\ &\leq C_j^{n_j(k)-1} - \delta|\Psi(x^{n_j(k)-1})| \text{ using Lemma 4.1(ii).} \end{aligned}$$

From Lemma 5.4, we have $\lim_{k\to\infty}|C_j^{n_j(k)-1} - C_j^k| = 0$. Thus, since $\delta > 0$ and $|\Psi(x^{n_j(k)-1})| > 0$, the above inequality gives $\lim_{k\to\infty} |\Psi(x^{n_j(k)-1})| = |\Psi(\hat{x})| = 0$. From Lemma 4.1(i), $\hat{x}$ is a critical point for the function F.

□

4.6 Numerical Experiments

For numerical comparison, we have considered the following methods:

1. We employ the nonmonotone line search Algorithm 5.

2. We employ the nonmonotone line search Algorithm 6.

3. We utilize the monotone line search technique by setting $\sigma = 0$ in the nonmonotone line search of Algorithm 5. We regard this algorithm as Algorithm 3.

In our implementation, we made the selection of the step length α_k in a way that fulfills conditions (4.8), and (4.9) with parameters $\Gamma_1 = 0.3, \Gamma_2 = 0.5, b_1 = 0.4$ and $b_2 = 0.7$. For the monotone line search, we replace C^k in the equation (4.8) with $F(x^k)$. For Algorithm 5, we found that $\sigma = 0.20$ yields satisfactory results for a wide range of problems.

We implemented our algorithms using MATLAB R2018b and conducted tests on various known test problems from the literature. The experiments were carried out on a Windows 11 machine equipped with a 2.4 GHz CPU and 16 GB RAM. To determine the stopping criterion, we employed the condition $\Psi(x^k) \leq \epsilon$, with $\epsilon = 10^{-4}$. To handle the box constraints, denoted as $L \leqq x \leqq U$, for the original multiobjective problem, we approached it by considering problem (4.4) as

$$\begin{aligned} \min_{w \in \mathbb{R}^n} \quad & \max_{j=1,2,\ldots,r} \langle \nabla f_j(x), w \rangle + \tfrac{1}{2} w^\top \mathscr{B}_j w \\ \text{subject to} \quad & L - x \leqq w \leqq U - x. \end{aligned}$$

The test problems considered in this study were obtained from various literature sources, as outlined in Table 4.3. The first column of the table displays the names of the test problems. The second column gives the source of the test problems. The third column "m," represents the number of objective functions associated with each test problem. The fourth column "n," provides information on the dimension of the decision variable. The next two columns "$L^\top$" and "$U^\top$" depict the lower and upper bound of the decision variables, respectively. Some problems are designed in a manner that allows the dimension n of the variable space to be arbitrarily chosen. This flexibility also applies to the lower and upper bounds, L and U, respectively. The column labeled "Method" indicates the implementation of considered algorithms over test problems. The last two columns n_f and n_{it} record the average number of function evaluations and iterations, respectively. For each problem, we performed 100 iterations with distinct starting points. The starting points were uniformly selected from the range between L and U.

It is noteworthy that among all the problems, only JOS1, AP1, AP4, and VFM1 exhibit strong convexity. Although our theoretical results primarily focus on strongly convex problems, we also conducted tests on some nonconvex problems. In order to address nonconvex problems, we have made a modification to the BFGS matrix, denoted as $\mathscr{B}_j(x^k)$, within the algorithm. Specifically, we add a positive scalar $\mu_j + 1$ times the identity matrix, denoted as $(\mu_j + 1)I$, to $\mathscr{B}_j(x^k)$ to ensure positive definiteness when $\mathscr{B}_j(x^k)$ itself is not positive definite. Here, μ_j is the minimum eigenvalue of the matrix $\mathscr{B}_j(x^k)$.

Let us now delve into the numerical results presented in Table 4.3. By conducting a comparative analysis between the performance of nonmonotone algorithms (Algorithm 5 and 6) and monotone algorithm (Algorithm 3), it becomes evident that the nonmonotone approach exhibits superior performance. Specifically, across nearly all problem instances, the nonmonotone methods (Algorithm 5 and 6) consistently demonstrate lower average numbers of function evaluations (n_f) and iterations (n_{it}) compared to the monotone method (Algorithm 3).

To measure the performance of the considered algorithms, we conduct a relative comparison between Algorithm 5 and Algorithm 6 against Algorithm 3. For each i^{th} test problem and j^{th} solver we calculate the following ratio

$$r(i,j) = \frac{n_f(i,j)}{n_f(i, \text{Algorithm 3})}. \tag{4.33}$$

The geometric mean of these ratios for j^{th} solver across all the test problems are computed as follows:

$$R(j) = \left(\prod_{i \in P} r(i,j)\right)^{\frac{1}{|P|}}. \tag{4.34}$$

which is also known as *relative efficiency* [152], where the set of the test problems under consideration is denoted as P, and the cardinality of P is denoted by $|P|$. It is crucial to highlight that through the equation (4.33), we have computed the relative efficiency with respect to the average number of function evaluations (n_f). However, it is also possible to calculate the relative efficiency for the average number of iterations (n_{it}).

Algorithm 5	Algorithm 6	Algorithm 3
0.7138	0.6673	1

Table 4.1: Relative efficiency (see (4.34)) of Algorithms 3, 5 and 6 and with respect to n_f

Algorithm 5	Algorithm 6	Algorithm 3
0.0976	0.0603	1

Table 4.2: Relative efficiency (see (4.34)) of Algorithms 3, 5 and 6 with respect to n_{it}

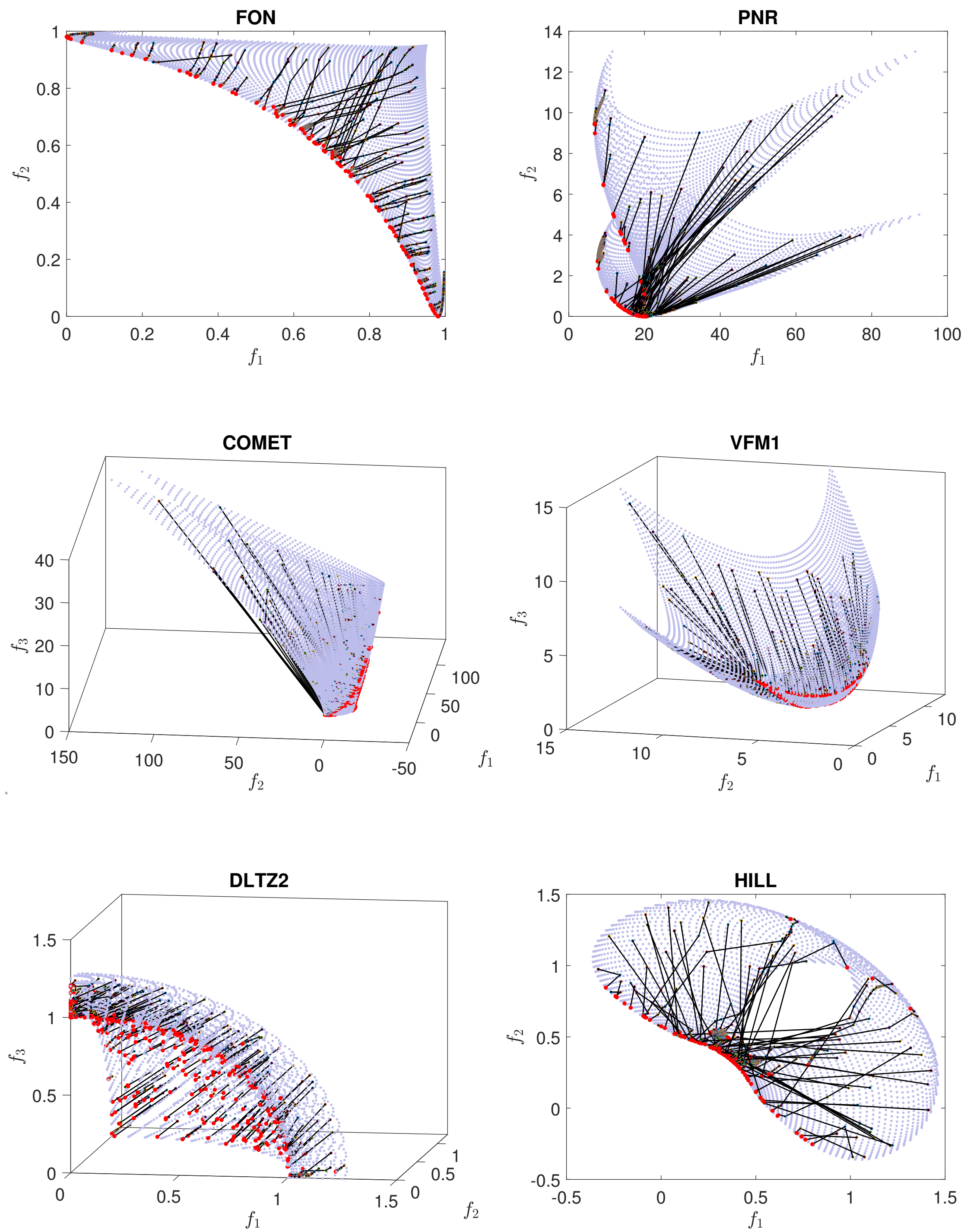

Figure 4.1: Objective feasible region and Pareto front approximation obtained by Algorithm 5 of the FON, PNR, VFM1, COMET, DLTZ2, HILL, MOP3 and JOS1. For each of the randomly taken 100 initial points, the iterated obtained by Algorithm 5 moves through a black line and finally reaches a red point

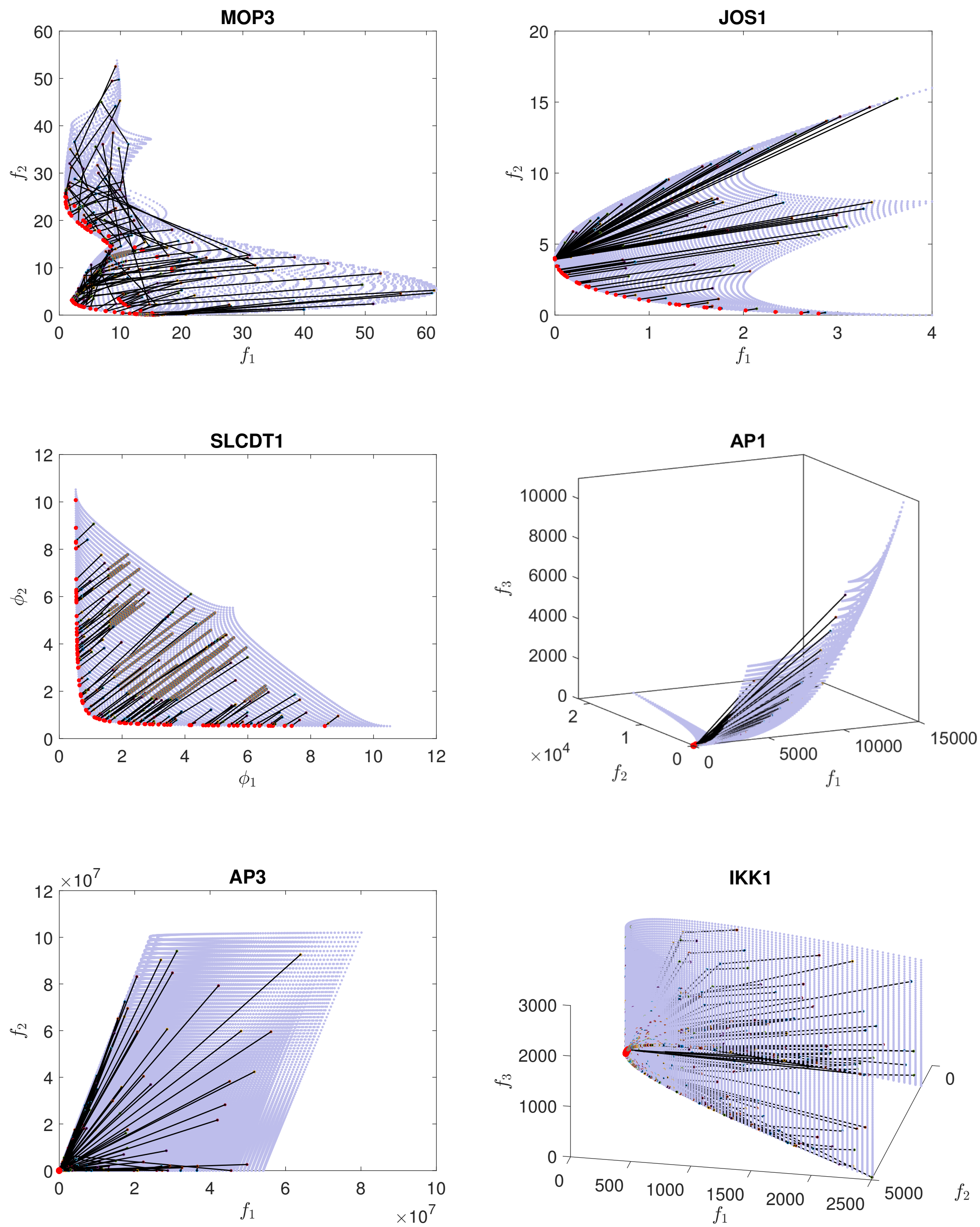

Figure 4.1: continued from Figure 4.1

The comparison between the considered methods is relative and not influenced by a small number of problems that may require a significant amount of function evaluations and gradient evaluations. Referring to the rule stated in (4.34), it is apparent from Tables 4.1 and 4.2 that $R(\text{Algorithm 3}) = 1$. Also, the lesser R, the better the performance. Tables 4.1 and 4.2 display values of $R(\text{Algorithm 5})$ and $R(\text{Algorithm 6})$. Tables 4.1 and 4.2 indicate that Algorithm 6 produces the best average performances with respect to both n_f and n_{it}. Additionally, Algorithm 5 produces better average performances than Algorithm 3.

Among the two proposed methods, we have considered Algorithm 5 for the plotting of the Pareto front approximations of the test problems. Figure 4.1 displays the output of the test results obtained by the proposed method. To check the ability to generate a Pareto front by the proposed method, we have considered multiobjective test problems of both strongly convex and nonconvex types.

In these graphs, the red points represent the final iterates obtained by Algorithm 5, while the initial point of a straight segment indicates the corresponding starting point. The blue points represent a discrete approximation of the criterion feasible region. Figure 4.1 demonstrates that by using an appropriate number of starting points and for the selected set of test problems, Algorithm 5 successfully produces an adequate portrayal of the Pareto front.

On the basis of test results reported in Table 4.1, we show the performance profile given by Dolan and Moré [37] to compare the performance of the considered methods. For detailed explanation of the performance profile given by Dolan and Moré [37], see Chapter 2.

Figures 4.2a and 4.2b illustrate the performance profile of three algorithms, namely Algorithm 5, Algorithm 6, and Algorithm 3. The profile is based on the average number of function evaluations (n_f) and iterations (n_{it}). The performance profile offers insight into the proportion ($\mathcal{P}(\tau)$) of test problems where each algorithm performs within a factor of τ of the best. According to Figure 4.2a, both Algorithm 5 and Algorithm 6 solve 65.22% of the test problems with the fewest average function evaluations, while

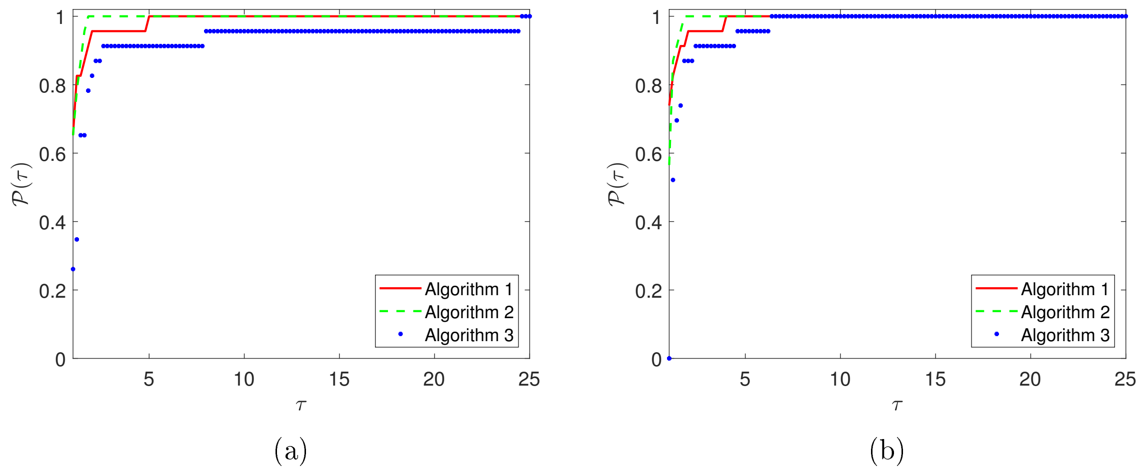

Figure 4.2: Performance profile of Algorithm 3, Algorithm 5 and Algorithm 6 measured by (a) n_f (b) n_{it}

Algorithm 3 only solves 26.09% of the test problems in this category. Moreover, Figure 4.2b depicts that Algorithm 5 solves 73.91% of the test problems with the least number of average iterations, whereas Algorithm 6 and Algorithm 3 solve 56.52% and 52.17% of the test problems in this category, respectively.

Name	Source	m	n	L^T	U^T	Method	n_f	n_{it}
AP1	[6]	3	2	$-[10, 10]$	$[10, 10]$	Algorithm 5	17.25	12.30
						Algorithm 6	20.73	14.53
						Algorithm 3	23.53	15.70
AP1	[6]	3	2	$-[50, 50]$	$[50, 50]$	Algorithm 5	31.61	22.31
						Algorithm 6	39.70	25.68
						Algorithm 3	42.14	27.18
AP3	[6]	2	2	$-[100, 100]$	$[100, 100]$	Algorithm 5	62.36	45.39
						Algorithm 6	58.83	44.95
						Algorithm 3	71.70	50.11
AP3	[6]	2	2	$-[500, 500]$	$[500, 500]$	Algorithm 5	94.25	66.32
						Algorithm 6	97.88	69.67
						Algorithm 3	105.61	76.23
AP4	[6]	3	3	$-[10, 10, 10]$	$[10, 10, 10]$	Algorithm 5	8.94	10.52
						Algorithm 6	5.17	10.52
						Algorithm 3	10.43	11.73
AP4	[6]	3	3	$-[50, 50, 50]$	$[50, 50, 50]$	Algorithm 5	30.57	23.96
						Algorithm 6	29.60	25.16
						Algorithm 3	31.20	26.52
MOP3	[75]	2	2	$-[\pi, \pi]$	$[\pi, \pi]$	Algorithm 5	12.91	7.84
						Algorithm 6	21.55	12.73
						Algorithm 3	25.50	13.77
COMET	[85]	3	2	$[1, -2]$	$[3.5, 2]$	Algorithm 5	4.08	2.09
						Algorithm 6	3.70	1.74
						Algorithm 3	4.58	2.15
JOS1	[75]	2	3	$-[2, 2, 2]$	$[2, 2, 2]$	Algorithm 5	1	1.90
						Algorithm 6	1	1.90
						Algorithm 3	1	1.91
JOS1	[75]	2	100	$-[2, \dots, 2]$	$[2, \dots, 2]$	Algorithm 5	1	1.90
						Algorithm 6	1	1.90
						Algorithm 3	1	1.91
JOS1	[75]	2	200	$-[2, \dots, 2]$	$[2, \dots, 2]$	Algorithm 5	1	1.90
						Algorithm 6	1	1.90
						Algorithm 3	1	1.91
JOS1	[75]	2	1000	$-[2, \dots, 2]$	$[2, \dots, 2]$	Algorithm 5	1	1.90
						Algorithm 6	1	1.90
						Algorithm 3	1	1.91
JOS1	[75]	2	100	$-[50, \dots, 50]$	$[50, \dots, 50]$	Algorithm 5	1	1.90
						Algorithm 6	1	1.90
						Algorithm 3	1	1.91
JOS1	[75]	2	100	$-[100, \dots, 100]$	$[100, \dots, 100]$	Algorithm 5	1	1.90
						Algorithm 6	1	1.90
						Algorithm 3	1	1.91
KW2	[87]	2	2	$-[5, 5]$	$[5, 5]$	Algorithm 5	11.66	8.74
						Algorithm 6	18.45	12.12
						Algorithm 3	20.24	12.93
PNR	[126]	2	2	$-[2, 2]$	$[2, 2]$	Algorithm 5	204.77	47.55
						Algorithm 6	191.44	44.96
						Algorithm 3	252.21	58.01
SLCDT1	[63]	2	2	$-[5, 5]$	$[5, 5]$	Algorithm 5	97.39	24.15
						Algorithm 6	101.34	25.74
						Algorithm 3	118.64	28.50
VFM1	[75]	2	2	$-[2, 2]$	$[2, 2]$	Algorithm 5	1.93	1.90
						Algorithm 6	1.93	1.91
						Algorithm 3	47.32	8.37
HILL	[74]	2	2	$[0, 0]$	$[1, 1]$	Algorithm 5	42.64	22.51
						Algorithm 6	8.88	5.79
						Algorithm 3	70.22	35.96
Viennet	[75]	3	3	$-[2, 2]$	$[2, 2]$	Algorithm 5	77.93	17.78
						Algorithm 6	53.79	12.23
						Algorithm 3	93.74	21.29
DTLZ2	[164]	3	3	$[0, 0, 0]$	$[1, 1, 1]$	Algorithm 5	78.93	18.23
						Algorithm 6	113.88	26.10
						Algorithm 3	135.19	30.73
FON	[164]	2	2	$[-1, -1]$	$[1, 1]$	Algorithm 5	30.06	8.93
						Algorithm 6	31.02	9.09
						Algorithm 3	39.41	10.63
IKK1	[76]	2	2	$-[50, 50]$	$[50, 50]$	Algorithm 5	62.35	32.25
						Algorithm 6	32.08	17.45
						Algorithm 3	80.21	40.96

Table 4.3: Performance of Algorithm 5, Algorithm 6 and Algorithm 3 on the set of commonly used test problems.

4.7 Conclusion

We employed two nonmonotone line search strategies to effectively tackle strongly convex unconstrained MOPs. The main innovation of this approach lies in the utilization of average-type Wolfe-like and max-type Wolfe-like line search rules, deviating from the traditional monotone line search strategies. Theoretical analysis, specifically Theorem 4.2 and Theorem 4.3, demonstrates that under suitable conditions, each limit point derived from the sequence of iterates produced by Algorithm 5 and Algorithm 6, respectively represents a Pareto critical point of the MOP (4.1).

To evaluate the efficacy of the proposed method compared to alternative approaches, a comparative analysis with Algorithm 5, Algorithm 6, and Algorithm 3 is performed. The efficiency of the proposed method is evaluated through empirical analysis, which includes the computation of relative efficiency (see Tables 4.1 and 4.2) and the generation of performance profiles (see Figures 4.2a and 4.2b) using the methodology developed by Dolan and Moré [37]. Our experimental study revealed that the nonmonotone Wolfe-type quasi-Newton algorithms required fewer function evaluations and iterations compared to the monotone quasi-Newton algorithm, further emphasizing their efficiency.

As future research, we can also focus on the implementation of average-type and max-type nonmonotone quasi-Newton methods for set optimization problems. These methods can provide valuable insights and solutions to complex optimization scenarios involving sets. Implementing these techniques in the context of set optimization problems could enhance the efficiency and effectiveness of optimization algorithms, opening up new avenues for solving challenging real-world problems.

****