Chapter 3

A Projection-Type Hybrid Conjugate Gradient Method for MOPs with Application to an SIR Model

3.1 Introduction

The conjugate gradient method, introduced by Fletcher and Reeves [47], addresses the unconstrained optimization problem:

$$\text{minimize} \quad f(x), \quad x \in \mathbb{R}^n, \tag{3.1}$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is a continuously differentiable function. The algorithm begins with an initial point $x^0 \in \mathbb{R}^n$ and generates a sequence of iterates:

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

Here, α_k represents the step size, determined through line search techniques. The search direction d^k is updated as follows:

$$d^0 = -\nabla f(x^0), \quad d^k = -\nabla f(x^k) + \beta_k d^{k-1}, \quad k = 1, 2, 3, \ldots, \tag{3.2}$$

where β_k is a parameter that varies depending on the specific conjugate gradient method employed. For strictly convex quadratic functions with Hessian Q, the parameter β_k is chosen to ensure Q-conjugacy of the directions:

$$(d^k)^\top Q d^{k+1} = 0, \quad k = 0, 1, 2, \ldots.$$

In this scenario, the step size is determined by exact line search:

$$\alpha_k = \text{argmin}_{\alpha>0} f(x^k + \alpha d^k).$$

This approach, known as the *linear conjugate gradient algorithm*, finds the minimizer of f over $\mathbb{R}^n$ within n iterations.

For non-quadratic functions, various conjugate gradient methods exist, differentiated by their choice of β_k. Notable variants include Hestenes-Stiefel (HS) [73], Polak-Ribiére-Polyak (PRP) [122], Liu-Storey (LS) [96], Hager-Zhang (HZ) [68], and Dai-Yuan (DY) [29]. These methods are classified on the basis of different expressions of β_k. Namely, in HS and DY methods, the algorithmic parameters β_k are defined as follows:

$$\text{HS: } \beta_k = \frac{\langle s^k, y^k \rangle}{\langle d^{k-1}, y^k \rangle} \text{ and}$$
$$\text{DY: } \beta_k = \frac{\langle s^k, s^k \rangle}{\langle d^{k-1}, y^k \rangle},$$

where $s^k = \nabla f(x^k)$, $y^k = s^k - s^{k-1}$, and $\langle \cdot, \cdot \rangle$ denotes the inner product. These

formulations ensure the *sufficient descent* condition:

$$\langle \nabla f(x^k), d^k \rangle \leq -c\|\nabla f(x^k)\|_2^2, \quad k = 0, 1, 2, \ldots,$$

for some $c > 0$, where $\|\cdot\|_2$ is the Euclidean norm.

In practice, inexact line search conditions are often used in the context of conjugate gradient methods , such as the *Standard Wolfe conditions*:

$$f(x^k + \alpha_k d^k) \leq f(x^k) + \rho\alpha_k \langle \nabla f(x^k), d^k \rangle \quad \text{and} \quad \langle \nabla f(x^k + \alpha_k d^k), d^k \rangle \geq \sigma \langle \nabla f(x^k), d^k \rangle,$$

or the *strong Wolfe conditions*:

$$f(x^k + \alpha_k d^k) \leq f(x^k) + \rho\alpha_k \langle \nabla f(x^k), d^k \rangle \quad \text{and} \quad \left| \langle \nabla f(x^k + \alpha_k d^k), d^k \rangle \right| \leq \sigma \left| \langle \nabla f(x^k), d^k \rangle \right|,$$

where $0 < \rho < \sigma < 1$.

A method is considered *globally convergent* if:

$$\liminf_{k \to \infty} \|\nabla f(x^k)\|_2 = 0, \text{ for any initial point } x^0.$$

To establish the convergence of the conjugate gradient methods, it is necessary that the methods follow the following *Zoutendjik condition* (see [98]):

$$\sum_{k=0}^{\infty} \frac{\langle \nabla f(x^k), d^k \rangle^2}{\left\| d^k \right\|_2^2} < +\infty.$$

In conjugate gradient methods, Zoutendjik and Wolfe conditions are frequently implemented to establish the global convergence result of the problem (3.1).

3.2 Motivation

According to [4], various conjugate gradient methods have distinct advantages and disadvantages. Methods such as CD, FR, and DY show strong convergence but often suffer from performance degradation due to jamming, while PRP, LS, and HS methods offer better computational efficiency but weaker convergence. To address this, hybrid conjugate gradient methods have been introduced, combining the strengths of existing algorithms through projection techniques or convex combinations (see Table 6.1 in [4]). These hybrids are generally more effective and robust than standard methods.

Lucambio Pérez and Prudente [98] expanded several conjugate gradient techniques (including FR, CD, DY, PRP, and HS) for unconstrained vector optimization, while Gonçalves and Prudente [63] adapted the HZ conjugate gradient method for similar challenges, and Gonçalves et al. [64] extended LS conjugate gradient methods to multiobjective scenarios. As noted in [90, 109, 137], conjugate gradient methods offer advantages over steepest descent and Newton's methods, such as not requiring Hessian matrices and faster convergence. The study in [1] shows that conjugate gradient methods generally outperform steepest descent.

Building upon these developments, we propose two novel approaches to further enhance the performance of conjugate gradient methods for MOPs:

- A conjugate gradient method for MOP is proposed by combining the HS and DY methods, with a focus on studying its global convergence. Given the robustness and efficiency of hybrid methods [4], we extend the HSDY hybrid conjugate gradient method to unconstrained MOPs.

- A nonmonotone hybrid conjugate gradient method combining PRP with HS and DY is studied. This approach builds upon our previously proposed nonmonotone PRP-type conjugate gradient method (see Chpater 2), aiming to explore the method's capabilities when combined with other conjugate gradient techniques.

This hybrid approach leverages the strengths of PRP, HS, and DY methods to potentially achieve improved performance in solving complex optimization problems. Additionally, we and apply it to a nonlinear SIR (Susceptible, Infected, Recovered) epidemiological optimal control model with vaccination and treatment as controls, aiming to reduce the number of infected individuals and the implementation cost of control strategies.

These proposed methods demonstrate the ongoing evolution of conjugate gradient techniques, particularly in addressing multiobjective optimization challenges. By combining established methods in novel ways and incorporating nonmonotone strategies, we aim to develop more robust and efficient optimization tools applicable to a wide range of real-world problems, including epidemiological modeling and control.

3.3 Contributions

The following points summarize the contribution of the present chapter:

- We introduce a novel projection-based hybrid conjugate gradient method that combines the HS and DY approaches for tackling MOPs. Additionally, we introduce two novel hybrid conjugate gradient approaches that combines the PRP method with HS and DY conjugate gradient method, implemented under a average-type nonmonotone line search strategy.

- Our method incorporates the strong Wolfe line search technique to determine appropriate step sizes.

- For the PRP hybrid method, we demonstrate how the nonmonotone line search strategy enhances the algorithm's flexibility and efficiency in handling nonconvex MOPs. The global convergence of both proposed schemes is studied without any convexity assumption. We provide a thorough analysis of the global convergence properties of our proposed methods, without relying on convexity assumptions.

- To validate our theoretical findings, we conduct extensive numerical experiments on both convex and non-convex test problems, showcasing the effectiveness of our method in various scenarios.

- We assess the performance of our monotone and nonmonotone hybrid approaches by comparing it to the individual HS and DY conjugate gradient methods, using relative efficiency metrics.

- For a comprehensive evaluation, we employ Dolan and Moré performance profiles to compare the HS, DY, and our hybrid HSDY, nonmonotone HSPRP and DYPRP methods across a diverse set of test problems.

- To demonstrate real-world applicability, we apply our proposed method to a multiobjective optimal control problem based on the SIR epidemiological model.

In this chapter, we consider the following unconstrained MOP:

$$\text{minimize } F(x) = (f_1(x), f_2(x), \ldots, f_m(x)), \quad x \in \mathbb{R}^n, \tag{3.3}$$

where $F = (f_1, f_2, \ldots, f_m) : \mathbb{R}^n \to \mathbb{R}^m$, and each $f_i : \mathbb{R}^n \to \mathbb{R}$, $i \in \{1, 2, \ldots, m\}$, is continuously differentiable.

To perform the convergence analysis of conjugate gradient method, we require the direction d^k to be a descent direction of F at x^k, for all $k = 0, 1, 2, \ldots$. In most of the convergence analysis, we require

$$\mathcal{M}(x^k, d^k) \leq c\mathcal{M}\big(x^k, d_{SD}(x^k)\big) \tag{3.4}$$

for some $c \in (0, 1]$ and for all $k = 0, 1, 2, \ldots$. We refer (3.4) as a *sufficient descent condition.*

The following is an auxiliary result from [63] which will be used to prove Lemma 3.5.

Lemma 3.1 [63, Lemma 3] *For any unit vectors $x, \hat{x} \in \mathbb{R}^n$ and $\gamma \geq 0$, we have*

$$\|x - \hat{x}\|_2 \leq 2\|x - \gamma\hat{x}\|_2 .$$

Now, we explore various conjugate gradient methods and their global convergence properties that have been proposed in the literature to address MOPs.

3.4 HSDY Algorithm and Its Global Convergence

Recently, Lucambio Pérez and Prudente [98] introduced the general conjugate gradient method for solving (3.3) which is defined by the following recurrence relation:

$$x^{k+1} = x^k + \alpha_k d^k, \quad \text{for } k \geq 0, \tag{3.5}$$

where the step size α_k is obtained by a line search method, and the search direction d^k is given by

$$d^0 = d_{SD}(x^0) \quad \text{and} \quad d^{k+1} = d_{SD}(x^{k+1}) + \beta_{k+1} d^k, \quad k \geq 0,$$

where β_k is the scalar-valued algorithmic parameter and $d_{SD}(x^{k+1})$ is given in (1.7). As given in [98], the vector extensions of β^{HS} and β^{DY} are given by

$$\beta_k^{\mathrm{HS}} = \frac{-\eta_1^k + \eta_2^k}{\eta_3^k - \eta_4^k}, \quad \text{and} \quad \beta_k^{\mathrm{DY}} = \frac{-\eta_1^k}{\eta_3^k - \eta_4^k},$$

where $\eta_1^k = \mathcal{M}(x^{k+1}, d_{SD}(x^{k+1}))$, $\eta_2^k = \mathcal{M}(x^k, d_{SD}(x^{k+1}))$, $\eta_3^k = \mathcal{M}(x^{k+1}, d^k)$ and $\eta_4^k = \mathcal{M}(x^k, d^k)$. Therefore, the vector extension of hybrid scalar-valued algorithmic parameter β_k^{HSDY} is given by

$$\beta_k^{\mathrm{HSDY}} = \max\left\{0, \min\{\beta_k^{\mathrm{HS}}, \beta_k^{\mathrm{DY}}\}\right\} = \max\left\{0, \min\left\{\frac{-\eta_1^k + \eta_2^k}{\eta_3^k - \eta_4^k}, \frac{-\eta_1^k}{\eta_3^k - \eta_4^k}\right\}\right\}, \tag{3.6}$$

which reduces to the classical one in the scalar minimization case (see [4, 98]). For the list of all hybrid scalar-valued algorithmic parameters for scalar minimization (see Table 6.1 of [4]).

Next, we propose an HSDY Algorithm 2 for multiobjective optimization in which the direction d^k satisfies the sufficient descent condition. It is a known fact that a suitable choice of line search guarantees descent in the context of conjugate gradient methods; for example, see [62, 98, 140].

For a given multiobjective function F and $\bar{n}$ (number of randomly chosen initial points), Algorithm 2 starts with a randomly chosen starting point x^0. If $d_{SD}(x^0) = 0$, then we stop the iteration process and declare x^0 as a Pareto critical point. Otherwise, we set the initial direction $d^0 = d_{SD}(x^0)$. To determine the next iterate x^1, the step size α is obtained by the following condition:

$$\begin{cases} F(x^k + \alpha d^k) \preceq F(x^k) + \rho\alpha\mathcal{M}(x^k, d^k)e \text{ and} \\ \left|\mathcal{M}(x^k + \alpha d^k, d^k)\right| \leq \sigma\left|\mathcal{M}(x^k, d^k)\right| \end{cases} \tag{3.7}$$

Then, we check if x^1 is Pareto critical. Otherwise, we calculate the hybrid algorithmic parameter β_0^{HSDY} from equation (3.6) to obtain a new direction d^1, and update k by $k+1$ to repeat the procedure until we get an x^k, which is Pareto critical. To generate a discrete approximation of the complete Pareto critical set of problem (3.3), repeat the above steps for $\bar{n}$ randomly chosen starting points. The method guarantees that (x^k, d^k) satisfies (3.4) for every $k \geq 0$, which we prove later. Under the assumptions (A1) and (A2), which are stated later, the existence of intervals of positive step sizes satisfying (3.4) is ensured by Proposition 3.2 in [98]. Thus, a line search method can be coded based on the work done in [99]; see the numerical experiment section for more details.

Algorithm 2 HSDY Algorithm to Solve the MOP (3.3)

Aim: To generate a discrete approximation of the complete Pareto critical set of problem (3.3)

Provide $F = \big(f_1, f_2, \ldots, f_m\big)$, where each f_i is continuously differentiable
Choose ρ and σ arbitrarily such that $0 < \rho < \sigma < 1$
Provide $\bar{n}$, the number of randomly chosen initial points
Provide the tolerance level $\epsilon > 0$ for the optimum solution of the problem (3.3)
Set Pareto set $\mathcal{S} \leftarrow \emptyset$
for $n = 1 : \bar{n}$ **do**
    Choose a random point $x^0 \in \mathbb{R}^n$
    Compute $d_{SD}(x^0)$ and $\Theta_{SD}(x^0)$ using (1.7) and (1.8), respectively
    Set $d^0 \leftarrow d_{SD}(x^0)$, $C^0 \leftarrow F(x^0)$, $m_0 \leftarrow 1$, $k \leftarrow 0$
    while $|\Theta_{SD}(x^k)| > \epsilon$ **do**
        Set $\alpha \leftarrow 1$
        while $F(x^k + \alpha d^k) \succ F(x^k) + \rho\alpha\mathcal{M}(x^k, d^k)e$ or $\big|\mathcal{M}(x^k + \alpha d^k, d^k)\big| > \sigma\big|\mathcal{M}(x^k, d^k)\big|$ **do**
            Set $\alpha \leftarrow \alpha q$
        end while
        Set $x^{k+1} \leftarrow x^k + \alpha d^k$
        Compute $\beta_{k+1}^{\text{HSDY}}$ using (3.6)
        Compute $d_{SD}(x^{k+1})$ and $\Theta_{SD}(x^{k+1})$ using (1.7) and (1.8), respectively
        Define $d^{k+1} = d_{SD}^{k+1} + \beta_{k+1}^{\text{HSDY}} d^k$
        Set $k \leftarrow k + 1$
    end while
    return $\bar{x} = x^k$ as a Pareto critical point of (3.3)
    Update set $\mathcal{S} \leftarrow \mathcal{S} \cup \{F(\bar{x})\}$
end for
return $\mathcal{S}$ as a discrete approximation of the complete Pareto set of problem (3.3)

Before showing the well-definedness of Algorithm 2, we show that under the condition of strong Wolfe line search, the direction d^k, obtained by Algorithm 2, satisfies the sufficient descent condition (3.4).

Theorem 3.1 *Consider Algorithm 2 and suppose that α_k satisfies the strong Wolfe condition* (1.10). *Then, d^k satisfies the sufficient descent condition* (3.4) *with $c = \frac{1}{1+\sigma}$ for any $k \in \mathbb{N}$.*

Proof: We use the method of induction to prove. For $k = 0$, the inequality (3.4) holds

trivially with $c = \frac{1}{1+\sigma}$. For some $k \geq 1$, assume that

$$\mathcal{M}(x^{k-1}, d^{k-1}) \leq \frac{1}{1+\sigma}\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1})) < 0. \tag{3.8}$$

By the definition of d^k, we have

$$\langle JF(x^k)d^k, w\rangle = \langle JF(x^k)d_{SD}(x^k), w\rangle + \beta_k^{HSDY}\langle JF(x^k)d^{k-1}, w\rangle, \quad \text{for all } w \in C.$$

First suppose that $\mathcal{M}(x^k, d^{k-1}) \leq 0$. As $\beta_k^{HSDY} \geq 0$, we obtain

$$\langle JF(x^k)d^k, w\rangle \leq \langle JF(x^k)v(x^k), w\rangle \leq \mathcal{M}(x^k, d_{SD}(x^k)) \leq \frac{1}{1+\sigma}\mathcal{M}(x^k, d_{SD}(x^k)),$$

for all $w \in C$.

Now, assume that $\mathcal{M}(x^k, d^{k-1}) > 0$. By the definition of $\mathcal{M}$, we have

$$\mathcal{M}(x^k, d^k) \leq \mathcal{M}(x^k, d_{SD}(x^k)) + \beta_k^{\mathrm{HSDY}}\mathcal{M}(x^k, d^{k-1}), \quad \text{for all } w \in C.$$

For $\beta_k^{HSDY} = 0$, the inequality (3.4) clearly holds.

Let $\beta_k^{HSDY} > 0$. Then, from the definition of β_k^{HSDY}, we have

$$\beta_k^{HSDY} = \min\{\beta_k^{HS}, \beta_k^{DY}\}.$$

Thus, $\beta_k^{HSDY} \leq \beta_k^{DY}$. Therefore,

$$\mathcal{M}(x^k, d^k) \leq \mathcal{M}(x^k, d_{SD}(x^k)) + \beta_k^{\mathrm{HSDY}}\mathcal{M}(x^k, d^{k-1}) \leq \mathcal{M}(x^k, d_{SD}(x^k)) + \beta_k^{\mathrm{DY}}\mathcal{M}(x^k, d^{k-1}). \tag{3.9}$$

Define $t^k = \mathcal{M}(x^k, d^{k-1})/\mathcal{M}(x^{k-1}, d^{k-1})$. Then, from (3.8), t^k is well-defined.

By the second inequality of strong Wolfe condition, we have $t^k \in [-\sigma, \sigma]$. By the

inequality (3.9) and the definition of β_k^{DY}, we have

$$\mathcal{M}(x^k, d^k) \leq \frac{1}{1-t^k}\mathcal{M}(x^k, d_{SD}(x^k)) \leq \frac{1}{1+\sigma}\mathcal{M}(x^k, d_{SD}(x^k)),$$

which completes the proof. □

It is important to note that the above proof is inspired from the proof of Lemma 5.5 in [98] with the modification in the choice of parameter $\beta_{k+1}^{\text{HSDY}} = \max\left\{0, \min\left\{\beta_{k+1}^{\text{HS}}, \beta_{k+1}^{\text{DY}}\right\}\right\}$. To show the well-definedness of Algorithm 2 and to prove further results, we make the following assumptions.

(**A1**) The Jacobian JF is L-Lipschitz continuous on an open set $\mathcal{N}$ containing $\mathcal{L} = \left\{x \in \mathbb{R}^n : F(x) \preceq F(x^0)\right\}$, i.e., $\left\|JF(x) - JF(y)\right\|_2 \leq L\|x-y\|_2$ for every $x, y \in \mathcal{N}$.

(**A2**) For any $\{M_k\}$ in $F(\mathcal{L})$ such that $M_{k+1} \preceq M_k$ for all k (i.e., a monotonic nonincreasing sequence), there exists $\mathcal{M} \in \mathbb{R}^m$ satisfying $\mathcal{M} \preceq M_k$ for all k. In other words, all monotonic nonincreasing sequences in $F(\mathcal{L})$ are bounded below.

Under the assumptions (A1) and (A2), the well-definedness of Algorithm 2 depends upon the existence of step size α and the well definedness of the algorithmic parameter β_{k+1}^{HS} and β_{k+1}^{DY}. For a given F, K, x^k, and d^k, where d^k is descent direction for F at x^k, the existence of interval of positive step sizes satisfying (3.4) is ensured by Proposition 3.2 in [98]. The algorithmic paramater β_{k+1}^{HS} and β_{k+1}^{DY} are well-defined, i.e., $\eta_3^{k+1} - \eta_4^{k+1} \neq 0$, which can be easily observed by the second inequality of (1.9).

The following lemma states that under the assumptions (A1) and (A2) the method (3.5) follows Zoutendijk's condition. It can be noted that the above assumptions are just the vector extension of those made for scalar case.

Lemma 3.2 *Assume that the assumptions (A1) and (A2) hold. Consider an iteration of the form* (3.5)*, where* d^k *is a sufficient descent direction (according to* (3.4)*) for* F

at x^k and α_k satisfies the strong Wolfe conditions (1.10). *Then,*

$$\sum_{k\geq 0} \frac{(\mathcal{M}(x^k, d^k))^2}{\left\|d^k\right\|_2^2} < +\infty. \tag{3.10}$$

Proof: It follows the lines of the proof of Proposition 3.3 in [98]. □

Remark 3.1 *The Zoutendijk condition* (3.10) *is used to establish the convergence for several line search algorithms. In particular, for conjugate gradient methods, in the case of vector optimization, corresponding to a starting point x^0, we declare the convergence of the sequence of iterates $\{x^k\}$ when it satisfies the condition $\liminf_{k\to\infty}\left\|d_{SD}(x^k)\right\|_2 = 0$; For a detailed explanation, see* [63, 98].

In order to prove the convergence of Algorithm 2, we assume that the algorithm generates an infinite sequence of iterates $\{(x^k, d^k)\}$ that satisfies (3.4) for all $k = 0, 1, 2, \ldots$. With this assumption we prove that Algorithm 2 solves the problem (3.3) asymptotically (i.e., $\liminf_{k\to\infty}\left\|d_{SD}(x^k)\right\|_2 = 0$). To prove the next result, we take the following assumption which is stronger than the assumption (A2).

(**A3**) The level set $\mathcal{L} = \left\{x \in \mathbb{R}^n : F(x) \preceq_K F(x^0)\right\}$ is bounded.

Lemma 3.3 *Assume that the assumption (A3) holds. Suppose that Algorithm 2 generates a sequence $\{(x^k, d^k)\}$ satisfying the sufficient descent condition* (3.4) *for all $k = 0, 1, 2, \ldots$. Then, there exist a positive constant $\bar{\gamma}$ such that*

$$\left\|d_{SD}(x^k)\right\|_2 \leq \bar{\gamma}, \quad \textit{for all } k = 0, 1, 2, \ldots. \tag{3.11}$$

Proof: It follows from the first inequality of (3.7) and the assumption (A3) that $\{x^k\}$ is contained in the bounded set $\mathcal{L}$. Hence, the result follows by using the continuity of $d_{SD}(x)$ (see Lemma 1.2(iii)). □

In order to prove the global convergence of Algorithm 2, we must establish that $d_{SD}(x^k)$ cannot be bounded away from zero. In this regard, we propose the following contradiction hypothesis:

(**A4**) There exists a positive constant $\underline{\gamma}$ such that

$$\underline{\gamma} \leq \left\| d_{SD}(x^k) \right\|_2, \quad \text{for all } k = 0, 1, 2, \ldots. \tag{3.12}$$

Inspired by the proof of Theorem 5.11 in [98] with the modification in the choice of parameter $\beta_{k+1}^{\text{HSDY}} = \max\left\{0, \min\left\{\beta_{k+1}^{\text{HS}}, \beta_{k+1}^{\text{DY}}\right\}\right\}$, we establish an estimate for the algorithmic parameter β_k^{HSDY}.

Lemma 3.4 *Assume that the assumptions (A1)–(A4) hold and Algorithm 2 generates a sequence* $\{(x^k, d^k)\}$ *satisfying the sufficient descent condition* (3.4) *for all* $k = 0, 1, 2, \ldots$. *Then, for every* $k = 1, 2, 3, \ldots$, *we have*

$$\beta_k^{HSDY} \leq \frac{2L\overline{\gamma}}{c(1-\sigma)\underline{\gamma}^2} \left\| x^k - x^{k-1} \right\|_2,$$

where $\overline{\gamma}$, $\underline{\gamma}$ *and* L *are from* (3.11), (3.12) *and the assumption (A1), respectively.*

Proof: Notice from the definition of β_k^{HSDY} that

$$\beta_k^{HSDY} = \max\left\{0, \min\{\beta_k^{HS}, \beta_k^{DY}\}\right\} \leq \max\{0, \beta_k^{HS}\} \leq \left|\beta_k^{HS}\right|.$$

Therefore, we get

$$\beta_k^{HSDY} \leq \left|\beta_k^{HS}\right| = \frac{\left|-\eta_1^k + \eta_2^k\right|}{\left|\eta_3^k - \eta_4^k\right|}, \tag{3.13}$$

where

$\eta_1^k = \mathcal{M}(x^k, d_{SD}(x^k))$, $\eta_2^k = \mathcal{M}(x^{k-1}, d_{SD}(x^k))$, $\eta_3^k = \mathcal{M}(x^k, d^{k-1})$ and $\eta_4^k = \mathcal{M}(x^{k-1}, d^{k-1})$.

Now, we obtain an expression for upper bound for the right-hand side of (3.13). From the definitions of η_1^k and η_2^k, and Lemma 1.1(c) and Lemma 3.3, we have

$$\begin{aligned}\left|-\eta_1^k+\eta_2^k\right| &= \left|-\mathcal{M}(x^k, d_{SD}(x^k)) + \mathcal{M}(x^{k-1}, d_{SD}(x^k))\right| \\ &\leq L\left\|d_{SD}(x^k)\right\|_2 \left\|x^k - x^{k-1}\right\|_2 \\ &\leq L\overline{\gamma}\left\|x^k - x^{k-1}\right\|_2. \end{aligned} \tag{3.14}$$

Also, by the expression of η_3^k and η_4^k, from (3.4), second inequality of (3.7), Lemma 1.2(ii) and (3.12) we have

$$\begin{aligned}\eta_3^k - \eta_4^k = \mathcal{M}(x^k, d^{k-1}) - \mathcal{M}(x^{k-1}, d^{k-1}) &\geq -(1-\sigma)\mathcal{M}(x^{k-1}, d^{k-1}) \\ &\geq -c(1-\sigma)\mathcal{M}(x^{k-1}, d_{SD}(x^{k-1})) \\ &\geq \frac{c(1-\sigma)\left\|d_{SD}(x^{k-1})\right\|_2^2}{2} \\ &\geq \frac{c(1-\sigma)\underline{\gamma}^2}{2} \geq 0.\end{aligned}$$

We can also write

$$\left|\eta_3^k - \eta_4^k\right| \geq \frac{c(1-\sigma)\underline{\gamma}^2}{2}. \tag{3.15}$$

Thus, from (3.14) and (3.15), we get

$$\beta_k^{HSDY} \leq \frac{2L\overline{\gamma}\left\|x^k - x^{k-1}\right\|_2}{c(1-\sigma)\underline{\gamma}^2}.$$

□

Before showing the global convergence for Algorithm 2, an important necessary result needs to be established, which is as follows.

Lemma 3.5 *Assume that the assumptions (A1)–(A4) hold and Algorithm 2 generates a sequence* $\{(x^k, d^k)\}$ *satisfying the sufficient descent condition* (3.4) *for all* $k = 0, 1, 2, \ldots$.

Then, $d^k \neq 0$, *for every* $k = 0, 1, 2, \ldots$ *and*

$$\sum_{k=1}^{\infty} \frac{1}{\left\|d^k\right\|_2^2} < +\infty \quad \textit{and} \quad \sum_{k=1}^{\infty} \left\|m^k - m^{k-1}\right\|_2^2 < +\infty,$$

where $m^k = d^k / \left\|d^k\right\|_2$.

Proof: It follows in the lines of proof of Lemma 6 in [63]. □

Next, we state the global convergence of Algorithm 2 under the assumption that Algorithm 2 generates a sequence of iterates $\{(x^k, d^k)\}$ that satisfies (3.4) for all $k \geq 0$.

Theorem 3.2 *Assume that (A1)–(A3) hold. Suppose that Algorithm 2 generates a sequence* $\{(x^k, d^k)\}$ *satisfying the sufficient descent condition* (3.4) *for all* $k = 0, 1, 2, \ldots$. *Then,*

$$\liminf_{k \to \infty} \left\|d_{SD}(x^k)\right\|_2 = 0.$$

Proof: It follows in the lines of proof of Theorem 2 in [63]. □

3.5 Nonmonotone Hybrid Conjugate Gradient Methods

In this section, we introduce a novel projection-type hybrid conjugate gradient method that combines the PRP conjugate gradient method with the HS and DY conjugate gradient methods. This approach is implemented under an average-type nonmonotone line search strategy, aiming to harness the strengths of these methods while mitigating their individual weaknesses.

The primary motivation for this work is to investigate the performance of the PRP method when combined with HS and DY conjugate gradient methods under an average-type nonmonotone line search strategy. To achieve this, we propose two Nonmonotone methods based on the hybrid projection of PRP with HS and DY:

- Hestenes-Stiefel-Polak-Ribiére-Polyak (HSPRP) conjugate gradient method.
- Dai-Yuan-Polak-Ribiére-Polyak (DYPRP) conjugate gradient method.

3.5.1 Nonmonotone DYPRP and HSPRP Algorithm and Its Global Convergence

Numerous research studies have demonstrated that the utilization of nonmonotone schemes can significantly enhance the performance of descent methods, particularly in scalar cases (see [43] and the references therein). In this section, we introduce novel projection-type hybrid conjugate gradient methods that combine the PRP method with the HS and DY methods, implemented under an average-type nonmonotone line search strategy. In the conventional monotone line search technique, we choose α_k to ensure $F(x^{k+1}) \prec F(x^k)$, meaning each iteration results in a lower objective function value. However, in nonmonotone line search, a certain amount of growth is allowed in the objective function values. We choose $\alpha_k > 0$ that satisfies

$$F(x^k + \alpha_k(d^k)) \preceq C^k + \rho\alpha_k\mathcal{M}(x^k, d^k)e \tag{3.16}$$

with $\rho \in (0, 1)$ and $C^k \succeq F(x^k)$.

Grippo et al. [66] introduced a nonmonotone line search framework based on taking the maximum of recent objective function values. Qu et al. [130] proposed two nonmonotone gradient methods formulated on the max-type nonmonotone line search, extending work by Fliege and Svaiter [49] and Cruz et al. [24]. In our current investigation, we employ the average-type approach proposed by Zhang and Hager [156], recognized for its efficiency, particularly in scalar cases. This method uses the average of objective function values rather than the maximum of recent values.

It is worth noting that we have previously detailed the strong Wolfe conditions (3.7). In our work, we have implemented an average-type nonmonotone version of the strong Wolfe line search (3.7) which is given by:

$$F(x^k + \alpha d^k) \preceq C^k + \rho\alpha\mathcal{M}(x^k, d^k)e \text{ and } \left|\mathcal{M}(x^k + \alpha d^k, d^k)\right| \leq \sigma\left|\mathcal{M}(x^k, d^k)\right| \tag{3.17}$$

where C^k is as given in (2.10). As given in [98], a vector extension of the PRP conjugate parameter is

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

Building upon this, we propose two projection-type hybrid conjugate gradient coefficients

$$\beta_k^{HSPRP} = \max\left\{0, \min\{\beta_k^{HS}, \beta_k^{PRP}\}\right\}, \quad \beta_k^{DYPRP} = \max\left\{0, \min\{\beta_k^{DY}, \beta_k^{PRP}\}\right\}. \quad (3.18)$$

These hybrid coefficients combine the strengths of PRP with HS and DY methods respectively, aiming to enhance the overall performance and robustness of the algorithm.

Our proposed nonmonotone projection-type hybrid conjugate gradient algorithm aims to generate a discrete approximation of the complete Pareto critical set for MOP. The algorithm incorporates the average-type nonmonotone line search strategy to ensure global convergence while allowing for more flexible exploration of the solution space.

The algorithm leverages the HSPRP and DYPRP coefficients to determine search directions, combining the efficiency of PRP with the stability of HS and DY methods. This hybrid approach, coupled with the average-type nonmonotone line search, allows for occasional increases in the objective function values, potentially leading to improved performance in navigating complex optimization landscapes. The detailed steps of the algorithm are as follows:

Algorithm 3 Average-type nonmonotone projection-type hybrid conjugate algorithm to solve the MOP (3.3)

Aim: To generate a discrete approximation of the complete Pareto critical set of problem (3.3)

Provide $F = \big(f_1, f_2, \ldots, f_m\big)$, where each f_i is continuously differentiable
Choose ρ and σ arbitrarily such that $0 < \rho < \sigma < 1$ and $0 < q < 1$
Provide $\bar{n}$, the number of randomly chosen initial points
Provide the tolerance level $\epsilon > 0$ for the optimum solution of the problem (3.3)
Set Pareto set $\mathcal{S} \leftarrow \emptyset$
for $n = 1 : \bar{n}$ **do**
    Choose a random point $x^0 \in \mathbb{R}^n$
    Set $C^0 \leftarrow F(x^0), m_0 \leftarrow 1, k \leftarrow 0$
    Compute $d_{SD}(x^0)$ and $\Theta_{SD}(x^0)$ using (1.7) and (1.8), respectively
    Set $d^0 = d_{SD}(x^0)$
    while $|\Theta_{SD}(x^k)| > \epsilon$ **do**
        $\alpha \leftarrow 1$
        while $F(x^k + \alpha d^k) \succ C^k + \rho\alpha\mathcal{M}(x^k, d^k)e$ or $\big|\mathcal{M}(x^k + \alpha d^k, d^k)\big| > \sigma\big|\mathcal{M}(x^k, d^k)\big|$ **do**
            $\alpha \leftarrow \alpha q$
        end while
        Set $x^{k+1} = x^k + \alpha d^k$
        Update m_{k+1} and C^{k+1} by:

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

        Compute $d_{SD}(x^{k+1})$ and $\Theta_{SD}(x^{k+1})$ using (1.7) and (1.8), respectively
        Define $d^{k+1} = d_{SD}^{k+1} + \beta_{k+1} d^k$, where β_{k+1} is either β_{k+1}^{HSPRP} or β_{k+1}^{DYPRP} as given in (3.18)
        Set $k \leftarrow k + 1$
    end while
    return $\bar{x} = x^k$ as a Pareto critical point (3.3)
    Update set $\mathcal{S} \leftarrow \mathcal{S} \bigcup \{F(\bar{x})\}$
end for
return $\mathcal{S}$ as a discrete approximation of the complete Pareto set of the problem (3.3)

The algorithm begins by providing the multiobjective function F composed of continuously differentiable functions. It selects parameters ρ and σ for the line search criteria and defines the number of initial points and the tolerance level for the solution. The main loop iterates over each initial point to generate a candidate Pareto critical point. Within this loop, the search direction and step size are updated iteratively using

a average-type nonmonotone line search to ensure the decrease of the objective function values. The set of Pareto critical points is updated and returned as the final output, providing a discrete approximation of the complete Pareto set.

3.5.2 Convergence Analysis

The following lemma provides important insights into how the sequence of function values and the approximation C^k behave relative to the average of past function values. It is crucial for understanding the convergence properties and ensuring that the iterates generated by the algorithm remain within appropriate bounds.

Lemma 3.6 [147] *Assume that for $k = 0, 1, 2, \ldots$, $x^k \in \mathbb{R}^n$ be a non Pareto critical point of (3.3) generated by Algorithm 3. For each iteration of Algorithm 3, we have $F(x^k) \preceq C^k \preceq S^k$, where $S^k = \frac{1}{k+1}\sum_{j=0}^{k} F(x^j)$.*

To ensure the well-definedness of the algorithm, we need to verify that certain conditions hold during the line search process. Specifically, we must demonstrate that there exists a suitable step size that satisfies the required descent conditions. This is crucial for the algorithm to proceed correctly and to guarantee convergence. We present the following lemma to establish these conditions.

Lemma 3.7 *Assume that d^k is a descent direction for F at x^k. Let $\rho \in (0,1)$, $\sigma \in (0,1)$, and $e = (1, 1, \ldots, 1)^\top \in \mathbb{R}^m$. Then, $\exists\ \bar{\alpha} \in (0,1]$ such that, $\forall\ \alpha \in (0, \bar{\alpha}]$,*

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

Proof: Since d^k is a descent direction of F at x^k, we have $\mathcal{M}(x^k, d^k) < 0$. Moreover, $\mathcal{M}(x^k, d^k) < \rho\mathcal{M}(x^k, d^k)$ because $\rho \in (0,1)$. Since F is differentiable, we obtain

$$f_i(x^k + \alpha d^k) = f_i(x^k) + \langle \nabla f_i(x^k), \alpha d^k \rangle + o(\alpha) \quad \forall\ i = 1, 2, \ldots, m. \tag{3.19}$$

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 $i = 1, 2, 3, \ldots$, there exists $\alpha' \in (0, 1)$ such that for all $i = 1, 2, \ldots, m$, (3.19) 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 $i = 1, 2, \ldots, m$ from Lemma 3.6, 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, define $\eta(\alpha) = \mathcal{M}(x^k + \alpha d^k, d^k)$. Since d^k is a descent direction of F, $\eta(0) = \mathcal{M}(x^k, d^k) < 0$. Due to the continuity of $\mathcal{M}$ (see Lemma 1.1(ii)) and the fact that $\mathcal{M}(x^k, d^k) < 0$, there exists $\alpha'' \in (0, 1)$ such that

$$|\mathcal{M}(x^k + \alpha d^k, d^k)| \leq \sigma |\mathcal{M}(x^k, d^k)| \quad \forall\ \alpha \in (0, \alpha''].$$

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

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

□

To establish the convergence properties of the algorithm, we first present the following lemma.

Lemma 3.8 [147] *Let $\sigma \in (0, 1)$ and $\{x^k\}$ be a sequence obtained by Algorithm 3. Then, for every $i = 1, 2, \ldots, m$, $\{C_i^k\}$ is a nonincreasing sequence and convergent.*

We now demonstrate that the sequence generated by Algorithm 3 converges to a Pareto critical point. Establishing this result is crucial as it guarantees that the algorithm is effective in solving MOPs.

Theorem 3.3 *Let $\{(x^k, d^k)\}$ be a sequence obtained by the Algorithm 3 using an average-type nonmonotone strong Wolfe line search. If $\hat{x}$ is a subsequential limit of $\{x^k\}$, then $\hat{x}$ is a Pareto critical point of the vector-valued objective function $F : \mathbb{R}^n \to \mathbb{R}^m$.*

Proof: Let $K = \{k_0, k_1, k_2, \ldots\} \underset{\infty}{\subset} \mathbb{N}$ be a sequence of indices such that $\lim_{j\to\infty} x^{k_j} = \hat{x}$. Given that $k_{j+1} \geq k_j + 1$. Taking into account the insights provided by Lemma 3.6 and 3.8 and by (2.10), we obtain, for all $j \in \mathbb{N}$,

$$f_i(x^{k_{j+1}}) \leq C_i^{k_{j+1}} \leq C_i^{k_j+1} \leq C_i^{k_j} + \frac{\sigma\alpha_{k_j}}{m_{k_j+1}}\mathcal{M}(x^{k_j}, d^{k_j}) \leq C_i^{k_j} \quad \forall\, i \in \mathcal{I}, \tag{3.20}$$

By the continuity assumption on the objective function, we have $\lim_{j\to\infty} F(x^{k_j}) = F(\hat{x})$. This observation implies that $\{F(x^{k_j})\}$ is bounded, which in turn leads us to conclude, based on (3.20), that $\{C_i^{k_j}\}$ is a bounded monotone sequence for each $i \in \mathcal{I}$. Consequently, each $\{C_i^{k_j}\}$ sequence possesses a limit. From (3.20), we have

$$C_i^{k_{j+1}} \leq C_i^{k_j} + \frac{\sigma\alpha_{k_j}}{m_{k_j+1}}\mathcal{M}(x^{k_j}, d^{k_j}) \quad \forall\, i \in \mathcal{I}. \tag{3.21}$$

By (3.21), we can write

$$0 \leq \lim_{j\to\infty} \frac{-\sigma\alpha_{k_j}}{m_{k_j+1}}\mathcal{M}(x^{k_j}, d^{k_j}) \leq \lim_{j\to\infty} C_i^{k_j} - C_i^{k_{j+1}} = 0. \tag{3.22}$$

Hence,

$$\lim_{j\to\infty} \left(\frac{\alpha_{k_j}}{m_{k_j+1}}\right)\mathcal{M}(x^{k_j}, d^{k_j}) = 0. \tag{3.23}$$

We claim that

$$\lim_{j\to\infty} \alpha_{k_j}\mathcal{M}(x^{k_j}, d^{k_j}) = 0. \tag{3.24}$$

If $\rho = 0$, then $m_{k_j+1} = 1$ for all $j \in \mathbb{N}$, and (3.24) holds as a trivial case. However,

assuming that $\rho \in (0,1)$, based on (2.10), we can observe that

$$m_{k_j+1} = 1 + \sum_{\ell=0}^{k_j} \rho^{\ell+1} \leq \sum_{\ell=0}^{\infty} \rho^{\ell} = \frac{1}{1-\rho}. \tag{3.25}$$

Consequently, $\{\frac{1}{m_{k_j+1}}\}$ is bounded from below.

By (3.22) and (3.25), we get

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

Thus (3.24) follows.

Now, by (3.24), there exists a $K' \underset{\infty}{\subset} K$ such that at least one the two cases holds:

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

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

Case 1: By (3.4) and Lemma 1.2(ii), we have that

$$\mathcal{M}(x^k, d^k) \leq c\mathcal{M}(x^k, d_{SD}(x^k)) \leq -\frac{c}{2}\|d_{SD}(x^k)\|_2^2 \leq 0. \tag{3.26}$$

Thus, applying limits in (3.26) 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: In this case, there exists a subsequence $K' \subset K$ such that $\lim_{k\in K'} \alpha_k = 0$. Following the same argument as in the Case 2 of Theorem 2.1, there must exist a $\bar{\alpha}_k = \alpha_k/v$ (for some $v > 1$) that violates one of the Wolfe conditions.

Case 2.1 If the first Wolfe condition is violated, then

$$F(x^k + \bar{\alpha}_k d^k) \not\preceq C^k + \rho\bar{\alpha}_k \mathcal{M}(x^k, d^k)e.$$

This means there exists at least one index $i \in \{1, 2, \ldots, m\}$ such that

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

Using the Mean Value Theorem, we can write

$$f_i(x^k + \bar{\alpha}_k d^k) - f_i(x^k) = \bar{\alpha}_k \nabla f_i(x^k + \gamma_k \bar{\alpha}_k d^k)^T d^k$$

for some $\gamma_k \in (0, 1)$. Combining the above inequalities, we get

$$\nabla f_i(x^k + \gamma_k \bar{\alpha}_k d^k)^T d^k > \rho \mathcal{M}(x^k, d^k).$$

Taking the limit as $k \to \infty$, $k \in K'$, and using the continuity of ∇f_i, we obtain

$$\nabla f_i(\hat{x})^T \hat{d} \geq \rho \mathcal{M}(\hat{x}, \hat{d}),$$

where $\hat{d}$ is the limit of d^k. Since this holds for all i, and by definition $\mathcal{M}(\hat{x}, \hat{d}) \geq \nabla f_i(\hat{x})^T \hat{d}$ for all i, as $\rho \in (0, 1)$ and $\mathcal{M} < 0$ for all $k = 0, 1, 2, \ldots$, we get $\mathcal{M}(\hat{x}, \hat{d}) = 0$. By (3.4) and Lemma 1.2(ii), we have

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

Thus, applying limits 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 2.2 Let the second condition of (3.17) is violated.

In this case, for infinitely many $k \in K'$, we have

$$|\mathcal{M}(x^k + \bar{\alpha}_k d^k, d^k)| > \sigma |\mathcal{M}(x^k, d^k)|. \tag{3.27}$$

By the Mean Value Theorem, there exists $\theta_k \in (0,1)$ such that

$$\mathcal{M}(x^k + \bar{\alpha}_k d^k, d^k) = \mathcal{M}(x^k, d^k) + \bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k.$$

Substituting this into the violated condition (3.27), we get

$$|\mathcal{M}(x^k, d^k) + \bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k| > \sigma |\mathcal{M}(x^k, d^k)|.$$

This inequality can be rewritten as either:

$$\mathcal{M}(x^k, d^k) + \bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k > \sigma \mathcal{M}(x^k, d^k)$$

$$\text{or } \mathcal{M}(x^k, d^k) + \bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k < \sigma \mathcal{M}(x^k, d^k).$$

In the first case, we have

$$\bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k > (\sigma - 1)\mathcal{M}(x^k, d^k),$$

and in the second case, we have

$$\bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k < (\sigma - 1)\mathcal{M}(x^k, d^k).$$

Now, recall that $\mathcal{M}(x^k, d^k) < 0$ for all k. Therefore, in both cases, we can conclude

that

$$|\bar{\alpha}_k \nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k| > (1-\sigma)|\mathcal{M}(x^k, d^k)|. \tag{3.28}$$

Dividing both sides of (3.28) by $\bar{\alpha}_k$ (which is positive), we get

$$|\nabla \mathcal{M}(x^k + \theta_k \bar{\alpha}_k d^k, d^k)^T d^k| > \frac{(1-\sigma)|\mathcal{M}(x^k, d^k)|}{\bar{\alpha}_k}. \tag{3.29}$$

As $k \to \infty$ (for $k \in K'$), we know that $\bar{\alpha}_k \to 0$. If $\lim_{k \in K'} |\mathcal{M}(x^k, d^k)| > 0$, then the right-hand side of the inequality (3.29) would approach infinity, which is impossible since the left-hand side is bounded (due to the boundedness of $\{d^k\}$ and the continuity of $\nabla \mathcal{M}$).

Therefore, we must have $\lim_{k \in K'} \mathcal{M}(x^k, d^k) = 0$. By the continuity of $\mathcal{M}$, this implies $\mathcal{M}(\hat{x}, \hat{d}) = 0$, where $\hat{d}$ is any limit point of the sequence $\{d^k\}_{k \in K'}$.

Now, using the same argument as in Case 1, we can conclude that $d_{SD}(\hat{x}) = 0$, and thus, by Lemma 1.2(i), $\hat{x}$ is a Pareto critical point of F.

Therefore, any subsequential limit point $\hat{x}$ of the sequence $\{x^k\}$ generated by the algorithm with average-type nonmonotone strong Wolfe line search is a Pareto critical point of F. □

3.6 Numerical Experiments

This section presents a comprehensive evaluation of the proposed method's effectiveness across a diverse set of commonly used test problems. Our experimental setup is designed to provide a thorough and fair comparison with other established methods in the field.

To assess the efficiency of our proposed monotone and nonmonotone hybrid conjugate gradient method, we compare it with two other well-known conjugate gradient methods, all implemented with the strong Wolfe line search:

(i) **HSDY**: Our proposed Hestenes-Stiefel, Dai-Yuan hybrid conjugate gradient method for MOPs, as detailed in Algorithm 2.

(ii) **HSPRP**: A nonmonotone Hestenes-Stiefel and Polak-Ribière-Polyak conjugate gradient method for MOPs, as detailed in Algorithm 3 with the algorithmic paramter $\beta_k = \beta_k^{HSPRP}$ as given in (3.18).

(iii) **DYPRP**: A nonmonotone Dai-Yuan and Polak-Ribière-Polyak conjugate gradient method for MOPs, as detailed in Algorithm 3 with the algorithmic paramter $\beta_k = \beta_k^{DYPRP}$ as given in (3.18).

(iv) **HS+**: The Hestenes-Stiefel method [98], modified to use only positive values of the HS parameter. This is implemented as Algorithm 2 with the algorithmic parameter $\beta_k = \max\{0, \beta_k^{\text{HS}}\}$, where

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

(v) **DY**: The Dai-Yuan method [98], implemented as Algorithm 2 with the algorithmic parameter $\beta_k = \beta_k^{\text{DY}}$, where

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

It is worth noting that the convergence properties of the HS+, DY and PRP methods have been previously established in [98].

3.6.1 Implementation Details

For the step size calculation in Algorithm 2, we employ the strong Wolfe condition (3.7) and for Algorithm 3, we employ nonmonotone strong Wolfe condition (3.17) with the following parameter values:

- $\rho = 10^{-4}$
- $c = 0.9$
- $\sigma = 0.1$
- $q = 0.4$

The termination criterion for our iterations is set as $\|\Phi(x)\| \leq \epsilon$, where $\epsilon = 10^{-4}$. All computations were performed using MATLAB software (version R2018b).

3.6.2 Performance Metrics

Table 3.1 provides a comprehensive list of the test problems used in our experiments, along with their key characteristics. This includes the problem name, source, number of decision variables (n), number of objectives (m), convexity, and the range for initial points (x^0).

For each test problem, we conducted 300 runs using different starting points randomly generated from a uniform distribution within the specified box constraints. For problems with original box constraints, we ensured that the starting points were within these bounds but ignored the constraints during optimization.

Our performance evaluation is based on several key metrics, as presented in Table 3.2:

- **per**: The percentage of runs that successfully reached a critical point.
- **it**: The average number of iterations for successful runs.
- **evalf**: The average number of function evaluations for successful runs.
- **evalg**: The average number of gradient evaluations for successful runs.

3.6.3 Results and Discussion

The results in Table 3.2 demonstrate the robustness of all considered methods, with most runs successfully reaching a critical point across all test problems. The few exceptions where critical points were not attained were due to computational errors in evaluating the steepest descent direction $v(x)$.

A notable observation from the 'it' values in Table 3.2 is the rapid convergence of all methods to critical points, highlighting the practical advantages of conjugate gradient methods for MOPs.

To visualize the effectiveness of our proposed HSDY method in generating Pareto fronts, we focused on nine biobjective optimization problems: PNR, FON, DGO1, HILL, LE1, MOP3, KW2, FAR1, and SLCDT1. For each problem, we plotted the criteria feasible region and the value-space acquired by the HSDY method using 150 randomly generated starting points.

In Figures 3.1 and 3.2, f_1 and f_2 representing the two objectives for each problem. The PNR problem is given special attention, showing the step-by-step Pareto point generation process for 5, 50, 100, 200, and 300 random starting points.

These visual results demonstrate that, given a sufficient number of starting points, the HSDY method effectively estimates the Pareto front for the considered problems. Notably, the method successfully identifies both global and local Pareto optimal points for challenging problems like MOP3, KW2, FAR1, and DGO1.

3.6.4 Performance Comparison

To quantitatively compare the HSDY method with the DY, HS+, nonmonotone HSPRP and DYPRP methods, we introduce a relative efficiency metric. For each test problem i and solver j, we calculate the ratio $r(i,j) = \frac{\text{it}(i,j)}{\text{it}(i,\text{HSDY})}$, where $\text{it}(i,j)$ is the total number of average iterations for problem i using solver j.

The overall relative efficiency of solver j is then computed as the geometric mean of

these ratios across all test problems:

$$r(j) = \left(\prod_{i \in P} r(i,j) \right)^{\frac{1}{|P|}}, \tag{3.30}$$

where P is the set of test problems and $|P|$ is its cardinality.

This comparison is relative and not dominated by a few problems where the method might require a large number of function and gradient evaluations. By definition, $r(\text{HSDY}) = 1$, and a lower value of r indicates better performance. The values of $r(\text{DY})$ $r(\text{HS+})$, $r(\text{HSPRP})$ and $r(\text{DYPRP})$ are presented in Table 3.3. These results demonstrate that the HSPRP method produces the best average performances.

To further visualize the performance of our proposed HSDY method, Figures 3.1 and 3.2 display the output of the test results obtained by Algorithm 2. We focused on nine biobjective optimization problems to assess the method's ability to generate Pareto optimal points. In these figures, blue points represent the feasible criterion space, obtained by discretizing the respective boxes $[lb^\top, ub^\top]$ through a fine grid. This provides an accurate representation of the image spaces of F and the geometric location of the Pareto front. The red points were generated by executing the proposed algorithm 150 times for each problem, using randomly obtained initial points from the respective box $[lb^\top, ub^\top]$.

These visual results demonstrate that, given a sufficient number of starting points, Algorithm 2 effectively estimates the Pareto front for the considered problems. The figures also show that for each test problem, a starting point reached a critical point in almost all iterations, highlighting the method's robustness. However, it is worth noting that failures may occur. For instance, in problems like HILL, FAR1, and MOP3, some runs terminated at local (nonglobal) Pareto optimal points due to errors in calculating the steepest descent direction $d_{SD}(x)$.

The table 3.2 provides a detailed performance comparison of five optimization meth-

ods (DY, HS+, HSDY, HSPRP, DYPRP) on various test problems. The performance metrics include per, it, evalf, and evalg. Here is a details of the observation on the performance:

- **For per**: All methods have consistently solved 100% of the problems in most cases. The only exceptions are in problems such as FAR1, VU1, AP1, and AP4 where the percentage slightly drops for some methods but not significantly. This indicates that all methods are generally reliable in solving the given problems.

- **For it**: The DY method tends to require a higher number of iterations for many problems (e.g., FON: 21.5, HILL: 41, SK2: 81.1, VU1: 1081.6). This suggests that DY may be less efficient in terms of convergence speed compared to other methods. The HS+ method performs better than DY, with lower iteration counts for most problems, though still somewhat high for problems like VU1 (228.6 iterations). HSDY shows improved performance with even lower iteration counts across most problems, indicating a more efficient convergence. HSPRP further improves efficiency, often requiring the fewest iterations among all methods (e.g., FON: 5.2, HILL: 11.2, VU1: 189.5). DYPRP also demonstrates high efficiency, comparable to HSPRP, with low iteration counts (e.g., FON: 5.5, HILL: 12.1).

- **For evalf**: DY requires a high number of function evaluations, especially for more complex problems (e.g., VU1: 4330.4, HILL: 168), indicating less efficiency. HS+ generally requires fewer function evaluations than DY, but still relatively high for some problems (e.g., VU1: 891.2). HSDY is more efficient, with a significant reduction in function evaluations (e.g., FON: 31.1, SK2: 189.3). HSPRP performs very efficiently, with even lower function evaluations (e.g., FON: 25.3, VU1: 762.2). DYPRP maintains similar performance to HSPRP, showing low function evaluations (e.g., FON: 26.8, VU1: 814.5).

- **For evalg**: Similar to function evaluations, DY has high gradient evaluations,

which is a sign of inefficiency (e.g., FON: 382.4, VU1: 19468). HS+ reduces the gradient evaluations but still has relatively high values for some problems (e.g., VU1: 4022). HSDY improves efficiency with lower gradient evaluations (e.g., HILL: 245.3, VU1: 3932.9). HSPRP is highly efficient, with the lowest gradient evaluations among the methods (e.g., FON: 98.7, VU1: 3412.6). DYPRP also performs efficiently, similar to HSPRP, with low gradient evaluations (e.g., FON: 102.4, VU1: 3648.5).

Overall, the DY method is the least efficient, requiring the highest number of iterations, function evaluations, and gradient evaluations. HS+ is more efficient than DY but still requires relatively high iterations and evaluations in some cases. HSDY shows improved efficiency with lower iterations and evaluations, making it a better choice. HSPRP is one of the most efficient methods, with consistently low iterations, function, and gradient evaluations. DYPRP performs similarly to HSPRP, making it one of the best choices for efficiency.

To provide a more comprehensive comparison, we employed the performance profile introduced by Dolan and Moré [37]. For detailed explanation of performance profile introduced by Dolan and Moré, see Chapter 2

Figures 3.3a–3.3c present these performance profiles, demonstrating that the nonmonotone HSPRP and DYPRP methods consistently outperformed all the other methods across all three metrics: it, evalf, and evalg.

This comprehensive analysis provides a clear picture of the nonmonotone hybrid conjugate method's superior performance relative to other established conjugate gradient methods, demonstrating its effectiveness and efficiency in solving MOPs.

Problem	n	m	Convex	x^0	Source
AP1	2	3	Y	$[-100, 100]^n$	[6]
AP3	2	2	N	$[-100, 100]^n$	[6]
AP4	3	3	Y	$[-100, 100]^n$	[6]
BK1	2	2	Y	$[-5, 10]^n$	[75]
DD1	5	2	N	$[-20, 20]^n$	[32]
Hill	2	2	N	$[0, 1]^n$	[74]
JOS1	1000	2	Y	$[-10^4, 10^4]^n$	[75]
KW2	2	2	N	$[-3, 3]^n$	[87]
LE1	2	2	N	$[-3, 3]^n$	[75]
FON	2	2	N	$[-1, 1]^n$	[164]
MOP1	1	2	Y	$[-10, 10]^n$	[75]
MOP3	2	2	N	$[-\pi, \pi]^n$	[75]
PNR	2	2	Y	$[-1, 1]^n$	[126]
SLCDT1	2	2	N	$[-5, 5]^n$	[98]
FAR1	2	2	N	$[-1, 1]^n$	[98]
FF1	2	2	N	$[-1, 1]^n$	[75]
SP1	2	2	Y	$[-10, 10]^n$	[75]
VU1	2	2	N	$[-3, 3]^n$	[75]
VU2	2	2	Y	$[-3, 3]^n$	[75]
SSFYY1	2	2	Y	$[-100, 100]^n$	[75]
SSFYY2	1	2	Y	$[-100, 100]$	[75]
LRS1	2	2	Y	$[-50, 50]^n$	[75]
QV1	50	2	N	$[-5.12, 5.12]^n$	[75]
DGO1	1	2	N	$[-10, 13]^n$	[75]
DGO2	1	2	Y	$[-9, 9]^n$	[75]
IM1	2	2	N	$[1, 4] \times [1, 2]$	[75]
SK2	4	2	N	$[-10, 10]^n$	[75]

Table 3.1: List of test problems

	DY				HS+				HSDY				HSPRP				DYPRP			
Problem	per	it	evalf	evalg	per	it	evalf	evalg	per	it	evalf	evalg	per	it	evalf	evalg	per	it	evalf	evalg
FON	100	21.5	90.2	382.4	100	15.5	48.2	201.3	100	6.8	31.1	116.3	100	5.2	25.3	98.7	100	5.5	26.8	102.4
HILL	100	41	168	733.5	100	31.8	101.2	497.9	100	13.9	59.7	245.3	100	11.2	48.6	201.8	100	12.1	52.3	215.6
DGO1	100	1	6.7	9.9	100	1	6.5	9.3	100	1	6.1	9	100	1	5.9	8.7	100	1	5.8	8.5
DGO2	100	1.8	11.3	27	100	1	8	18.6	100	1	6.9	15.3	100	1	6.5	14.2	100	1	6.3	13.8
LE1	100	15.7	105.3	421.7	100	13.8	101.4	417.6	100	15.1	120	419.6	100	12.3	98.5	340.2	100	12.8	102.1	352.8
SK2	100	81.1	491	894.2	100	67.8	362.1	812.9	100	45.4	189.3	559.9	100	37.2	155.4	458.1	100	39.8	166.2	489.7
PNR	100	3.2	16.6	51.8	100	3.1	16.6	51.3	100	3.1	16.5	50.8	100	2.8	14.9	45.7	100	2.9	15.3	47.2
MOP1	100	1	8	12	100	1	8	12	100	1	8	12	100	1	7.5	11.3	100	1	7.3	11
MOP3	100	10.4	45.7	182.1	100	6.8	24	88.4	100	3.4	13.1	54.9	100	2.9	11.2	46.7	100	3.1	12.0	50.1
KW2	100	14.4	61.6	257.6	100	13.9	58.6	241.7	100	12.1	52.3	215	100	10.5	45.4	186.8	100	11.2	48.5	199.7
FAR1	87	22.8	25.1	96.9	89.1	18.2	22.2	84.1	91	14.2	16	55.5	93	11.8	13.3	46.1	92	12.5	14.1	48.9
FF1	100	43.2	177	774.6	100	12.8	55.3	227.8	100	10.2	45	180.6	100	8.7	38.3	153.5	100	9.3	41.0	164.2
BK1	100	3.8	19.4	63.5	100	3.3	15.1	50.6	100	3	12.9	41.6	100	2.6	11.2	36.1	100	2.8	12.0	38.7
LRS1	100	4.8	22.9	79.4	100	5	27	88.3	100	4.1	21.4	74.6	100	3.6	18.7	65.1	100	3.8	19.9	69.4
VU1	85	1081.6	4330.4	19468	890	228.6	891.2	4022	91	218.6	878.4	3932.9	93	189.5	762.2	3412.6	92	202.8	814.5	3648.5
VU2	100	12	52.2	219.3	100	8.5	40.1	188.9	100	3.3	23.3	89.1	100	2.8	19.8	75.7	100	3.0	21.2	81.1
SP1	100	6.6	30.7	114.6	100	5.7	25	91.1	100	4.7	23	79.8	100	4.1	20.0	69.4	100	4.3	21.2	73.6
SSFYY1	100	5.1	24.7	83.9	100	5.3	28	90.1	100	4.3	23	80.4	100	3.7	19.9	69.5	100	4.0	21.3	74.4
SSFYY2	100	2.2	11.1	37.7	100	1.1	8.7	23.2	100	1	7.1	21.4	100	1	6.2	18.6	100	1	6.5	19.5
JOS1	100	2	12.1	30.6	100	1.7	11	29.8	100	1.7	11	28.3	100	1.5	9.6	24.6	100	1.6	10.2	26.1
AP1	92.8	5	34.1	99.8	97	4.2	31.3	90.7	97	4	30.5	86.3	98	3.5	26.5	75.1	97.5	3.7	28.2	79.8
AP3	100	8.2	58.2	91.1	100	4	21.7	63.4	100	3.2	17.1	60.4	100	2.8	14.9	52.5	100	3.0	15.9	56.2
AP4	90	3.7	24.4	67	91	3	23.4	63	91	2.8	20.3	54	92	2.4	17.6	46.8	91.5	2.6	18.8	50.1
SLCDT1	100	3	23	49.1	100	2.7	17.1	45	100	2.5	14.1	41.2	100	2.2	12.3	35.8	100	2.3	13.1	38.2
DD1	100	7	32.3	127.7	100	4.9	23.5	81.9	100	4.5	22.1	80.7	100	3.9	19.2	70.2	100	4.2	20.5	75.0
QV1	100	11.2	48.7	197	100	11.6	50.6	207.8	100	10.8	47.4	192.3	100	9.4	41.2	167.3	100	10.0	43.9	178.5
IM1	100	10.2	61.2	133.5	100	6.1	28.4	112.1	100	6	28	110	100	5.2	24.3	95.7	100	5.6	26.0	102.3

Table 3.2: Performance of DY, HS+, HSDY, HSPRP, and DYPRP methods on a set of test problems

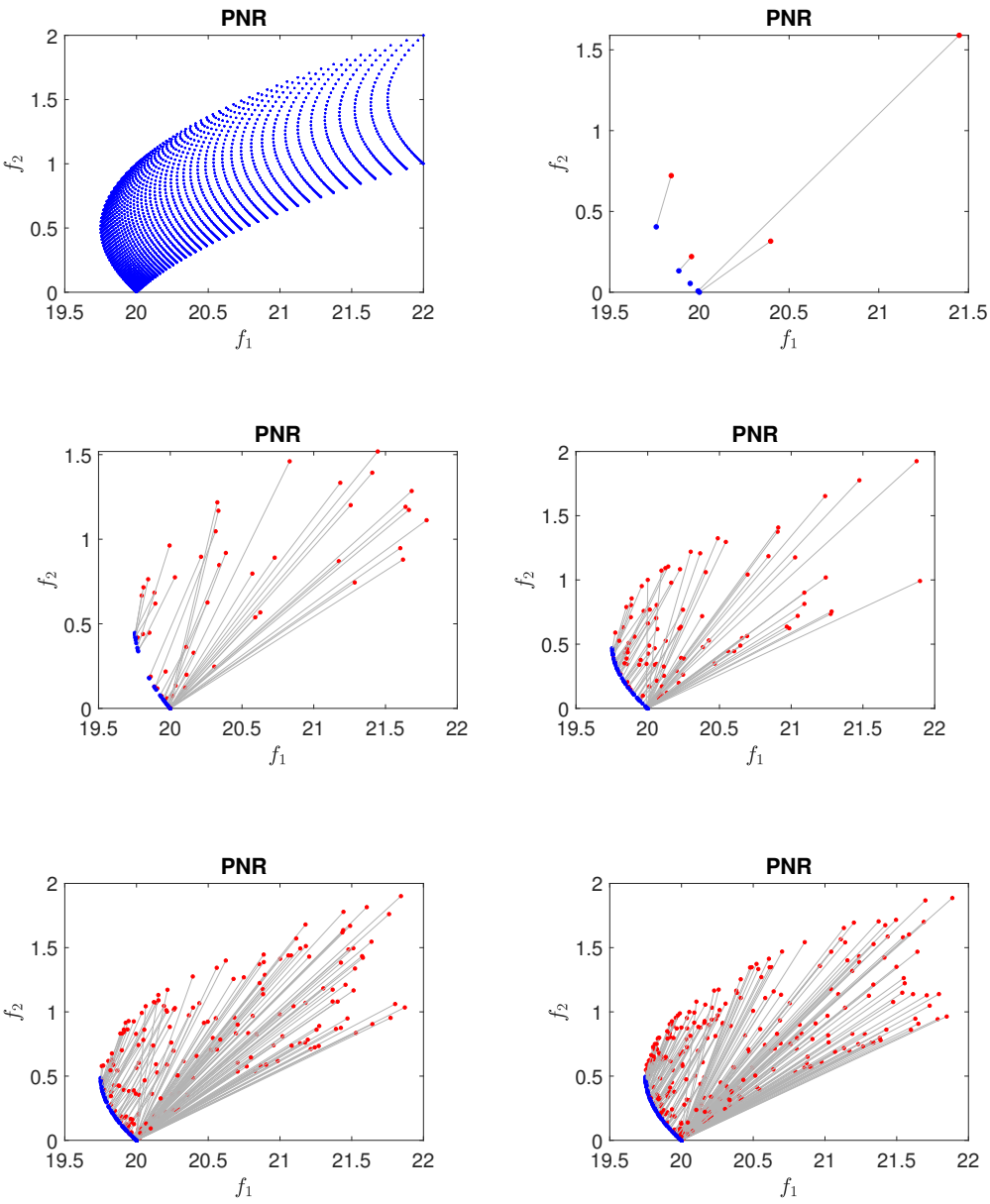


Figure 3.1: Image of the final iterates in the objective space obtained by the HSDY method using 5, 50, 100, 200, and 300 randomly generated starting points for the PNR problem. A blue point shows a final iterate while the red point at the other end of straight lines represent the corresponding starting points in the objective space.

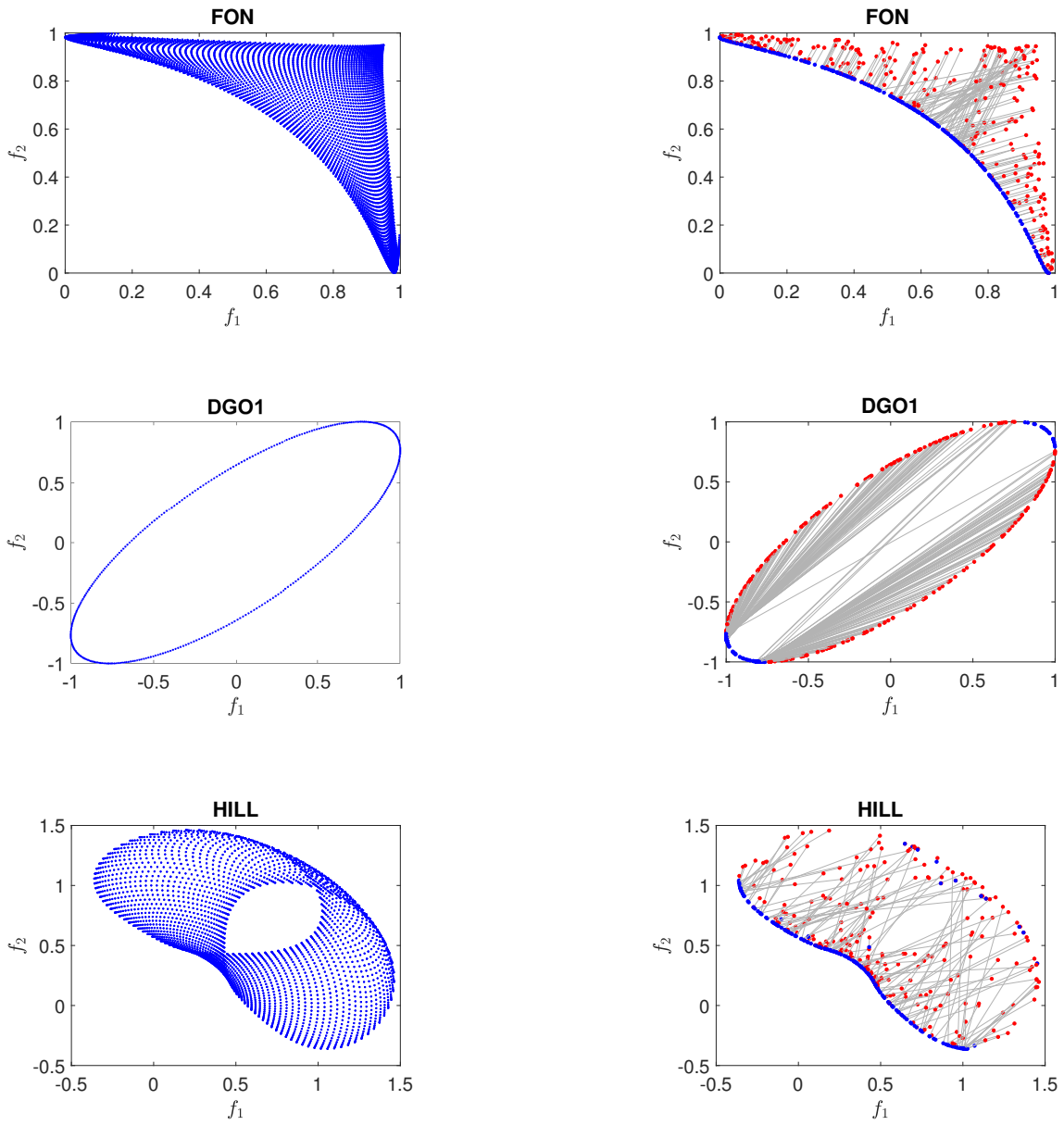


Figure 3.2: Image of the final iterates in the objective space obtained by the HSDY method using 150 randomly generated starting points for the problems FON, DGO1, HILL LE1, MOP3, PNR, KW2, FAR1, and SLCDT1. A blue point shows a final iterate while the red point at the other end of straight lines represent the corresponding starting points in the objective space.

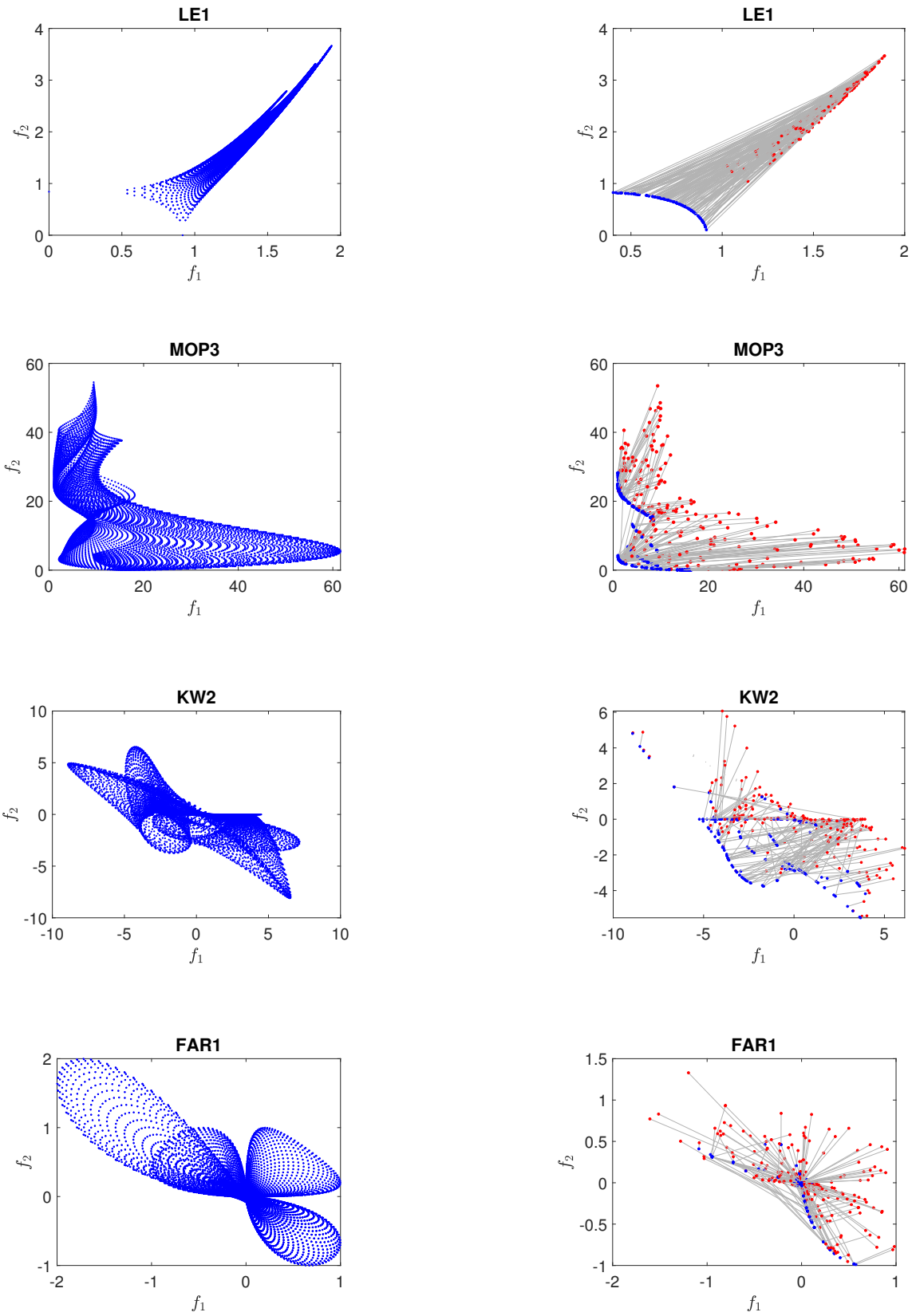


Figure 3.2: continued from Figure 3.2

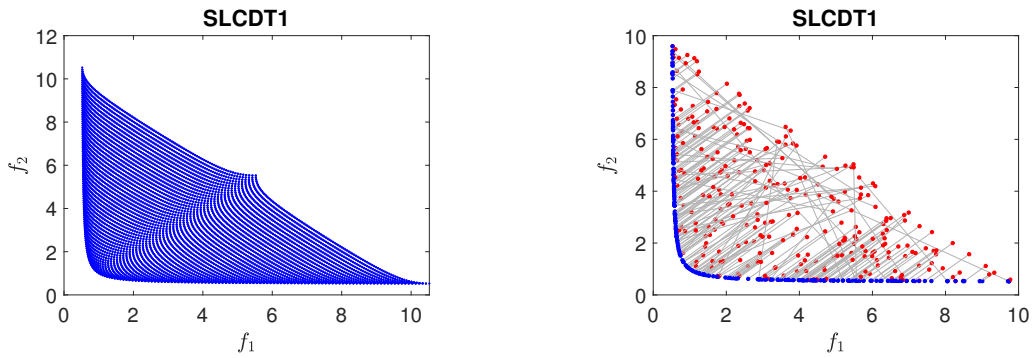
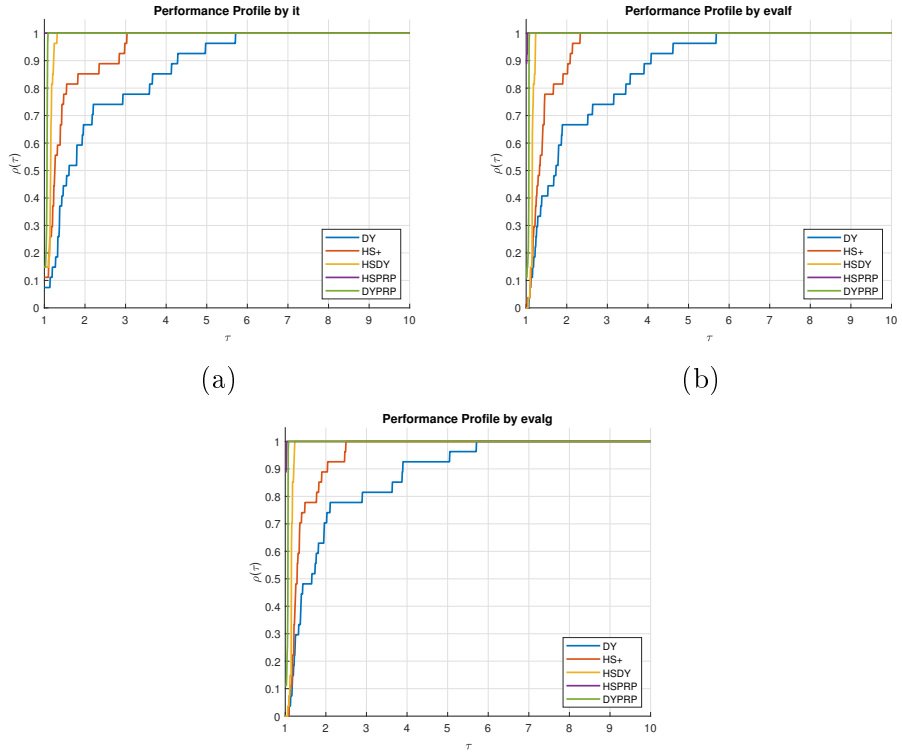


Figure 3.2: continued from Figure 3.2

Measure	DY	HS+	HSDY	HSPRP	DYPRP
it	1.681	1.231	1	0.863	0.912
evalf	1.525	1.206	1	0.872	0.921
evalg	1.587	1.206	1	0.871	0.919

Table 3.3: Relative efficiency (see (3.30)) of DY, HS+, HSDY, HSPRP, DYPRP methods measured by it, evalf, and evalg



(a) (b)

(c)

Figure 3.3: Performance profile by (a) it, (b) evalf, (c) and evalg

By an application, we show the proposed method can be nice tool to find the the optimal control measures for non-linear mathematical models.

3.7 An Application

In our society, many of the diseases become endemic due to the lack of adequate policies and timely interventions to mitigate their spread. As a result, proactive measures are needed to combat outbreaks of infectious diseases. These include both vaccines and cures for diseases that can be controlled. The current section provides an application of the proposed technique to an SIR epidemiological model [153] with vaccination and treatment as their controls.

3.7.1 Model Formulation

For our application, we consider a standard SIR model with bilinear incidence, as described by Yusuf et al. [153]. In this model, the total population is variable, which allows for a more realistic representation of population dynamics during an epidemic.

Based on the model system described in equation (3.31), the population is divided into three distinct categories:

- **Susceptible (S):** This group represents individuals who are not infected but are capable of contracting the disease if exposed.
- **Infected (I):** This category includes individuals who have contracted the disease and are capable of spreading it to susceptible individuals.
- **Recovered (R):** This group consists of individuals who have been removed from the infection cycle, either through recovery from the disease (which confers permanent immunity) or through vaccination.

The total population N at any given time is the sum of these three categories:

$$N = S + I + R.$$

A key assumption in this model is that individuals can acquire immunity against the disease through two pathways: recovery after treatment or vaccination. This assumption allows for the exploration of different intervention strategies.

To control the spread of the disease, we introduce two time-dependent control variables:

- **Vaccination rate ($u_1(t)$):** This represents the proportion of the susceptible population that is vaccinated per unit time. Effective vaccination moves individuals directly from the susceptible to the recovered category.
- **Treatment rate ($u_2(t)$):** This denotes the proportion of infected individuals that receive treatment per unit time. Successful treatment accelerates the movement of individuals from the infected to the recovered category.

These control variables allow us to model and optimize intervention strategies, such as vaccination campaigns and treatment programs, to mitigate the spread of the disease. By adjusting these controls, we can explore various scenarios and determine optimal strategies for disease management.

Building upon the work of Yusuf and Benyah [153], we present a controlled SIR model incorporating two time-dependent control variables, $u_1(t)$ and $u_2(t)$. The dynamics of this system are described by the following set of nonlinear, time-varying state equations:

$$\begin{cases} \dot{S} = b - \beta SI - dS - u_1(t)S, \\ \dot{I} = \beta SI - u_2(t)I - dI - \alpha I, \\ \dot{R} = u_1(t)S + u_2(t)I - dR, \end{cases} \tag{3.31}$$

with the initial conditions

$$S(0) = 0.95,\ I(0) = 0.05 \text{ and } R(0) = 0. \tag{3.32}$$

The parameters used in the system (3.31) are crucial for accurately modeling the disease dynamics. Table 3.4 provides a detailed description of these parameters along with their values:

Symbols	Description	Value
b	Immigration rate of susceptible population	3×10^{-2}
β	Disease transmission rate	$25 \times 10^{-2}, 75 \times 10^{-2}$
d	Natural death rate	0.02
α	Disease-induced death rate	0.1
T	Total simulation duration	5 yr

Table 3.4: Description and values of the parameters in the model problem (3.31)

It is worth noting that we have adopted the values for rates b, d, α, and β from the study by Yusuf and Benyah [153]. The disease transmission rate β is given two values, allowing for the exploration of scenarios with different levels of disease transmissibility.

3.7.2 Optimal Control Problem Formulation

Our primary objective is to determine the optimal control strategies u_1^* and u_2^* that minimize the impact of the epidemic while considering the cost of interventions. These optimal controls should satisfy two key conditions:

(i) They should generate state trajectories S^*, I^*, and R^* that solve the system defined by equations (3.31) and (3.32) over the time interval $[0, T]$.

(ii) They should minimize the following objective functional:

$$J(u_1, u_2) = \int_0^T \left[I(t) + P_1 u_1^2(t) + P_2 u_2^2(t) \right] dt, \tag{3.33}$$

where T represents the final time of the simulation. The constants P_1 and P_2 denote the relative costs associated with the vaccination and treatment interventions, respectively. These weights reflect the relative importance and resource requirements of each control measure.

The objective functional J encompasses two main components:

- The number of infectious individuals (I).
- The implementation costs of vaccination ($P_1 u_1^2$) and treatment ($P_2 u_2^2$) strategies.

It is worth noting that this objective functional differs from the one presented in the study by Yusuf and Benyah [153], as we have tailored it to our specific research goals.

The control functions are constrained to be Lebesgue measurable and bounded between 0 and 1, representing the proportion of the population undergoing vaccination or treatment at any given time. Formally, we define the set of permissible control functions as

$$\eta = \left\{ (u_1, u_2) : 0 \leq u_1(t), u_2(t) \leq 1, \forall t \in [0, T] \right\}.$$

Thus, our optimal control problem can be succinctly stated as:

$$J(u_1^*, u_2^*) = \underset{(u_1, u_2) \in \eta}{\text{minimize}}\ J(u_1, u_2). \tag{3.34}$$

This approach aims to achieve two main goals: (i) minimizing the number of infected individuals and (ii) reducing the cost of implementation of vaccination and treatment policies.

3.7.3 Multiobjective Optimization Approach

3.7.3.1 Limitations of Singleobjective Optimal Control

Traditional optimal control theory, while useful, has a notable drawback when applied to problem (3.34). It typically yields only a single optimal solution, as noted in studies by Verma et al. [148] and Misra et al. [110]. This approach requires the decision-maker to pre-determine weights (P_1 and P_2) for different aspects of the problem, as discussed in works by Silva et al. [36] and Yusuf et al. [153]. However, selecting appropriate values for P_1 and P_2 is challenging and often requires prior knowledge or preferences that may not be available. Moreover, a single optimal solution may not provide a comprehensive understanding of potential strategies and their impacts on system dynamics. Consequently, many valuable alternative solutions remain unexplored when using conventional optimal control methods.

3.7.3.2 Proposed Multiobjective Optimization Approach

To address these limitations, we propose a multiobjective optimization approach. This method aims to generate a set of Pareto optimal solutions, offering a more comprehensive view of the trade-offs involved in the problem.

We decompose the original cost functional (3.33) into two distinct objectives:

$$f_1(u_1, u_2) = \int_0^T I(t)dt \quad \text{and} \tag{3.35}$$

$$f_2(u_1, u_2) = \int_0^T \left(u_1^2(t) + u_2^2(t)\right) dt \tag{3.36}$$

Here, f_1 represents the total number of infected individuals over the time period $[0, T]$, while f_2 quantifies the cost associated with implementing the control strategies during the same period.

Based on this decomposition, we reformulate the original problem as a biobjective

optimization problem:

$$\left.\begin{array}{ll} \text{minimize} & \big(f_1, f_2\big) \\ \text{subject to} & \dot{S} = b - \beta SI - dS - u_1(t)S, \\ & \dot{I} = \beta SI - u_2(t)I - dI - \alpha I, \\ & \dot{R} = u_1(t)S + u_2(t)I - dR, \\ & S(0) = 0.95,\ I(0) = 0.05,\ R(0) = 0, \\ & 0 \leq u_1 \leq 1, \\ & 0 \leq u_2 \leq 1. \end{array}\right\} \tag{3.37}$$

In this formulation, we have taken $P_1 = P_2 = 1$. The physical significance of taking equal weights ($P_1 = P_2 = 1$) is that we are giving equal importance to both the medical and economic aspects of the problem. This means that in our optimization process, reducing the total number of infected individuals over time is considered just as important as minimizing the cost of implementing control strategies. This balanced approach ensures that neither the public health concerns nor the economic impacts are prioritized over the other, allowing for solutions that seek an equilibrium between these two crucial factors in epidemic management.

The objectives f_1 and f_2 explicitly represent the medical and economic perspectives of the problem, respectively. This approach allows for a more comprehensive exploration of the solution space, providing decision-makers with a range of options that balance disease control and economic considerations.

3.7.3.3 Algorithm For Solving The Biobjective Optimization Problem (3.37)

To solve the biobjective optimal control problem (3.37), we propose the following algorithm. This procedure aims to obtain a discrete approximation of the Pareto set, providing a comprehensive view of the trade-offs between minimizing infections and

control costs.

Algorithm 4 Pareto Set Approximation for Biobjective Optimal Control Problem (3.37)

1: **Numerical Integration:**

Solve the system of ODEs (3.31) using fourth-order Runge-Kutta method

Time interval: $[0, T]$, where $T = 5$ years

2: **Control Discretization:**

Divide $[0, T]$ into 80 equally spaced time intervals for u_1 and u_2

3: **Objective Function Approximation:**

Approximate integrals in (3.35) and (3.36) using trapezoidal rule

Obtain discrete approximations of f_1 and f_2

4: **Quadratic Approximation:**

Apply curve fitting to obtain quadratic approximations of f_1 and f_2

5: **Pareto Set Computation:**

Apply Algorithm 3, with $\beta_k = \beta_k^{HSPRP}$ to solve:

minimize (f_1, f_2) subject to $0 \leq u_1 \leq 1$, $0 \leq u_2 \leq 1$

Obtain discrete approximation of the complete Pareto set

Detailed Explanation:

1. **Numerical Integration:** We begin by solving the system of ordinary differential equations (3.31) numerically. The fourth-order Runge-Kutta method is chosen for its accuracy and stability. This step provides the time evolution of the state variables (S, I, R) over the specified time interval of 5 years.

2. **Control Discretization:** The control variables $u_1(t)$ and $u_2(t)$ are discretized into 80 equal time intervals. This discretization allows us to approximate the continuous control functions with a finite set of decision variables, making the problem computationally tractable.

3. **Objective Function Approximation:** The integrals in equations (3.35) and

(3.36), which define our objective functions, are approximated using the trapezoidal rule. This numerical integration technique provides discrete approximations of f_1 and f_2, which represent the total number of infections and the total control cost, respectively.

4. **Quadratic Approximation:** To further simplify the optimization process, we apply curve fitting techniques to obtain quadratic approximations of f_1 and f_2. The smoothness of these objective functions enhances their compatibility with the proposed algorithm.

5. **Pareto Set Computation:** Finally, we apply Algorithm 2 to solve the biobjective optimization problem (3.37). This algorithm is designed to find a discrete approximation of the complete Pareto set, representing the range of optimal trade-offs between minimizing infections and control costs.

The resulting Pareto set provides decision-makers with a comprehensive view of the possible optimal strategies, allowing them to choose a solution that best balances the competing objectives based on their preferences and constraints.

3.7.3.4 Experimental Results

In this section, we present our analysis of the optimal solutions to the problem (3.37), considering how these solutions change as we vary the parameter β. This parameter represents the transmission coefficient of the disease, which is a crucial factor in epidemiological modeling.

We begin by examining the criterion feasible region and Pareto front of problem (3.37) for two distinct values of β: 0.25 and 0.75. These results are visualized in Figures 6.3a and 3.4b for $\beta = 0.25$, and in Figures 6.3c and 3.4d for $\beta = 0.75$. These figures provide a comprehensive view of how the feasible solutions and optimal trade-offs change with different transmission rates.

To facilitate a direct comparison between these two scenarios, we have combined the Pareto fronts from Figures 3.4b and 3.4d into a single graph, presented in Figure

3.5a. This consolidated view allows us to observe the trade-off curves for both values of the transmission coefficient β simultaneously.

A notable feature in Figure 3.5a is the presence of a common point shared by both curves in the objective space. This point represents what we might call the *medical perspective*–a scenario where the primary focus is on minimizing the number of infected individuals. At this point, we can infer that maximum Pareto optimal control measures have been implemented to control the spread of infection.

Conversely, when we look at the point where $f_2 = 0$ in Figure 3.5a, we see the *economic perspective*-a situation where the sole focus is on minimizing costs. This approach, unfortunately, leads to an uncontrolled spread of the disease and a consequent increase in the number of infected individuals for both values of β.

An interesting observation from Figure 3.5a is that as β increases, so does the difference between the best and worst values of f_1. This suggests that the impact of control measures becomes more pronounced as the disease becomes more transmissible.

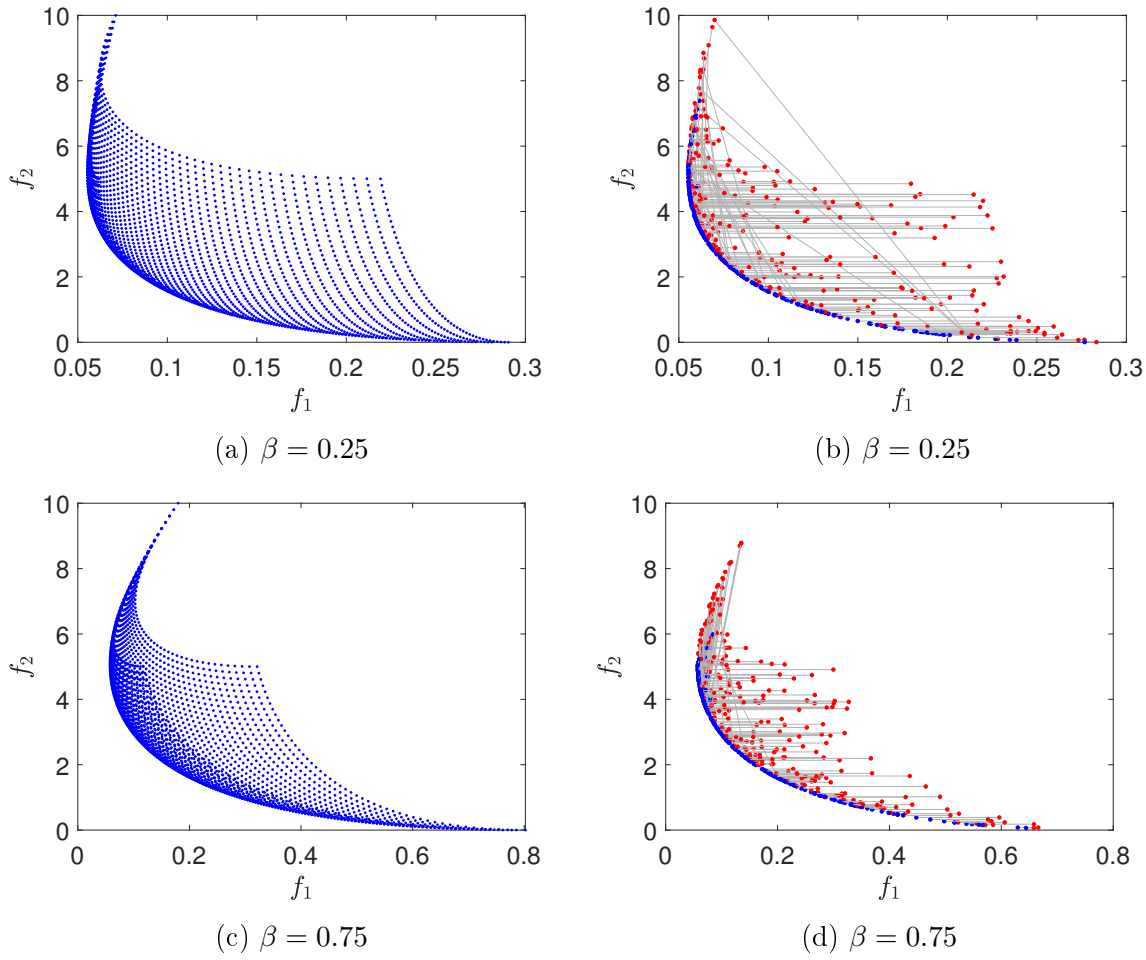
Figure 3.5a also illustrates how the implementation of Pareto optimal control policies affects the number of infected individuals differently for the two considered values of β. These various scenarios along the Pareto front represent different trade-offs between economic considerations and public health outcomes.

To further elucidate our findings, we present Figure 3.5b, which plots the optimal values of f_1 corresponding to different optimal control strategies (u_1^*, u_2^*) for both values of β. This figure provides insight into how the effectiveness of control strategies varies with the transmission rate.

A key observation from Figure 3.5b is that when no controls are implemented, the value of f_1 (representing the number of infected individuals) is lower for $\beta = 0.25$ compared to $\beta = 0.75$. This aligns with our intuition that a lower transmission rate naturally results in fewer infections.

In conclusion, our analysis demonstrates the complex interplay between disease

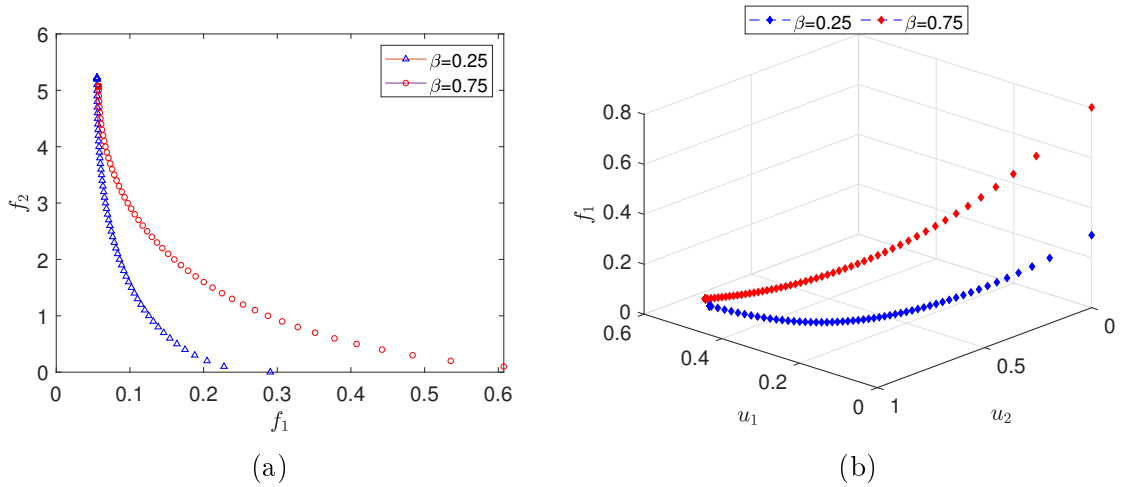
transmission rates, control strategies, and the resulting trade-offs between public health outcomes and economic costs. These insights can provide valuable guidance for policy-makers in crafting effective responses to epidemiological challenges.



(a) $\beta = 0.25$ (b) $\beta = 0.25$

(c) $\beta = 0.75$ (d) $\beta = 0.75$

Figure 3.4: (a)–(d) show the image sets and function value spaces of the problem (3.37) for $\beta = 0.25,\ 0.75$, obtained by the HSPRP method using 150 randomly generated starting points. A blue point indicates the final iterate whereas the red point at the other end of straight lines represent the corresponding starting points



(a) (b)

Figure 3.5: (a) Trade-off curve for two different values of β, (b) Representation of f_1 as a function of Pareto optimal strategies u_1 and u_2 for two values of β

3.8 Conclusion

This chapter presents significant advancements in the application of conjugate gradient methods to unconstrained MOPs. We introduce three novel hybrid approaches: the monotone HSDY, nonmonotone HSPRP, and DYPRP conjugate gradient methods, all specifically designed for MOPs. These methods incorporate the strong Wolfe line search technique to determine optimal step sizes, enhancing both efficiency and reliability.

Our work extends the properties of hybrid conjugate gradient methods into the domain of multiobjective optimization, representing a considerable advancement in the field. Specifically, we have rigorously proven that the search directions d^k in the HSDY method satisfy the sufficient descent condition (3.4) under the strong Wolfe line search conditions. This crucial property ensures the algorithm's effectiveness in finding optimal solutions. Additionally, we have established the global convergence of the proposed methods under a set of assumptions that naturally extend from those in the scalar optimization case, providing a solid theoretical foundation for our approach.

To validate the performance of our proposed methods, we conducted comprehensive comparisons with established conjugate gradient methods, specifically the DY and

HS+ algorithms. The empirical results strongly support the efficacy of the nonmonotone HSPRP method in solving practical MOPs. Among the methods, HSPRP and DYPRP proved to be the most efficient across various problems, requiring fewer iterations, function evaluations, and gradient evaluations. HSDY also performed well but was slightly less efficient than HSPRP and DYPRP, while HS+ showed moderate performance, and DY was the least efficient method. These results are detailed in Table 3.3.

To provide a visual representation of the comparative performance, we employed the performance profile technique introduced by Dolan and Moré [37]. The resulting graphs, illustrated in Figures 3.3a–3.3c, clearly demonstrate the superiority of the HSDY method over the DY and HS+ conjugate gradient methods for the considered multiobjective optimization test problems.

A key application of our proposed method is showcased in the field of epidemiology. We applied the HSPRP method to an SIR epidemiological model, incorporating both vaccination and treatment as control measures. The primary objective was to determine optimal control strategies that simultaneously minimize the number of infected individuals and the associated implementation costs. In this study, we examined the model's behavior under two different disease transmission rates: $\beta = 0.25$ and $\beta = 0.75$. The resulting trade-off curves revealed an important insight: as the transmission rate increases, so does the disparity between the best and worst-case scenarios for the objective function f_1, which represents the number of infected individuals. This finding underscores the critical importance of early intervention in high-transmission scenarios to mitigate the spread of the disease.

In conclusion, our research not only advances the theoretical understanding of conjugate gradient methods in multiobjective optimization but also provides a powerful, practical tool for solving complex real-world problems. The HSPRP method's demonstrated superiority in both theoretical properties and empirical performance, coupled

with its successful application to a relevant epidemiological model, positions it as a valuable addition to the optimization toolkit for researchers and practitioners across various fields.

Future research could focus on extending the HSPRP method to uncertain MOPs. Investigating the method's performance under different uncertainty representations, such as interval-based or scenario-based uncertainties, could broaden its applicability. Additionally, developing techniques to balance robustness and optimality in the presence of uncertainties could enhance the method's practical value for decision-making under real-world uncertainties.

****