Chapter 1

Introduction

Optimization is the process of identifying the best alternative or the most efficient. It involves changing some aspects of a system or a problem to obtain the best outcomes and reduce the unwanted ones. In mathematics, optimization means finding the best possible alternative from a set of options. It requires defining a function that measures the quality of a solution and finding the input values that maximize or minimize that function. Mathematical optimization is a powerful tool for modeling and solving complex problems in various fields, such as computer science, engineering, operations research, and economics.

Mathematically, optimization models comprise three significant components: decision variables, objective function, and constraints.

- *Decision variables* designate a value that may vary within the scope of a given optimization problem.

- In a mathematical optimization problem, the *objective function* expresses the problem's main criteria, whose value is either minimized or maximized over the set of feasible alternatives.

- *Constraints* are the logical conditions or allowable values or scopes for the variables in an optimization problem that the solution of a given problem must satisfy.

Optimization is a branch of applied mathematics, computational mathematics, and operations research that deals with finding the best possible solutions to complex problems. These problems involve decision variables, an objective function, and constraints that define the feasible region. The nature of these mathematical relationships affects the difficulty and the methods of optimization. Optimization has many applications in science, engineering, business management, military, and space technology. It requires studying the optimality conditions of the problems, building model problems, choosing the algorithmic method of the solution, proving the convergence theory of the algorithms, and testing them with typical and real-life problems.

One of the most common and primary problems in scientific research and engineering practice is optimization problems. According to the number of the components of the objective function, optimization problems can be classified into two categories: singleobjective and multiobjective. Although a substantial amount of research in optimization is conducted concerning singleobjective problems, optimization problems with more than one objective are inevitable in various fields, such as engineering design [3], optimal control systems [120], chemical engineering [132], machine learning [83], and many more. Multiobjective optimization is a branch of mathematical optimization that deals with problems that have more than one objective function to be optimized simultaneously. For example, when designing a car, one might want to minimize the cost, maximize the performance, and reduce the environmental impact. These objectives are often conflicting, meaning that improving one objective may worsen another. Therefore, there is usually no single optimal solution that satisfies all objectives but rather a set of trade-off solutions that are equally good in some sense. These solutions are called Pareto optimal or nondominated, and they form the Pareto frontier or Pareto set.

This chapter describes the principles and basic concepts of multiobjective optimization. This is followed by a discussion of some classical optimization techniques to obtain the Pareto optimal solutions and, in particular, their potential disadvantages.

Thereafter, the motivation and the contribution of the thesis will be discussed.

In the next section, the fundamentals of MOPs are introduced.

1.1 Multiobjective Optimization Problem

In general, a *constraint* MOP includes a set of n decision variables, a set of m objective functions, a set of $\mathcal{P}$ inequality constraints, and a set of $\mathcal{Q}$ equality constraints. Without loss of generality, a minimization MOP is defined as

$$\left.\begin{array}{ll} \text{minimize} & F(x) = \big(f_1(x), f_2(x), \ldots, f_m(x)\big)^\top, \ m \geq 2 \\ \text{subject to} & g_i(x) \leq 0, \ i = 1, 2, \ldots, \mathcal{P}, \\ & h_j(x) = 0, \ j = 1, 2, \ldots, \mathcal{Q}, \end{array}\right\} \tag{1.1}$$

where $f_r : \mathbb{R}^n \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$ and $h_j : \mathbb{R}^n \to \mathbb{R}$ for all $r = 1, 2, \ldots, m$, $i = 1, 2, \ldots, \mathcal{P}$ and $j = 1, 2, \ldots, \mathcal{Q}$. We refer to $x = (x_1, x_2, \ldots, x_n)^\top$ as the vector of decision variables and the set
$\mathcal{X} = \{x \in \mathbb{R}^n : g_i(x) \leq 0, \ i = 1, 2, \ldots, \mathcal{P}, \ h_j(x) = 0, \ j = 1, 2, \ldots, \mathcal{Q}\}$ as the feasible set of the problem (1.1) in the decision space $\mathbb{R}^n$. The set $\mathcal{X}$ is also called *decision feasible region*. We denote the image of $\mathcal{X}$ under the vector-valued objective function F by $\mathcal{Y}$, i.e.,
$\mathcal{Y} = F(\mathcal{X}) = \Big\{\big(f_1(x), f_2(x), \ldots, f_m(x)\big)^\top : x \in \mathcal{X}\Big\}$. The set $\mathcal{Y}$ is referred as the feasible set in the objective space $\mathbb{R}^m$ or *objective feasible region*. Note that $\mathcal{X}$ is a subset of $\mathbb{R}^n$ and $\mathcal{Y}$ is a subset of $\mathbb{R}^m$.

One can note that if $\mathcal{P} = \mathcal{Q} = 0$, then the MOP (1.1) is reduced to an *unconstrained* MOP. In general, an unconstrained MOP includes a set of n decision variables and a set of m objective functions. An unconstrained MOP is mathematically defined by the following:

$$\text{minimize } F(x) = \big(f_1(x), f_2(x), \ldots, f_m(x)\big)^\top, x \in \mathbb{R}^n. \tag{1.2}$$

Geometrical view of the sets $\mathcal{X}$ and $\mathcal{Y}$ for biobjective optimization problem with three decision variables are shown in Figure 1.1.

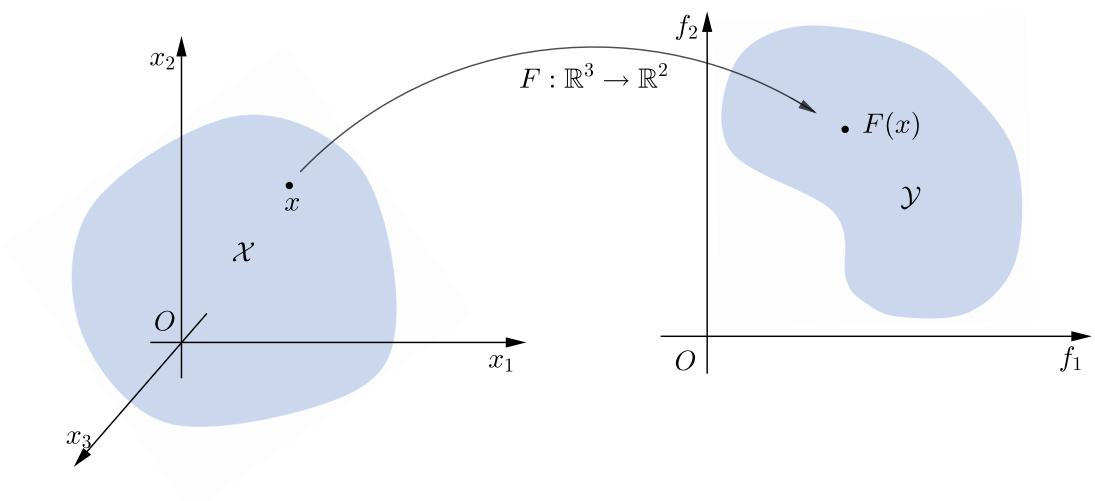

Figure 1.1: Decision Feasible region (left) and objective feasible region (right). F is a bi-objective function that maps the three-dimensional decision feasible region to the two-dimensional objective feasible region.

When dealing with MOPs, it is necessary to clarify what 'solving' an MOP means, as it involves identifying trade-offs among multiple conflicting objectives. This requires defining various types of solutions, such as Pareto optimal sets, which inform decision-making in complex scenarios. Understanding these concepts sets the stage for the subsequent discussions on solution methods. In the next section, some definitions are provided to explain the concept of solution for (1.1).

1.2 Pareto Optimality

It is rarely the case that there is a single point that simultaneously optimizes all the objective functions. Therefore, we normally look for 'trade-offs', rather than single solutions when dealing with MOPs. The notion of 'optimality' is therefore, different from single objective optimization problems in MOPs. The most commonly adopted notion of optimality is originally proposed by Francis Ysidro Edgeworth [52] and later generalized by Vilfredo Pareto [118]. The definition of Pareto optimality is based on a dominance relation in $\mathbb{R}^m$. For the required dominance relation for Pareto optimality, we use the following notations:

- $\mathbb{R}_+$, the set of nonnegative real numbers.
- $\mathbb{R}_{++}$, the set of positive real numbers.

Throughout this thesis, the notations $\succeq$ and $\succ$ stands for the following orders on $\mathbb{R}^m$:

$$y_1 \succeq y_2 \iff y_1 - y_2 \in \mathbb{R}^m_+,$$
$$y_1 \succ y_2 \iff y_1 - y_2 \in \mathbb{R}^m_{++}.$$

Suppose we have a set $K = \{k_1, k_2, \ldots\} \subseteq \mathbb{N}$, where $k_j < k_{j+1}$ for all $j \in \mathbb{N}$. In this case, we use the notion $K \underset{\infty}{\subset} \mathbb{N}$ to indicate that K is an increasing sequence of natural numbers. Furthurmore, for any vector $y \in \mathbb{R}^n$, the Euclidean norm of y is denoted as $\|y\| = \sqrt{\langle y, y \rangle}$, where $\langle y, x \rangle$ denotes their usual inner product of $y, x \in \mathbb{R}^n$.

Definition 1.1 *(Pareto optimality (e.g. [40])). A feasible solution $\hat{x} \in \mathcal{X}$ is called efficient or Pareto optimal if there is no $x \in \mathcal{X}$ such that $F(x) \preceq F(\hat{x})$. If $\hat{x}$ is efficient, $F(\hat{x})$ is called a nondominated point.*

The set of all efficient solutions of the MOP (1.1) is denoted by $\mathcal{X}_E$. The set of all nondominated points is represented by $\mathcal{Y}_N$. Evidently, $\mathcal{Y}_N = F(\mathcal{X}_E)$.

Definition 1.2 *(Weak Pareto optimality (e.g. [40])). A feasible solution $\hat{x} \in \mathcal{X}$ is called weakly efficient or weakly Pareto optimal if there is no $x \in \mathcal{X}$ such that $F(x) \prec F(\hat{x})$. The point $\hat{y} = F(\hat{x})$ is then said to be weakly nondominated.*

The set of all weakly efficient solutions of the MOP (1.1) is denoted by $\mathcal{X}_{wE}$. The collection of all weakly nondominated points is represented by $\mathcal{Y}_{wN}$.

The solution notion for MOP (1.1) can also be defined with respect to an ordering cone $\mathbb{R}^m_{\preceq}$ which is used for ordering the criterion space $\mathbb{R}^m$. Utilizing the ordering cone $\mathbb{R}^m_{\preceq}$, a feasible solution $\hat{x} \in \mathcal{X}$ is called an efficient solution of MOP (1.1) if

$$\left(F(\hat{x}) - \mathbb{R}^m_{+}\right) \bigcap F(\mathcal{X}) = \left\{F(\hat{x})\right\}.$$

Likewise, we say $\hat{x} \in \mathcal{X}$ is a weakly efficient solution of MOP (1.1) if

$$\left(F(\hat{x}) - \mathbb{R}^m_{++}\right) \bigcap F(\mathcal{X}) = \emptyset.$$

Apparently, every efficient point is also weakly efficient [40]. The two concepts are illustrated in the feasible objective space ($\mathcal{Y}$) for a biobjective optimization problem ($\mathcal{P} = 2$) in Figure 1.2. It shows an efficient point $\hat{x}^1$ and a weakly efficient point $\hat{x}^2$.

A point in the objective space, which is often referenced in multiobjective optimization methods, are the ideal point (or utopia point). This is formally defined in Definition 1.3 [40].

Definition 1.3 *The point $y^I = \left(y^I_1, y^I_2, \ldots, y^I_m\right)$ given by*

$$y^I_i = \min_{x \in \mathcal{X}} f_i(x)$$

is ideal point (or utopia point) of the MOP (1.1).

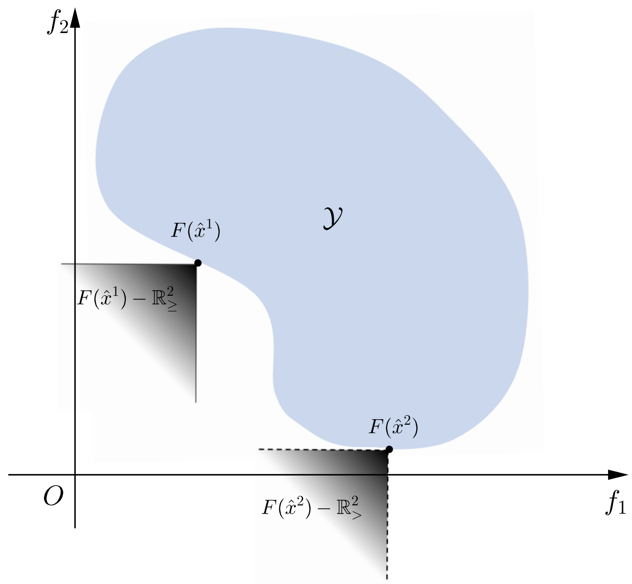

Figure 1.2: Nondominated and weakly nondominated points

1.3 Pareto Critical Points

Under the assumption that each f_m is continuously differentiable in this part we present the first-order optimality condition [64] for problem (1.2). The following condition is necessary but not sufficient condition for Pareto optimality of $\hat{x} \in \mathbb{R}^n$:

$$(-\mathbb{R}^m_{++}) \cap \text{Image}(JF(\hat{x})) = \emptyset, \tag{1.3}$$

where $JF(x) \in \mathbb{R}^{m \times n}$ represents the Jacobian of F at x and the image set of $JF(x)$ is denoted as Image$(JF(x))$. A point $\hat{x} \in \mathbb{R}^n$ that follows (1.3) is termed as *Pareto critical* or *stationary*. If x is not Pareto critical, there exists $w \in \mathbb{R}^n$ such that $JF(x)w \in -(\mathbb{R}^m_{++})$. Such vector w is called as a *descent direction* for F at x (see [64]). To characterize descent directions of F, define $\mathcal{M}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ by

$$\mathcal{M}(x,d) = \max_{j=1,2,\ldots,m} \langle \nabla f_j(x), d\rangle. \tag{1.4}$$

The function $\mathcal{M}$ can also be defined as

$$\mathcal{M}(x,d) = \sup\{\langle \nabla JF(x)d, w\rangle : w \in \Omega\}, \tag{1.5}$$

where Ω is the canonical basis of $\mathbb{R}^m$. Note that d is a descent direction of F at $\bar{x}$ if and only if $\mathcal{M}(\bar{x}, d) < 0$ (see [49]). The following result depicts some other significant characterstic of $\mathcal{M}$.

Lemma 1.1 (See [64, Lemma 2]). For a given continuously differentiable function $F\colon \mathbb{R}^n \to \mathbb{R}^m$ and the function $\mathcal{M}\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as in (1.4), it holds that

(i) $\mathcal{M}(x, y + \alpha d) \leq \mathcal{M}(x,y) + \alpha\mathcal{M}(x,d)$ for any $x, y, d \in \mathbb{R}^n$ and $\alpha \geq 0$.

(ii) the map $(x,d) \mapsto \mathcal{M}(x,d)$ is continuous, and

(iii) we have $|\mathcal{M}(x,d) - \mathcal{M}(y,d)| \leq \|JF(x) - JF(y)\|_2 \|d\|_2$ for any $x, y, d \in \mathbb{R}^n$. As a consequence, if JF is L-Lipschitz continuous, then $\mathcal{M}(\cdot, d)$ is $L\|d\|_2$-Lipschitz continuous.

In the following, we recall the extensions of the steepest descent method for MOPs. We suggest to read [49] for further information on steepest descent method for MOPs. For a given $x \in \mathbb{R}^n$, consider the scalar-valued problem:

$$\min_{d \in \mathbb{R}^n} \left(\mathcal{M}(x,d) + \tfrac{1}{2}\|d\|_2^2\right). \tag{1.6}$$

Since the objective function of (1.6) is a closed convex function, due to which (1.6) always has a unique optimal solution. The solution of (1.6) is denoted by $d_{SD}(x)$ and its optimal value by $\Theta_{SD}(x)$, i.e.,

$$d_{SD}(x) = \arg\min\left\{\mathcal{M}(x,d) + \tfrac{\|d\|_2^2}{2} : d \in \mathbb{R}^n\right\}, \tag{1.7}$$

and

$$\Theta_{SD}(x) = \mathcal{M}(x, d_{SD}(x)) + \tfrac{1}{2}\|d_{SD}(x)\|_2^2. \tag{1.8}$$

Here, the direction $d_{SD}(x)$ is single-valued and well-defined due to the strong convexity of the objective function $\mathcal{M}(x,d) + \frac{\|d\|_2^2}{2}$ in the variable d. This property ensures a unique minimizer $d_{SD}(x)$. The direction $d_{SD}(x)$ extends the notion of the conventional SD direction to the multiobjective optimization case. For single objective minimization case, i.e., if $F\colon \mathbb{R}^n \to \mathbb{R}$, then we get $\mathcal{M}(x,d) = \langle \nabla F(x), d\rangle$, $d_{SD}(x) = -\nabla F(x)$ and $\Theta_{SD}(x) = -\frac{1}{2}\|\nabla F(x)\|_2^2$. The direction $d_{SD}(x)$ and the optimum value $\Theta_{SD}(x)$ can be utilized, specifically, to characterize stationary points of (1.2).

Lemma 1.2 (See [49, Lemma 3.3]). Let $d_{SD} : \mathbb{R}^n \to \mathbb{R}^n$ and $\Theta_{SD} : \mathbb{R}^n \to \mathbb{R}$ be given by (1.7) and (1.8), respectively. Then, the following holds:

(i) if $\bar{x}$ is Pareto critical, then $d_{SD}(\bar{x}) = 0$ and $\Theta_{SD}(\bar{x}) = 0$,

(ii) if $\bar{x}$ is not Pareto critical, then $d_{SD}(\bar{x}) \neq 0$, $\Theta_{SD}(\bar{x}) < 0$, $\mathcal{M}(\bar{x}, d_{SD}(\bar{x})) < -\frac{1}{2}\|d_{SD}(\bar{x})\|_2^2 < 0$, and $d_{SD}(\bar{x})$ is a descent direction for F at $\bar{x}$, and

(iii) the maps d_{SD} and Θ_{SD} are continuous on $\mathbb{R}^n$.

Note that the problem (1.6) can be also be written as

$$\begin{aligned} &\min_{(t,d)\in\mathbb{R}\times\mathbb{R}^n} \left(t + \tfrac{1}{2}\|d\|_2^2\right) \\ &\text{subject to } \ \langle \nabla f_j(x), d\rangle \le t, \quad j = 1, 2, \ldots, m, \end{aligned}$$

which is a convex quadratic problem in which the inequality constraints are linear.

1.4 Line Searches for MOPs

In the realm of MOPs, line search techniques play a pivotal role in determining optimal step sizes along descent directions. These techniques, first reported by Lucambio [98], are crucial for vector-valued functions $F : \mathbb{R}^n \to \mathbb{R}^m$. The primary goal of line search methods is to find a step length that sufficiently reduces the objective function in the chosen direction.

1.4.1 Exact Line Search

We begin with the concept of exact line search, which forms the theoretical foundation for understanding more practical inexact methods.

Definition 1.4 (Exact Line Search) *Let $d \in \mathbb{R}^n$ be a descent direction for F at $x \in \mathbb{R}^n$. We say that $\alpha > 0$ is obtained through* exact line search *if $\mathcal{M}(x + \alpha d, d) = 0$.*

Here, $\mathcal{M}(x, d)$ represents the directional derivative of F at x in the direction d. The condition $\mathcal{M}(x + \alpha d, d) = 0$ implies that we have found a stationary point along the line $x + \alpha d$.

While exact line searches provide the theoretically optimal step size, they are often computationally expensive or impractical for complex objective functions. This limitation motivates the development of inexact line search methods, which aim to find a good approximate step size with less computational effort.

1.4.2 Wolfe-like Inexact Line Search

Wolfe-like inexact line search conditions for the vector case offer a practical alternative to exact line searches. These conditions ensure sufficient decrease in the objective function while also guaranteeing that we have moved far enough along the descent direction. They are classified into two categories:

(i) *Standard Wolfe line search*

(ii) *Strong Wolfe line search*

1.4.2.1 Standard Wolfe Condition

Let $d \in \mathbb{R}^n$ be a descent direction for $F : \mathbb{R}^n \to \mathbb{R}^m$ at $x \in \mathbb{R}^n$, and $0 < \rho < \sigma < 1$.

Definition 1.5 (Standard Wolfe Condition) *We say that $\alpha > 0$ satisfies the standard Wolfe condition if*

$$F(x + \alpha d) \preceq F(x) + \rho\alpha \mathcal{M}(x, d)e \quad \textit{and} \quad \mathcal{M}(x + \alpha d, d) \geq \sigma \mathcal{M}(x, d), \tag{1.9}$$

where e is the vector $(1, 1, \ldots, 1)^\top \in \mathbb{R}^m$.

The first inequality is known as the *sufficient decrease condition*. It ensures that the function value decreases by at least a fraction ρ of the initial rate of decrease. The second inequality is the *curvature condition*, which ensures that the slope at the new point is greater than a fraction σ of the initial slope. This prevents the step size from being too small.

1.4.2.2 Strong Wolfe Condition

Definition 1.6 (Strong Wolfe Condition) *We say that $\alpha > 0$ satisfies the strong Wolfe condition if*

$$F(x + \alpha d) \preceq F(x) + \rho\alpha \mathcal{M}(x, d)e \quad \textit{and} \quad |\mathcal{M}(x + \alpha d, d)| \leq \sigma |\mathcal{M}(x, d)|. \tag{1.10}$$

The key difference between the standard and strong Wolfe conditions lies in the second inequality. The strong Wolfe condition imposes a tighter constraint on the directional derivative at the new point, requiring its absolute value to be smaller than a fraction of the initial directional derivative. This additional restriction helps to prevent the algorithm from choosing points where the function is increasing too rapidly.

1.4.3 Armijo-like Inexact Line Search

The Armijo-like condition is another important inexact line search technique, which focuses on ensuring sufficient decrease in the objective function.

Definition 1.7 (Armijo condition) *For $d \in \mathbb{R}^n$, a descent direction for $F : \mathbb{R}^n \to \mathbb{R}^m$ at $x \in \mathbb{R}^n$, and $0 < \rho < 1$, we say that $\alpha > 0$ satisfies the Armijo condition if*

$$F(x + \alpha d) \preceq F(x) + \rho\alpha\mathcal{M}(x, d)e. \tag{1.11}$$

The Armijo condition ensures a sufficient decrease in the objective function value, proportional to the step size and the initial rate of descent. It is often used in backtracking line search algorithms, where the step size is iteratively reduced until the condition is satisfied.

1.5 Significance of Nonmonotone Line Search in MOPs

In the field of classical methods for addressing MOPs, the integration of nonmonotone line search techniques is a critical factor in enhancing the resilience and efficiency of optimization algorithms [111]. These techniques are particularly significant in overcoming the challenges posed by the complex nature of MOPs, which are characterized by intricate interactions among objectives and the presence of nonconvex, discontinuous, or multimodal features [111]. Unlike monotone methods that strictly adhere to a monotonic decrease in the objective function, nonmonotone line search techniques introduce adaptability into the optimization process [111]. This adaptability proves indispensable in navigating the complex solution landscape of MOPs, enabling algorithms to escape local optima and converge more effectively toward globally optimal solutions [111]. Furthermore, nonmonotone line search techniques offer a valuable means of handling noisy or uncertain objective functions, a common occurrence in real-world optimization scenarios [111]. A thorough exploration and refinement of nonmonotone line search techniques become imperative for advancing the capabilities of optimization algorithms in addressing the complexities of multiobjective decision-making [111]. Through an in-depth investigation of nonmonotone methodologies, this research seeks to contribute to the classical toolkit, providing a more comprehensive and adaptive approach for addressing the challenges inherent in MOPs [111].

1.6 Nonmonotone Line Search for MOPs

Nonmonotone line search techniques have shown to enhance the performance of descent methods, particularly in scalar optimization cases. These methods allow for occasional increases in the objective function values, potentially leading to faster convergence and the ability to escape local minima.

1.6.1 Monotone vs. Nonmonotone Approach

In the classical monotone line search technique, the choice of step size α_k ensures that $F(x^{k+1}) \prec F(x^k)$, meaning each iteration results in a lower objective function value. This approach, while intuitive, can sometimes lead to slow convergence, especially in the presence of narrow valleys or when the function has many local minima.

In contrast, nonmonotone line search allows for a certain amount of growth in the objective function values. This flexibility can lead to several advantages:

1. Faster convergence in many cases, as the algorithm can take larger steps.
2. Increased ability to escape local minima.
3. Better handling of noise or inexact function evaluations.

1.6.2 General Nonmonotone Line Search Condition

In nonmonotone line search, we choose $\alpha_k > 0$ satisfying

$$F(x^k + \alpha_k d^k) \preceq C^k + \rho\alpha_k \mathcal{M}(x^k, d^k)e, \tag{1.12}$$

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

The key difference from monotone methods lies in the reference value C^k. Instead of comparing the new function value to $F(x^k)$, we compare it to C^k, which is typically some measure of previous function values. This allows for occasional increases in the function value, as long as there is an overall decreasing trend.

1.6.3 Types of Nonmonotone Line Searches

Two primary types of nonmonotone line search methods have been proposed, differing in how they choose the reference value C^k.

1.6.3.1 Max-type Nonmonotone Line Search

Proposed by Grippo et al. [66], this technique is based on the maximum of recent objective function values from previous iterations.

Definition 1.8 (Max-type Nonmonotone Line Search) *The reference value C^k is typically chosen as*

$$C^k = \max_{0 \le j \le m(k)} F(x^{k-j}), \tag{1.13}$$

where $m(k)$ is a nonnegative integer, often called the "memory" of the method.

This approach allows the objective function to increase as long as it remains below the maximum value observed in the last $m(k)+1$ iterations. The memory parameter $m(k)$ controls the level of nonmonotonicity: larger values allow for more flexibility but may slow down overall convergence.

1.6.3.2 Average-type Nonmonotone Line Search

Introduced by Zhang and Hager [156], this method uses the average of recent objective function values instead of the maximum.

Definition 1.9 (Average-type Nonmonotone Line Search) *The reference value C^k is usually updated as:*

$$C^{k+1} = \frac{\sigma m_k}{m_{k+1}} C^k + \frac{1}{m_{k+1}} F(x^{k+1}).$$

Typically, $\sigma \in (0, 1)$ and m_k are updated recursively as $m_{k+1} = \sigma m_k + 1$.

This method provides a smoother reference value compared to the max-type approach, potentially leading to more stable behavior.

Both max-type and average-type nonmonotone line search methods offer more flexibility compared to monotone line searches, potentially leading to improved convergence properties in MOPs. The choice between these methods often depends on the specific characteristics of the problem at hand and the desired trade-off between exploration and exploitation in the search process.

Ideally, solving MOP (1.2) means obtaining the whole efficient set, that is, all the points which are efficient, and its corresponding nondominated set. However, for a majority of MOPs, it is not easy to obtain an exact description of the efficient set or nondomintaed set since those sets typically include an infinite number of points. To the best of our knowledge, only two exact general methods, namely, two interval branch-and-bound methods [44, 45] have been proposed in literature which obtain an enclosure of those sets up to a pre-specified precision. Specifically, they offer a list of boxes (multi-dimensional intervals) whose union contains the complete efficient set (and their images, the corresponding nondominated set) as a solution. However, they are time consuming. Furthermore, they have large memory requirements, so that only small instances can be solved with them. The common approach in literature is to approximate the nondominated set by a finite set of points. Multiobjective optimization methods try to get solutions that are as close as possible to the nondominated set and also uniformly distributed. Such methods will have good convergence and diversity. For this purpose, several performance metrics have been defined [133]. In the next section, we discuss about three main performance metrics.

1.7 Performance Metrics

It is necessary to ascertain how close is the obtained solution to the actual solution. For this propose, the literature contains various evaluation criteria [95]. In this thesis, we used three main performance measures such as GD+ (modified generational distance) [77], HV (hyper volume) [97] and IGD (inverted generational distance) [162].

Let x^* be an efficient point that is obtained by the proposed algorithm, and $F^* = (f_1^*, f_2^*, \ldots, f_{\mathcal{P}}^*)$ be an ideal point. Then, the modified generational distance (GD+) is calculated by the following formula

$$\text{GD+} = \sqrt{\sum_{i=1}^{\mathcal{P}} \left(\max\{f_i(x^*) - f_i^*, 0\}\right)^2}. \tag{1.14}$$

Apart from GD+, HV is calculated [97]. HV is the value obtained by finding the area enclosed by the generated nondominated solution points and a reference point, say $R(x_{(0)})$=$(f_1(x_{(0)})$,$f_2(x_{(0)}), \ldots,$ $f_r(x_{(0)}))$, which must be dominated by all $F(x^i)$, for $i \in \{1, 2, \ldots, \mathcal{J}\}$. Suppose that the solution set is sorted by increasing order with respect to f_1. Then, HV is calculated by

$$\text{HV} = \sum_{j=0}^{\mathcal{J}-1} \prod_{i=1}^{\mathcal{P}} \Big(f_i(x^{(0)}) - f_i(x_{j+1})\Big). \tag{1.15}$$

A higher HV value indicates better algorithm performance.

Another commonly used performance measure is IGD, which is calculated as follows

$$\mathrm{IGD}(G, G^*) = \frac{\sum_{x \in G^*} d(x, G)}{|G^*|}, \tag{1.16}$$

where the set G consists the approximation of the Pareto front and the set G^* contains uniformly distributed known nondominated points.

The following section discusses the existing methods to solve MOPs in the literature.

1.8 Classical Methods to Solve MOPs

1.8.1 Scalarization-Free Techniques

In the recent past, many researchers have started extending the classical methods of scalar optimization to solve unconstrained MOPs. These extensions include steepest descent [39,49], conditional gradient [8], Newton [50,149], quasi-Newton [123], conjugate gradient [63, 98], projected gradient [38, 56], proximal methods [15] and trust-region methods [18, 129]. A very short details of these papers are as follows.

Fliege and Svaiter [49] proposed the steepest descent methods for smooth unconstrained multiobjective optimization, aiming to find optimal solutions by iteratively moving in the direction of the steepest descent in the objective space. In their work, they introduced an adaptive step-size strategy that allows for efficient convergence to the Pareto front by controlling the trade-off between exploration and exploitation. The methods demonstrate good performance in terms of convergence and computational efficiency, making them suitable for various MOPs. However, challenges still exist, including handling nonconvex and nonsmooth objective functions, dealing with high-dimensional search spaces, and addressing the issue of local optima [50].

In the extension of Newton's method [50], multiple conflicting objectives need to be simultaneously optimized. The method leverages the second-order information about the objective functions to efficiently converge to local Pareto-optimal solutions. However, there are several challenges in applying Newton's method to multicriteria optimization, including the need to compute or approximate the Hessian matrix, handle nondifferentiable or discontinuous objective functions, and address the trade-off between convergence speed and solution quality.

The conjugate gradient method is a popular optimization algorithm that can be extended to tackle MOPs, where multiple conflicting objectives need to be optimized simultaneously. This method efficiently utilizes the gradient information to iteratively find optimal solutions along conjugate directions, thereby avoiding unnecessary computations. However, applying the conjugate gradient method to multiobjective optimization poses several challenges. These challenges include handling non-convex and non-differentiable objective functions, addressing the trade-off between convergence speed and solution quality, and dealing with the presence of constraints. A comprehensive discussion of these challenges and potential solutions can be found in the paper [98]. Lucambio Pérez and Prudente [98] generalize various conjugate gradient methods, such as the Fletcher-Reeves, conjugate descent, Dai-Yuan, Polak-Ribiére-Polyak, and Hestenes-Stiefel methods for continuously differentiable MOPs.

In [38], an extension of the projected gradient method has been introduced for MOPs. In [38], vector-valued functions are used directly instead of scalar-valued objectives.

In the extension of the proximal point method [15] for scalar-valued convex optimization, the subproblems consist of finding weakly efficient points for suitable regularizations; in this paper, an exact and an inexact version of subproblems are solved approximately.

The above discussed approaches are line search techniques. Trust-region methods are also developed to solve unconstrained MOPs in [18, 129]. In [129], the trust-region method is established for smooth and nonsmooth MOPs with convergence results. Trust-region methods developed in [18] are able to handle convex and nonconvex

MOPs.

1.8.2 Scalarization Techniques

Apart from the classical methods, another common approach towards constrained and unconstrained MOPs is to combine the objective functions and form single objective optimization subproblems which can be solved using conventional optimization techniques. This is known as scalarization [40, 41] or decomposition [157].

There are several scalarization techniques in the literature such as weighted sum [42, 50, 53], ϵ-constraint [46, 49, 70], normal constraint [106], normal boundary intersection [35, 63], physical programming [105], directed search domain [42, 146], etc. A brief discussion of the above scalarization techniques are as follows. Weighted sum method constructs the single objective as a weighted sum of the objective functions. Then, the problem can be solved repeatedly using different weights to approximate the nondominated set. If the Pareto surface is nonconvex, not all solutions on the trade-off surface can be found [22, 31, 135].

Another traditional method from the field of multiobjective optimization to generate the nondominated points is the ε-constraint method [46, 49]. The ε-constraint method works by choosing one objective function as the only objective and the remaining objective functions as constraints. Different elements of the nondominated set can be obtained by a systematic variation of the constraint bounds. The idea of the ε-constraint method is to iteratively increase the constraint bound by a pre-defined constant ε. The necessity to choose such a value represents the main drawback of this approach.

Normal constraint [106] and normal boundary intersection methods [35, 63] are the scalarization scheme for generating a set of evenly spaced efficient solutions. A significant drawback of these algorithms is that they can generate nonPareto optimal points.

Physical programming [105] and directed search domain [42] methods require the prior knowledge of the location of nondominated set in order to solve the MOPs.

In addition, for more detailed information on classical methods for solving MOPs, one can refer to [5,40,79–82,87,103,108,119,134,144,150,163] and the references therein.

1.9 Motivation

In the realm of computational science, the quest for efficient and robust algorithms to solve MOPs is a topic of paramount importance. These problems, characterized by multiple conflicting objectives, are ubiquitous in various fields such as engineering, economics, and logistics, to name a few. The complexity and diversity of these problems necessitate the development of innovative and effective algorithms.

The motivation behind this research work stems from the need to address the challenges associated with MOPs. The primary objective is to develop different algorithms that not only solve these problems but also offer superior performance measures. The performance of an algorithm is a critical factor that determines its applicability and effectiveness. Therefore, a comprehensive analysis of the performance measure forms an integral part of this research.

Empirical analysis plays a pivotal role in evaluating the efficiency of the proposed method. It involves the computation of relative efficiency and the generation of performance profiles. The relative efficiency provides a measure of the algorithm's performance in comparison to other algorithms. On the other hand, performance profiles offer a graphical representation of the algorithm's performance, providing a clear and concise overview of its efficiency.

Furthermore, this research incorporates a nonmonotone line search to enhance the performance of the method. Nonmonotone line search strategies have been proven to be effective in accelerating the convergence of algorithms, thereby improving their performance. The integration of such a strategy in the proposed method is expected to yield significant improvements in its efficiency.

In conclusion, the motivation behind this research is to contribute to the ongoing efforts in the field of multiobjective optimization by developing efficient algorithms and employing innovative strategies such as nonmonotone line search. Through rigorous empirical analysis and performance measure evaluation, this research aims to provide robust solutions to MOPs, thereby paving the way for advancements in various fields that rely on these problems. The potential impact and applicability of this research underscore its relevance and necessity in the current scientific landscape.

1.10 Contribution of the Thesis

In this thesis, we proposed new algorithms to solve MOPs. The main contribution of this thesis are as follows.

- New Algorithms for Multiobjective Optimization: We have developed new algorithms to solve MOPs, which could potentially offer more efficient or effective solutions.

- Comparative Analysis: We have conducted a comparative analysis of our proposed methods with standard existing methods, providing valuable insights into their relative strengths and weaknesses.

- Empirical Evaluation: We have carried out an empirical analysis to evaluate the efficiency of your proposed methods, contributing to the body of knowledge on the performance of these algorithms.

- Application of Dolan and Moré's Methodology: We have applied the methodology developed by Dolan and Moré [37] for generating performance profiles, demonstrating its utility in evaluating the performance of optimization algorithms.

- Use of Performance Matrices: We have utilized two commonly used performance matrices—inverted generational distance and hypervolume indicators for empirical comparison, providing a robust assessment of algorithm performance.

- We apply the proposed algorithm in Chapter 3 and Chapter 6 to solve some real-life optimal control problems.

1.11 Outline of the Thesis

The thesis is organized into seven chapters, beginning with an introduction and concluding with a chapter that outlines potential future research directions. In the introductory chapter, a comprehensive literature review is presented, offering insights into existing multiobjective optimization methods, emphasizing the significance of nonmonotone techniques, and identifying gaps in the current literature.

Chapter 2 introduces the "A Nonmonotone Polak-Ribière-Polyak Conjugate Gradient Method for Multiobjective Optimization Problems." This section details the formulation of the nonmonotone variant, its asymptotic convergence properties, and the establishment of Pareto critical points. Empirical validation includes comparisons with Hager-Zhang and Liu-Storey conjugate gradient methods, utilizing performance metrics for assessment. This chapter has been submitted to the *Journal of Computational and Applied Mathematics*.

Chapter 3 introduces a "Projection-type Hybrid Conjugate Gradient Method for Multiobjective Optimization Problems with Application to an SIR Model." The chapter presents novel methods combining the HS and DY approaches under strong Wolfe line search conditions. Additionally, two new nonmonotone hybrids conjugate gradient methods are introduced in which we take hybrid of PRP with HS and DY, these methods are analyzed using an average-type nonmonotone line search to enhance flexibility and efficiency in nonconvex MOPs. The global convergence of these methods is studied without convexity assumptions. Empirical analysis and application to an SIR optimal control model demonstrate the method's efficiency and practical applicability. This chapter has been accepted for publication in the *Journal of Nonlinear and Convex Analysis*.

Chapter 4 introduces the "Nonmonotone Wolfe-type Quasi-Newton Methods for Multiobjective Optimization Problems." This section integrates Wolfe line searches into quasi-Newton algorithms, exploring nonmonotone variants with the BFGS quasi-Newton method. Global convergence is established under reasonable assumptions, and empirical analysis compares the proposed method with monotone quasi-Newton algorithms, emphasizing efficiency. This chpater has been revised and submitted to *Optimization*.

Chapter 5 introduces the "A Nonmonotone Conditional Gradient Method for Multiobjective Optimization Problems." The chapter delves into the formulation of the proposed methods, its asymptotic convergence properties, and the establishment of Pareto critical points. The effectiveness of the approach is validated through empirical comparisons with Hager-Zhang conjugate gradient, steepest descent, and monotone conditional gradient methods. Performance metrics such as inverted generational distance and hypervolume indicators are employed for evaluation. This chpater is pubished in *Soft Computing* (2024).

Chapter 6 presents the "Augmented Lagrangian Cone Method for Multiobjective Optimization Problems." The method is introduced, addressing the computation of Pareto optimal sets without prior information about the Pareto surface or convexity assumptions. A nonmonotone line search technique is integrated within the steepest descent method for solving the transformed subproblems, enhancing the method's convergence properties. Convergence analysis and global Pareto optimality are discussed. Empirical validation includes solving standard test problems and application to an unemployment optimal control model. This chapter is published in *Optimization and Engineering* (2022).

The concluding chapter, Chapter 7, summarizes the contributions of the thesis, discusses implications of the proposed methods, acknowledges limitations, and outlines potential avenues for future research.

****