Novel Evolutionary Approaches for Multi-Modal Multi-Objective Problems

DISSERTATION

zur Erlangung des akademischen Grades

Doktoringenieur (Dr.-Ing.)

angenommen durch die Fakultät für Informatik
der Otto-von-Guericke-Universität Magdeburg

von M.Sc. Mahrokh Javadi

geb. am 04.04.1987 in Abhar

Gutachterinnen/Gutachter
Prof. Dr.-Ing. habil. Sanaz Mostaghim
Prof. Dr. Carlos Coello Coello
Prof. Dr.-Ing. Yaochu Jin

Eingereicht am: 26.02.2023

Verteidigt am: 15.09.2023

To my parents

E h r e n e r k l ä r u n g

Ich versichere hiermit, dass ich die vorliegende Arbeit ohne unzulässige Hilfe Dritter und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe; verwendete fremde und eigene Quellen sind als solche kenntlich gemacht. Insbesondere habe ich nicht die Hilfe eines kommerziellen Promotionsberaters in Anspruch genommen. Dritte haben von mir weder unmittelbar noch mittelbar geldwerte Leistungen für Arbeiten erhalten, die im Zusammenhang mit dem Inhalt der vorgelegten Dissertation stehen.

Ich habe insbesondere nicht wissentlich:

- Ergebnisse erfunden oder widersprüchliche Ergebnisse verschwiegen,
- statistische Verfahren absichtlich missbraucht, um Daten in ungerechtfertigter Weise zu interpretieren,
- fremde Ergebnisse oder Veröffentlichungen plagiiert,
- fremde Forschungsergebnisse verzerrt wiedergegeben.

Mir ist bekannt, dass Verstöße gegen das Urheberrecht Unterlassungs- und Schadensersatzansprüche des Urhebers sowie eine strafrechtliche Ahndung durch die Strafverfolgungsbehörden begründen kann. Die Arbeit wurde bisher weder im Inland noch im Ausland in gleicher oder ähnlicher Form als Dissertation eingereicht und ist als Ganzes auch noch nicht veröffentlicht.

Magdeburg, den 12.06.2023

Mahrokh Javadi

Abstract

Optimization problems with conflicting objectives occur in various domains and can be tackled through simultaneous optimization of multiple objectives. Metaheuristic techniques like evolutionary algorithms are used to search for optimal decision variables. Multi-modal multi-objective optimization problems have gained recent interest in the field, although they remain relatively unexplored and require further investigation. The motivation for studying these problems arises from decision-makers' preferences that cannot be mathematically expressed or incorporated into existing frameworks. Providing diverse solutions enables informed decision-making, and studying these problems offers alternative solutions when implementation challenges arise.

In the existing literature, numerous studies focus on capturing and preserving diverse solutions in the search space. Nevertheless, there is a need for further research to develop methods that can more accurately estimate solution density in the neighborhood of each individual solution. Additionally, it is crucial to develop approaches that effectively preserve diverse solutions in the search space and prevent getting trapped in local optima. By tackling these challenges, the performance and efficiency of multi-modal optimization algorithms can be significantly improved, leading to a more effective exploration of the search space for these specific problems.

This thesis contributes to multi-modal multi-objective optimization in two key ways. Firstly, it proposes approaches to handle two types of problems: those with multiple global optimal solution sets and those with both local and global Pareto optimal solution sets. The developed algorithms effectively address these problems. Secondly, novel algorithms are introduced to overcome the limitations of the crowding distance method, ensuring an accurate representation of solution diversity in the search space. Additionally, a classification scheme for multi-modal multi-objective optimization algorithms based on their selection mechanism is presented.

This thesis includes a thorough experimental evaluation of proposed and existing methods, analyzing their advantages, disadvantages, and performance. The results demonstrate that these approaches are competitive and frequently outperform the state-of-the-art methods in the field.

Zusammenfassung

Mutlikriterielle Optimierungsprobleme treten in verschiedenen Bereichen auf und können durch die gleichzeitige Optimierung mehrerer Ziele gelöst werden. Metaheuristiken wie evolutionäre Algorithmen sind Werkzeuge, um optimale Lösungen unter Berücksichtigung mehrerer Kriterien zu suchen. In den letzten Jahren ist das Interesse an multimodalen, multikriteriellen Optimierungsproblemen gewachsen. Allerdings gibt es nicht viele Forschungsarbeiten in diesem Bereich. Die Motivation, diese Probleme weiter zu untersuchen, sind Präferenzen von Entscheidungsträgern, welche in den existierenden Frameworks bisher nicht berücksichtigt werden können. Das Berechnen von möglichst diversen Lösungen ermöglicht es besser informierte Entscheidungen zu treffen und Alternativlösungen zu wählen, sollte bei der Umsetzung der eigentlich ausgewählten Lösung ein Problem auftreten.

Die bisherige Literatur konzentriert sich darauf, die multikriteriellen Lösungen gut im Lösungsraum zu verteilen. Dabei wird oft die Verteilung der Lösungen im Suchraum vernachlässigt, was dazu führen kann, dass nur unimodale Lösungen gefunden werden. Eine gute Verteilung im Suchraum ist essenziell für die Performance und Effizienz von multimodalen Optimierungsalgorithmen.

Diese Dissertation trägt in zwei wesentlichen Aspekten zur mutlimodalen, multikriteriellen Optimierung bei. Erstens werden Ansätze zur Bewältigung von zwei Problemklassen vorgeschlagen: Probleme mit mehreren global optimalen Lösungen und Probleme mit sowohl lokalen als auch globalen Pareto-optimalen Lösungen. Die entwickelten Algorithmen adressieren diese beiden Problemklassen effektiv. Zweitens werden neue Algorithmen vorgestellt, um die Limitation der üblich verwendeten Crowding-Distance Metrik zu adressieren und eine bessere Verteilung im Suchraum sicherzustellen. Außerdem wird eine Klassifikation für mutlimodale, multikriterielle Optimierungsalgorithmen basierend auf ihrem Selektionsmechanismus präsentiert.

Diese Dissertation umfasst eine gründliche experimentelle Evaluation der bereits bestehenden und hier neu präsentierten Methoden, in welcher Vorteile, Nachteile und die Performance analysiert werden. Die Ergebnisse zeigen, dass die hier vorgestellten Ansätze im Vergleich zu den bisherigen Methoden wettbewerbsfähig sind und sie häufig übertreffen.

Contents

Nomenclature

Ω	An optimization problem's Search space
IGD^+	Inverted Generational Distance Plus in the objective space
M	An optimization problem's objective space
m	Number of objective functions of multi-objective optimzation problems
n	Number of decision variables of multi-objective optmization problems
PF	Pareto-front for multi-objective optmization problems
PS	Pareto-set for mutli-objective optmization problems
CD	Crowding Distance
DM	Decision Maker
DN-NSGA-II	Decision space-based Niching Non-dominated Sorting Genetic Algorithm II
EA	Evolutionary Algorithm
EMO	Evolutionary Multi-objective
HAD	Harmonic Average Distance
IGD	Inverted Generational Distance indicator in the objective space
IGDx	Inverted Gerneational Distance indicator in the decision space
MMOEA	Multimodal Multi-Objective Optimization Evolutionary Algorithm
MMOP	Multimodal Multi-Objective Optimization Problem

MOEA Multi-Objective Optimization Evolutionary Algorithm

MOEA/D Multiobjective Evolutionary Algorithm based on Decomposition

MOP Multi-objective Optimization Problem

NBM Neighborhood-Based polynomial Mutation

NBM Neighborhood-based mutation

NSGA-II Non-dominated Sorting Genetic Algorithm II

NSGA-II-MDCD NSGA-II-manhatan-distance-crowding-distance

NSGA-II-NBM NSGA-II-Neigborhood-based Mutation

NSGA-II-WSCD NSGA-II-Weighted-Sum-Crowding-Distance

NSGA-II-WSCD-NBM NSGA-II-Weighted-Sum-Crowding-Distance-Neigborhood-based Mutation

NSGA-III Non-dominated Sorting Genetic Algorithm III

NxEMMO Neighborhood-based Evolutionary Multimodal Multi-objective Optimization

PF Pareto-Front

PS Pareto-Set

PSP Pareto-Set Proximity

SBX Simulated Binary Cross-over

SOP Single-objective Optimization Problem

WSCD Weighted Sum Crowding Distance

Chapter 1

Introduction

Many optimization problems feature multi-modal properties, which means that there are multiple solutions of similar or slightly inferior quality according to the objective function values. Multi-modality has been extensively studied within the context of single-objective optimization problems. Nevertheless, the field of multi-modal, multi-objective optimization is still relatively unexplored and requires a great deal of attention. The motivation for the study of these types of problems is attributed to preferences of decision-makers that cannot be mathematically expressed and, as a result, cannot be incorporated into multi-modal multi-objective optimization problems (MMOPs). Therefore, providing diverse ranges of nearly equivalent solutions will facilitate decision-makers' capacity to make informed decisions. In addition, studying these problems offers the advantage of providing an equivalent alternative in the event of difficulty implementing a solution [1]. Classical multi-objective optimization evolutionary algorithms (MOEAs) struggle to solve these problems since they neglect to preserve diverse solutions in the search space while focusing on preserving diversity in the objective space; this means that multiple solutions with the same or similar objective values are not preserved by the search procedure. In this dissertation, we develop methodologies and approaches to tackle these types of problems.

1.1 Motivation

The majority of real-world problems involve the simultaneous optimization of one or more incompatible and sometimes conflicting objectives, which are referred to as optimization problems. Based on the number of objectives, optimization problems are classified as either single-objective, multi-objective (two to three objectives), or many-objective (at least four objectives). We see problems such as these every day, such as those faced by engineers optimizing their design parameters or manufacturers maximizing their production efficiency. In general, the optimization process aims to find and maintain solutions until no better alternative can be found. In optimization problems involving at least two competing objectives, improving one objective results in worsening the others; hence, it is necessary to identify tradeoff solutions that satisfy all the competing objectives concurrently. They are known as multi-objective optimization problems (MOPs). A multi-objective optimization approach can be applied to a wide variety of problems in real life, including feature selection, where the two competing objectives minimize the number of features while maximizing the quality of the features, or the optimization of delivery and inventory costs.

In recent decades, direct or gradient-based methods have been introduced to solve optimization problems using mathematical principles. The design and implementation of classical and exact optimization algorithms typically involve techniques such as dynamic programming, branch-and-bound, and backtracking. Nevertheless, in some multi-objective optimization problems, complexity factors such as a large search space, uncertainty noise, multi-modality, disjoint Pareto curves, and so on may prevent classical methods from being applied [2]. In addition, classical approximation methods such as greedy algorithms involve several assumptions that are difficult to verify in many situations.

Over recent decades, evolutionary algorithms (EAs) have become an increasingly popular tool for searching, optimizing, and providing solutions to complex problems [3] as an alternative to classical methods. These algorithms are population-based, meta-heuristic algorithms that employ aspects of biological evolution, such as reproduction, mutation, recombination, and selection, in their development. Since their introduction in the 1960s, EAs have become increasingly popular for solving various optimization problems, primarily due to the population-based nature of EAs that can produce multiple elements of the Pareto-optimal set in a single run as well as deal with large search spaces [4]. Multi-objective evolutionary algorithms (MOEAs) aimed at optimizing multi-objective problems emerged in the 1990s, and especially since 2001, there has been exponential growth in the number of approaches and methods aimed at addressing these types of problems. These algorithms seek to gradually achieve a set of Pareto-optimal solutions evenly distributed across the Pareto front. Due to the fact that there is no single-best solution in multi-objective evolutionary algorithms, their selection scheme differs from the one used in single-objective optimizations.

Numerous real-world problems (e.g., multi-objective knapsack optimization [5], path-finding optimization [6, 7, 8, 7], functional brain imaging problems [9], and rocket engine design problems [10]) possess multi-modal characteristics, either as a result of the nature of the underlying function or as a consequence of constraints or due to changes in the environment imposing the island of infeasibility on the problem, or both [11], which makes them difficult to solve using classical optimization techniques. As a result of multi-modality in multi-objective optimization problems, there can be multiple local/global optimal solutions whose objective functions are similar or somewhat inferior.

The two classical challenges emphasize the difficulty resulting from these types of problems. A first challenge is the preservation of multiple diverse optimal solutions in the decision space, with a similar quality in the objective space. A second challenge is avoiding the trap of local optima due to the presence of multiple basins of attraction—a fact that multi-modality views as an obstacle to achieving global optima [12]. To overcome the first challenge, methods and approaches must be developed that identify and retain all optimal Pareto-sets of solutions, even if they correspond to similar sets in the objective space. As such, preserving diverse solutions in the decision space is crucial when dealing with MMOPs. As a consequence, classical MOEAs are often ineffective when applied to MMOPs because they lack mechanisms to keep solutions whose fitness values are relatively similar. Addressing the second challenge is implicitly considered in all MOEAs, in which these algorithms seek to increase the diversity in the search space, thus pursuing further exploration of the search space in undiscovered areas.

The rationale behind investigating these problems stems from decision-makers' preferences. This exploration can prove advantageous if the optimizer can present various sets of optimal solutions that possess equivalent quality based on their objective function values. Moreover, it can be advantageous for the optimizer to provide local optimal solutions that may have lower quality but are still acceptable alternatives for decision-makers. The search for multiple optimal solutions and the preservation of these obtained solutions are essential for various reasons. One reason is that the global optimal solution provided may not be feasible or attainable in practical terms, especially when limited to a discrete set of available parts or shapes. Additionally, decision-makers may reject specific solutions for reasons not accounted for in the model, such as subjective preferences or "soft constraints" that cannot be expressed mathematically but are crucial for decision-making [11]. Additionally, maintaining these different optimal solutions increases the reliability of the decision-making process by eliminating uncertainties that may arise during the actual implementation process.

A simple real-world example is apparent in path-finding problems, in which a traveler wishes to arrive at the destination with the fewest number of intersections and in the shortest amount of time. Optimizers may find two different optimal routes that offer the same number of intersections and time taken to reach a destination. However, the optimal route selected by the user becomes inaccessible due to sudden environmental changes (e.g., heavy snowfall). By identifying an alternative route, the optimizer allows the traveler to take the alternative route in place of the intended route in order to reach a destination without any further delay.

In the scope of this thesis, MMOPs is divided into two types. The first type of problem consists of preserving the global optimal set of solutions (i.e., Type I MMOPs), while the second type involves multiple locally optimal sets with slightly inferior yet acceptable quality to the decision-maker (i.e., Type II MMOPs). The purpose of this thesis is to address both types of problems by identifying, developing, and evaluating techniques and approaches that are crucial to finding multi-modal solutions to multi-objective optimization problems.

Over the past few decades, multi-modal single-objective optimization problems have received a significant amount of research attention. The area of multi-objective, multi-modal optimization is a somewhat new field that was first introduced in 2005. The field of MMOPs did not receive explicit attention from

2007 to 2012, although some activities have been carried out to obtain diverse solutions within the search space. To the best of the author's knowledge, the term "MMOP" reemerged in 2016, drawing researchers' attention and resulting in the development of several methods. This thesis discusses the proposed techniques from the last few years in light of the theoretical challenges we identify and our own proposed optimization approaches. This thesis also proposes a new classification of state-of-the-art multi-modal multi-objective optimization evolutionary algorithms (MMOEAs) as a contribution to the field.

1.2 Research goals and objectives

The purpose of this dissertation is to propose and enhance various methods of addressing the challenges of MMOPs, particularly regarding developing strategies for preserving diverse solutions within the search space. The following objectives are defined:

In addition to the objectives, there is also a set of questions to address to better quantify/qualify each objective's achievement.

Objective 1: Develop methods and approaches to address Type I MMOPs

The first part of this thesis proposes strategies to solve MMOPs with multiple global global Pareto-sets (global PSs). For this reason, current state-of-the-art methodologies and search strategies are studied, modified and applied to Type I MMOPs. To achieve this goal, the following research questions are addressed:

- RQ 1: Which methodologies can be used to enhance the distribution of solutions in the decision space?
- RQ 1.1: Which techniques can be used to drive the search toward areas in the search space?

One of the primary research questions regards how to develop methods to guide the population to explore the different search space areas and find global optima by preventing premature convergence and becoming stuck in local optimum regions. The proposed approaches are discussed in Section 4.1.

- RQ 1.2: How can the optimizer preserve the multiple distinct optimal solutions in the search space which are relatively close in the objective space?

A substantial part of this thesis focuses on the challenge that arises because traditional selection operators of MMOEAs tend to remove useful diverse solutions in search space located in crowded regions of objective space to enhance the distribution of optimal solutions over the approximated Pareto-front (PF). We have developed different selection operators to cope with this challenge. Using these approaches, it is possible to make a better assessment of the similarity between the obtained solutions as well as of the survival of solutions located in distinct regions of the search space while they are close in the objective space. The main discussion of the proposed approaches is presented in Sections 4.2 and 5.2.1.

- RQ 2: How to make a trade-off between the diversity of solutions in the search and objective spaces?

Achieving a sufficient diversity of solutions in both the decision and objective spaces presents a challenge in MMOPs. This challenge arises due to the inherent tradeoff between improving the distribution of solutions in the search space and the distribution in the objective space. To address this issue, we have developed a method that measures density in both spaces, allowing us to strike a balance between the divergence of solutions in both domains. The detailed explanation of our proposed approach can be found in Section 4.2.1.

Objective 2: Develop methods and algorithms to solve Type II MMOPs.

To achieve this goal, we developed a method that preserves a local Pareto-set (local PS) while evolving toward the respective global PS.

- RQ 3: Which methods can be used to preserve the non-Pareto-optimal solutions that are possible to implement by the decision-maker?

Objective 3: Classification of the selection mechanisms in Pareto-dominance-based MMOEAs

Given the absence of a structured categorization of algorithms that focuses on selection operators, which play a crucial role in MMOEAs, we examined various limitations of crowding distance techniques. These limitations encompass the imprecise estimation of solution density within the search space's neighborhood region. To address this gap, we introduce a novel classification of MMOEAs in this thesis.

- RQ 4: How to efficiently classify algorithms based on the density estimation of the solutions in their neighborhood area to preserve diversity of the solutions?

In Pareto-based multi-modal, multi-objective optimization algorithms, we present a new classification for selection operations. This classification considers the inclusion of nearby solutions from the current front (referred to as $Front_i$) and the solutions from previous fronts ($Front_1$ to $Front_{i-1}$) when estimating the density of the neighborhood area of solutions. To overcome the limitations of existing crowding methods, we propose two classifications: inter-front and intra-front selection operators.

Objective 4: Evaluation of the proposed algorithms

- RQ 5: How do modern MOEAs such as niching non-dominated sorting genetic algorithm II (NSGA-II), NSGA-III, and MOPSO perform when dealing with MMOPs?

- RQ 6: How do the proposed algorithms compare in terms of distribution of obtained optimal solutions both in decision and objective spaces?

In this thesis, the proposed strategies are empirically evaluated by conducting experiments on various state-of-the-art algorithms found in the existing literature, as well as comparing them against each other. To assess the performance of these algorithms, a set of diverse test functions, featuring varying levels of complexity and decision variables, are employed. Furthermore, the algorithms are comprehensively evaluated, and their strengths and weaknesses are analyzed based on specific criteria. Performance comparisons are made by examining metrics such as diversity and convergence in both the search space and objective space. Additionally, the influence of population size on the diversity and convergence speed of the obtained solutions for the Pareto-set (PS) and PF is also investigated.

1.3 Thesis Outline

This thesis develops methods and approaches as solutions to the previously discussed challenges of dealing with multi-modality in MOPs. The thesis consists of six chapters that cover all relevant challenges associated with solving optimization problems of this type. The organization of this thesis is as follows.

Chapter 2 (Multi-Modal Multi-Objective Optimization: Detailed Study of Properties): This chapter outlines the major principles that guide the remainder of the dissertation. These include concepts such as multi-objective optimization, Pareto optimality, metaheuristic methods, and multi-objective evolutionary algorithms. Additionally, this chapter discusses the challenges posed by multi-modality when solving multi-objective optimization problems and outlines the different types of MMOPs. Section 2.6 defines the notion of niching techniques to solve these problems. The most common multi-modal and multi-objective benchmark problems are described in Section 2.7, as are characteristics of these problems. This chapter ends with a discussion of the most common performance indicators used to assess an algorithm's performance.

Chapter 3 (Overview of State-of-the-Art Algorithms): This chapter provides a brief overview of the techniques employed to tackle single-objective multi-modal problems. Furthermore, it conducts a comprehensive review of the existing literature on multi-modal, multi-objective optimization, categorizing and comparing the various methods utilized in this field. Additionally, the last section of this chapter presents an explanation of the algorithm setup and configurations, establishing the foundation for subsequent analyses.

Chapter 4 (Proposed Techniques for Multi-Modal Multi-Objective Optimization): Our purpose in this chapter is to present new methods and approaches to solving MMOPs of type I and type II that have been developed over the course of this thesis and in the author's previous publications. Additionally, we propose approaches for exploring the search space in more depth. Throughout this chapter, we discuss RQ 1, RQ 2, and RQ 3. We assess the performance of each method described by comparing it to the most prominent state-of-the-art algorithms across multiple test functions. These test functions encompass various levels of complexity, including factors such as the number of decision variables, shape, and number of Pareto sets. The analysis carried out in this study addresses research questions 5 and 6.

Chapter 5 (Classification of the Pareto-Dominance Based MMOEAs into Inter and Intra-Front Selection Operations): In this chapter, we present two novel algorithms aimed at overcoming the limitations of the crowding distance method in effectively representing the diversity of solutions within the search space. MMOEAs are classified based on their selection mechanism, which covers RQ 4 in further depth. As each algorithm is presented, the experimental results are examined and evaluated.

Chapter 6 (Conclusions and perspectives): The present manuscript is concluded with some concluding remarks, in addition to some detailed perspectives on the work described, together with an outlook on future research topics in the area.

Chapter 2

Multi-Modal Multi-Objective Optimization

In this chapter, we provide an overview of multi-objective optimization, evolutionary algorithms, and multi-modal multi-objective optimization. The fundamental principles of multi-objective optimization are discussed in Section 2.1, followed by an exploration of the concepts of multi-modality in the context of single and multi-objective optimization in Section 2.2. Afterward, we provide a description of meta-heuristic approaches in Section 2.3, followed by an explanation of evolutionary techniques for solving problems associated with multi-objective optimization in Section 2.4. Niching methods, originally developed for addressing multi-modal single-objective optimization problems, are introduced in Section 2.6. Additionally, we provide a brief description of several multi-modal multi-objective function benchmarks in Section 2.7. The performance metrics utilized to evaluate algorithm performance are outlined in Section 2.8. Finally, in Section 2.9, we conclude this chapter with a succinct summary of the key concepts discussed.

2.1 Multi-objective 0ptimization

As mentioned, the purpose of MOP is to optimize more than one criterion simultaneously, unlike single-objective optimization problems (SOPs) which aim to maximize/minimize one objective. Based on these objectives, pairwise relationships can be either independent, complementary, or conflicting. In the first two relationships, changes in one independent objective do not affect the other independent objectives. However, in complementary objectives, improvements or deterioration in one objective have similar impacts on the other objective. For complete independence objectives, such problems can be broken down into several single-objective optimizations, while for complete harmony objectives, the problems can be reduced to a single-objective optimization. The most complicated scenarios involve conflicting objectives, and improving one may prevent progress on another. It is therefore necessary to find an optimal solution set that demonstrates different trade-offs between the opposing objectives and functions.

A general definition of MOP is as follows:

$$\begin{aligned} \underset{x \in S}{\text{minimize}} \quad & \vec{f}(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \cdots, f_m(\vec{x})) \\ \text{subject to} \quad & h_i(\vec{x}) = 0, \quad i = 1, 2, \cdots, H \\ & g_j(\vec{x}) \leq 0, \quad j = 1, 2, \cdots, G \end{aligned} \tag{2.1}$$

Without loss of generality, we consider minimization problems. In such type of MOP, $S \subseteq \mathbb{R}^n$ denotes the feasible region in the search space, also called decision space, defined as follows:

$$S = \{\vec{x} \in \mathbb{R}^n \mid h_i(\vec{x}) = 0 \quad \& \quad g_j(\vec{x}) \leq 0\} \tag{2.2}$$

where the vector $\vec{x} \in S$ represents the design (or decision) variables, and n represents the number of the decision variables. $h_i(\vec{x}) : \mathbb{R}^n \to \mathbb{R}, i = 1, ..., H$ and $g_j(\vec{x}) : \mathbb{R}^n \to \mathbb{R}, j = 1, ..., G$ are equality and inequality constrains, respectively.

The objective functions represented by the vector $\vec{f}(\vec{x}) : S \to \mathbb{R}^m$

$$f_i(\vec{x}) : S \to \mathbb{R} \quad \forall i \in \{1, \cdots, m\} \tag{2.3}$$

In the case of $m = 1$, this problem belongs to the class of single-objective optimizations. These problems are regarded as multi-objective optimization for $2 \leq m \leq 3$, and as many-objective optimization for $m \geq 4$.

An optimization problem that does not contain any constraints is considered an unconstrained problem. A minimization problem can be converted into a maximization problem by multiplying the objective functions by -1.

Throughout this dissertation, we use the term MOP to describe problems where the feasibility region of each problem is defined as a box-constraint (S_{Box}), and the domain for each decision variable is defined as follows:

$$S_{Box} = \{\vec{x} \in \mathbb{R}^n \mid lb_i \leq x_i \leq ub_i\} \tag{2.4}$$

where a lower and upper bound for each of the decision variables is represented by lb_i and ub_i respectively, $\forall i \in \{1, \cdots, n\}$.

Definition 2.1 (Dominance relationship) The domination relation is employed to compare two solution vectors $\vec{x}$, $\vec{y} \in S$. $\vec{x}$ is said to be dominated by $\vec{y}$ (denoted by $\vec{y} \prec \vec{x}$) if and only if $\forall j \in \{1, \cdots, m\}$, $f_j(\vec{y}) \leq f_j(\vec{x})$, and $\exists k \in \{1, \cdots, m\}$, $f_k(\vec{y}) < f_k(\vec{x})$.

Definition 2.2 (Pareto-optimality) A Pareto-optimal set is the set of all solutions $\vec{x} \in S$ where there is no other solution in the entire feasible search space that dominates any solution within this set.

$$PS := \{\forall \vec{x} \in S \Leftrightarrow \quad \nexists \vec{y} \in S \mid \vec{y} \prec \vec{x}\} \tag{2.5}$$

Within a search space, the collection of optimal solutions is known as the Pareto-set (PS). These solutions exhibit a non-dominance relationship, meaning that none of them are superior to others or any other feasible solutions in the search space. As a result, it is not feasible to establish a ranking among them, as they inherently involve a trade-off between different objectives. In the objective space, the Pareto-front (PF) represents the set of solutions that corresponds to the Pareto set.

Definition 2.3 (Non-dominated Set) A non-dominated set of solutions refers to all the solutions in a given set of solutions that are not dominated by any other solution.

Definition 2.4 (Local Pareto-optimal-set (Local PS)) In the search space S, for any arbitrary solution $\vec{x}$ in the solution set L_{PS} (where L_{PS} is the set of the solutions in the neighborhood of the solution $\vec{x}$), there is not any other solution y (where $\|\vec{y} - \vec{x}\|_\infty \leq \epsilon$, ϵ represents a very small positive value) that dominates the solution $\vec{x}$. This set of solutions is referred to as the Local Pareto Set (or Local PS).

The local PF (or Local PF) represents the corresponding solutions in the objective space.

Definition 2.5 (Global Pareto-optimal-set (Global PS))

The set of all solutions that are Pareto-optimal is called the global Pareto-set (also known as Pareto-set).

$$Global\text{–}PS := \{\vec{x} \mid \vec{x} \in Pareto\text{–}optimal\text{–}Set\} \tag{2.6}$$

Global Pareto-front (also known as Pareto-front) represents the objective space image of these solutions.

$$PF := \left\{\vec{f}(\vec{x}) \mid \vec{x} \in Pareto\text{–}optimal\text{–}Set\right\} \tag{2.7}$$

An example of such problems is depicted in Fig. 2.1.

2.2 Multi-modal Optimization

The concept of "multi-modal optimization" refers to the simultaneous search for multiple solutions to a complex objective function [11]. In general, multi-modal optimization involves applying specific optimization methods to multi-modal functions in an attempt to capture as many optimal solutions as possible, including basin attractions, as described in [13, 11]. The implementation of MMOEAs cannot succeed unless the problems being addressed are multi-modal in nature. It has been widely assumed, however, that most problems in Applications that cannot be effectively solved by classical optimization algorithms (e.g., gradients, quasi-Newton methods) and are therefore addressed by metaheuristics have multi-modal characteristics [14].

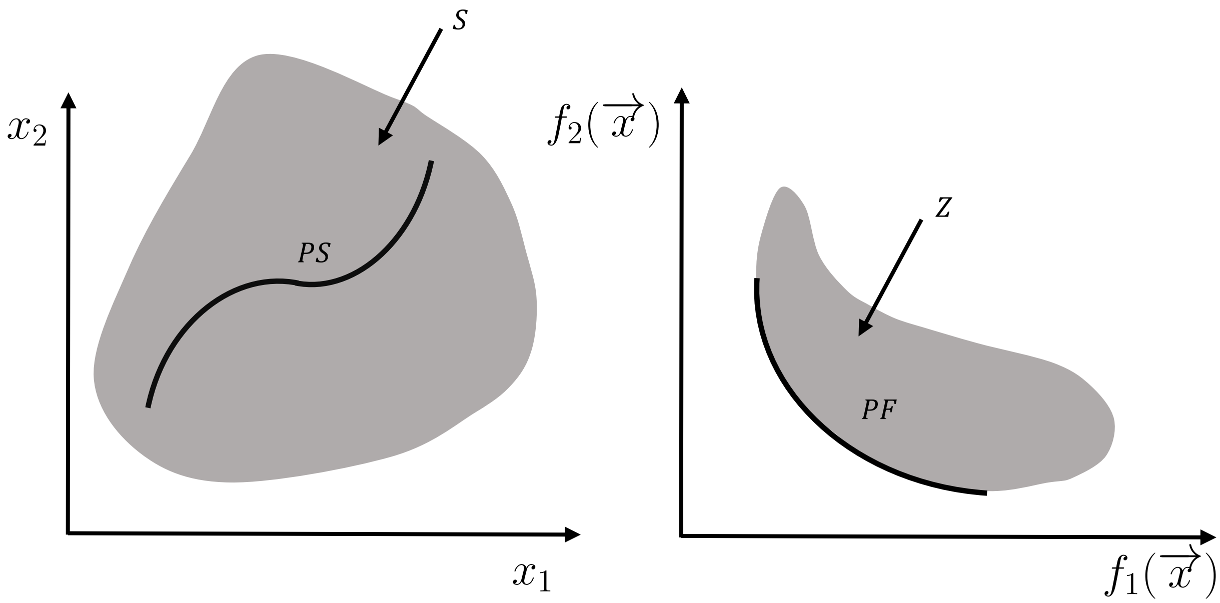


Figure 2.1: Visual representation of the PS in the decision space (left figure), which is mapped to the PF in the objective space (right figure). In the search space, the feasible area is highlighted in blue, which corresponds to the highlighted area in the objective space. In the decision and objective spaces, the hypothesized PS and PF are marked in red.

When dealing with single-objective optimization problems, multi-modal functions refer to problems that are characterized by the existence of multiple global and/or local optima [15]. Similarly, MOPs' definition of multi-modality specifies the existence of two or more global Pareto-optimal solutions for a given point on the PF or at least one local Pareto-optimal solution [16, 17].

2.2.1 Single-objective Multi-modal Optimization

Assuming that $\vec{v}$ represents the local minima $f^*_{(1)}, f^*_{(2)}, \cdots, f^*_{(v)}$ for the objective function $f : S \rightarrow \mathbb{R}$. If f is ordered as $f^*_{(1)} < f^*_{(2)} < \cdots < f^*_{(p)} < TH < \cdots < f^*_{(v)}$ then set $\bigcup_{i=1}^{p} \{X^*_i\}$ is a multimodal optimization solution for a single objective optimization problem where $X^*_i = \left\{\vec{x} \in S \mid f(\vec{x}) = f^*_{(i)}\right\}$

where TH represents the quality threshold beyond which local optimization solutions of a lower quality are excluded from consideration. Assuming that $TH = f^*_{(1)}$, the decision-maker is only concerned with finding the global optimal solution and not the local ones. When $TH = \infty$ is considered, it means that all local minima of the problem are accepted by the decision-maker.

2.2.2 Multi-objective Optimization

MMOP refers to problems in which at least k (≥ 2) decision vectors ($C = \{\vec{x_1}, \vec{x_2}, \cdots, \vec{x_k}\}$) result in a similar objective vector value, i.e. $\forall(\vec{x_i}, \vec{x_j}) \in C \times C, \left\| \vec{f}(\vec{x_i}) - \vec{f}(\vec{x_j}) \right\| \leq \delta$ (which $\|d\|$ can be any arbitrary distance). Where δ (> 0) denotes the degree of similarity between the solutions provided by the decision-makers.

In the case $\delta = 0$, this indicates that the decision-maker is only interested in finding equivalent optimal solutions of the same quality located in different areas of the search space. $\delta > 0$ indicates that the decision-maker accepts equivalent solutions as well as the solutions that dominate. However, the accepted dominant solutions must satisfy another condition—that is, they must be locally optimal solutions according to definition 2.4, which are of acceptable quality based on the decision-maker's opinion. This implies that the equivalent solutions are not from the same regions in the search space. The goal of MMOEAs is to discover and preserve as many of these equivalent Pareto-optimal solutions as possible [18].

2.2.2.1 Type-I MMOP: Multiple Global Pareto-sets

In the first form of MMOP, it is possible to have multiple global PSs of solutions that indicate the same PF in the objective space. An example of such problems appears in Fig. 2.2.

2.2.2.2 Type-II MMOP: Coexistence of Global and Local Pareto-sets

In the second category of multi-modal multi-objective problems, we encounter scenarios where multiple local PSs exist. In such cases, decision makers often prioritize Pareto-optimal solutions that offer acceptable quality. This preference arises when global solutions are too costly or unattainable, or when there is a need to mitigate risk in situations where global PSs become unfeasible. Figure 2.3 illustrates an example of this particular type of problem.

2.3 Population-based Meta-heuristic Approaches

In contrast to classical search algorithms, which often focus on local optimization and can become trapped in local optima, population-based meta-heuristic

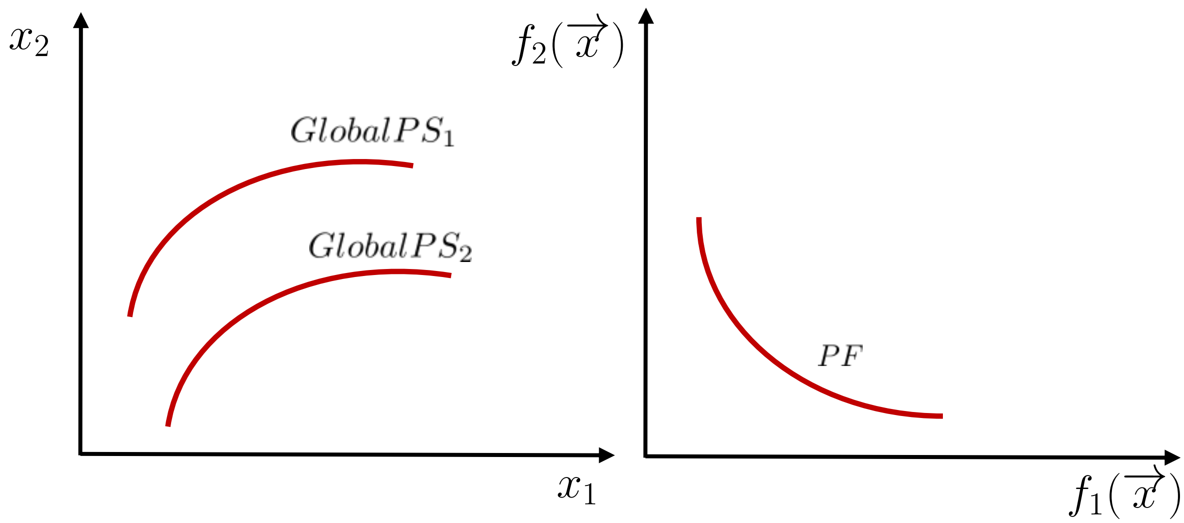


Figure 2.2: An example of a MMOP with two global Pareto-sets of solutions (Type I MMOP)

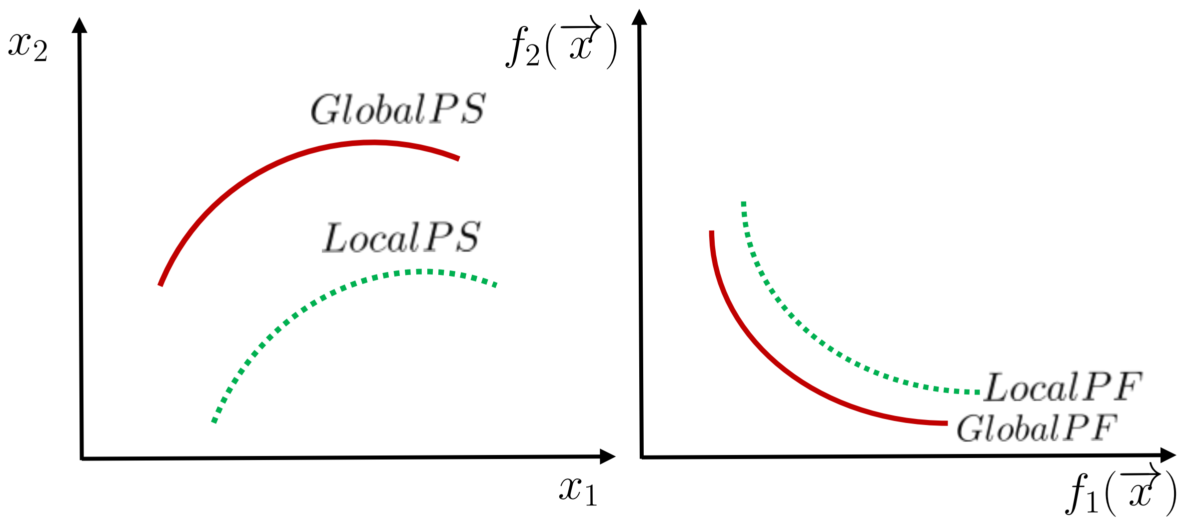


Figure 2.3: An example of a MMOP with both global and local Pareto-sets of solutions (Type II MMOP)

algorithms adopt a different approach. These algorithms strike a balance between two essential strategies: randomization, also known as exploration, and local optimization, also known as exploitation. Exploration enables the algorithm to search unexplored regions, while exploitation allows for the discovery of improved solutions within the proximity of previously found promising solutions. This trade-off between exploration and exploitation helps meta-heuristic algorithms avoid being trapped in suboptimal solutions and promotes the discovery of better solutions within the search space [19]. Since randomization is incorporated into the search process, these algorithms are well suited to finding the global optimum while avoiding the trap of the local optimum.

Meta-heuristics have an advantage over gradient and direct search algorithms in that their convergence toward optimal solutions is not necessarily impacted by the initial position of the population, while the performance of such classical search algorithms is highly influenced by the initial appearance of the solutions [20, 21]. Many meta-heuristic techniques have been developed over the years. Simulated annealing [22], tabu search [23], Monte Carlo methods [24], ant colony optimization [22, 25, 26], particle swarm optimiza- tion [22, 27, 28], and evolutionary algorithms [22, 29] are popular examples of these techniques. Among the various applied mechanisms, evolutionary algorithms have received significant attention during the last few decades due to their ability to generate a well-dispersed set of non-dominated solutions that approximate the Pareto-optimal solution set over a single algorithm run. In contrast to classical search algorithms, these stochastic global search algorithms incorporate biological evolution (Goldberg 1989 [30]), including mating selection, mutation, crossover, and selection of the fittest individuals, into their algorithms to gradually improve the current populations.

2.4 Multi-objective Evolutionary Algorithms

In the field of multi-objective optimization, various evolutionary algorithms have emerged since the 1980s to tackle the complexities associated with solving problems that involve multiple objectives [30]. These algorithms are intended to approximate Pareto-optimal solutions to a given multi-objective optimization problem as uniformly distributed solutions that are close to the ideal Pareto front. It is, however, challenging to achieve this goal. Generally, an approximate Pareto-optimal solution can be obtained by satisfying two conflicting goals: minimization of the distance to the true Pareto front (i.e., convergence to the true PF) and maximization of the diversity of the evolved solutions [31]. To achieve the first objective, Pareto-based fitness assignment (e.g., non-dominance sorting) is often incorporated into some of the most current state-of-the-art algorithms [30] to guide solutions toward the Pareto-

```
   Input: Optimization Problem, Search Space S, Population Size N
   Output: Final population P
 1 t = 0 ;
 2 P(t) ← InitPop(P(t));
 3 Evaluate(P(t));
 4 repeat
 5 |  P_mate ← Select(P(t));
 6 |  P' ← Recombine(P_mate);
 7 |  Q ← Mutate(P');
 8 |  Evaluate(Q);
 9 |  t = t + 1 ;
10 |  P(t) ← Environmental Selection(P', Q)
11 until Termination criteria is fulfilled;
12 return P
```

Algorithm 1: General framework for an evolutionary algorithm [34]

optimal solution [32].

As a means of achieving the second objective, various density estimation methods are integrated with multi-objective evolutionary algorithms to increase population diversity. Moreover, a number of niche technologies have been employed to improve the distribution of the solutions, including archive truncation [31], crowding distance comparison [33], and elitist schemes [31, 33].

Algorithm 1 illustrates a general outline for a multi-objective evolutionary algorithm [34]. Once the population has been randomly initialized (Line 2), solutions are evaluated (Line 3). In the following step, the parents are selected using a mating selection operator (Line 5). The offspring Q are generated by applying the crossover operator and then the mutation operator (Lines 6 & 7). Lastly, the environmental selection process employs the combination of generations and offspring to determine which solutions are carried forward into the next generation (Line 10). The iterative process continues until the specified termination criteria are satisfied (Line 11). Upon meeting the termination criteria, the final population is returned as the optimal solution for single-objective optimization problems, or as the set of Pareto-optimal solutions for multi-objective optimization problems (Line 12).

The following list presents brief descriptions of each of the algorithm components:

- **Evaluation:**

  In optimization problems, the objective function value of each individual serves as an expression of their raw performance. Following calculation of the objective function values of the individuals, the fitness function converts the objective function values into a measure of relative fitness [35]. Within the realm of fitness functions, one widely recognized approach is the Pareto-based fitness evaluation, initially introduced by Goldberg [36] as a means to address Schaffer's problem [37]. This method utilizes non-dominated ranking and selection techniques to facilitate the progression of a population towards the Pareto front during the optimization of multi-objective problems.

- **Mating selection mechanism:** The mating selection process is characterized by a stochastic process in which solutions are selected as parents based on fitness function values derived from evaluation. Selection of mating solutions is influenced by fitness function values, where solutions with higher values have a greater likelihood of being selected and propagating to the next generation. This is to produce a new generation of offspring employing a recombination process. There are a variety of selection methods to select the mates, such as roulette wheel selection [38], stochastic universal selection [39], and tournament selection [40], each of which has its own advantages and disadvantages.

  The tournament selection process provides one way of determining which individuals from one generation will survive and reproduce offspring. The binary tournament method involves the selection of pairs of individuals at random from a population to participate in the tournament. Participants who win the tournament progress to the next generation [41]. An example of typical tournament selection is NSGA-II algorithm 3.2.2, with the front number as the primary consideration and the crowding distance as the secondary consideration. In this manner, it is assured that solutions that are located in less-crowded regions and that have a higher fitness ranking will be preferred.

- **Variation operators (Recombination and Mutation):**

  To further explore the search space, a variation operator is implemented that introduces diverse solutions to the population. The variation operator consists of the recombination and mutation operations derived from the field of natural biology. The operators are implemented in turn on the solutions chosen from the mating process to produce offspring.

- **Recombination:**

  Recombination, also known as crossover, is a genetic operator used to combine the genetic information for two or more parental individuals in order to produce offspring solutions. Currently, several crossover operators are available, including the single-point binary cross-over [38] (the simplest form of recombination), multipoint crossover [42], uniform crossover [43], and simulated binary crossover (SBX) [44].

  The SBX is a variable-wise recommendation operator that simulates a one-point crossover of a binary number with a real number and is well known for handling real-coded applications. For each decision variable, offspring values are calculated by using a probability distribution with a standard deviation determined by the Euclidean distance between the parents.

- **Mutation:**

  Once the recombination operation is implemented for the parental population, a mutation operator is typically applied to the individuals generated from this operator. Mutation is a genetic operator that only requires a single parent to produce a child solution. These operators allow the random aspect to be maintained in the evolution of the population, thus avoiding premature convergence and escaping local optimum. A number of mutation operators are introduced, including the Polynomial Mutation Operator [45, 46] developed for real-valued variables. This operator is used in many metaheuristic optimization algorithms.

- **Environmental selection:**

  Evolutionary algorithms usually have a fixed population size over successive generations that is maintained throughout the optimization process. Since the parent population and its offspring population exceed the predetermined population size, the environmental selection process—known as a selection for survival—is used to determine which solutions are perpetuated to the next generation. To control the complexity of the algorithms and to maintain the same population size over time, it is necessary to implement a strategy for selecting new solutions in each generation.

2.4.1 Exploitation versus Exploration

One of the criteria that is taken into account during the environmental selection process is the quality of the solution candidates, with higher quality solutions being favored, as determined by a fitness function. Selection pressure is one of the major differences between various evolutionary algorithms.

Higher selection pressures are more likely to result in fitter solutions surviving or being chosen as mating parents and less fit solutions not surviving [47]. Therefore, increasing selection pressure results in faster convergence of a population toward particular solutions (exploitation), but the search space is not sufficiently explored. In contrast, under low selection pressure, even after prolonged computation time, it is possible that the population will have a high degree of diversification but may not converge to the optimal solution set (exploration). A good balance between exploration and exploitation processes is essential to improving the performance of MOEAs as exploration of new regions should be complemented by exploitation of those that already have appropriate solutions.

Selection processes (mating and environmental selections) are often referred to as exploitation processes since they provide the best solutions for the next generation of individuals, while variation processes are regarded as a means of identifying the search space for exploration processes [48]. During the early stages of the evolutionary process, exploration is typically essential. Nevertheless, as the search process progresses, exploration becomes increasingly important to enhance the quality of the results [49].

2.5 Multi-modal Multi-objective Evolutionary Algorithms

The majority of MOEAs are ill-equipped to effectively handle MMOPs due to their inherent limitation of lacking mechanisms to preserve diversity within the solution space [18]. When using MOEA to solve MMOPs, it is typically feasible to find only one Pareto-optimal solution for each point on the PF.

This limitation arises because the current design objective of MOEA does not prioritize the maintenance of multiple solutions in the decision space. Consequently, even if an MOEA is applied multiple times, it cannot guarantee the identification of all Pareto-optimal solutions for MMOPs. The dynamics inherent in evolutionary algorithms often lead to a reduction in population diversity in the decision space, further exacerbating this issue. As a consequence, the search may concentrate in a single basin of attraction, as opposed to multiple basins in the landscape, based on the complex dynamics of the various forces [50]. To preserve the distribution of the solutions, MOEAs are introduced to find as many optimal Pareto sets of solutions as possible in the decision space.

2.5.1 Niching

Search space exploration in MMOPs is a crucial and challenging task that involves navigating a complex landscape to discover multiple optimal solutions. The goal of search space exploration in MMOPs is to efficiently explore and uncover the different regions in a search space while maintaining a balance between diversity in both decision and objective spaces. Within evolutionary algorithms, there are three main components that can be attributed to a reduction in population diversity: environmental selection pressure, genetic drift, and crossover and mutation. In this regard, these components are responsible for quick convergence to a single basin of attraction. This prevents algorithms from converging in parallel into more than one basin of attraction [50]. When it comes to selection pressure, most MOEAs apply environmental selection to the whole population, which leads to the following results in MMOPs: the solution with better quality in the objective space eliminates many other dominant solutions, regardless of their position in the decision space. MMOPs are characterized by the possibility that two solutions that are distant in the decision space may be similar in the objective space or may even be identical. As a result, MOEAs eliminate solutions that are crowded in the objective space, regardless of how far they are from one another in the decision space. This may lead to the elimination of equivalent (or slightly inferior) solutions.

The phenomenon of genetic drift [51, 52, 53] can result in a reduction in the population's distribution within the decision space. This natural occurrence in evolutionary processes arises from sampling errors within finite populations, leading to the gradual loss of diversity. The transfer of a genetic characteristic distribution to subsequent generations is constrained by the limited number of offspring generated, which results in a reduced statistical representation of the distribution. As a consequence, the distribution tends to approach an equilibrium state, such as when specific alleles become fixed at the same level of fitness. Genetic drift is commonly recognized as a neutral process. Notably, as the population size decreases, this process intensifies [54].

A third factor that can affect diversity loss in the decision space is mutation and recombination operators [54]. Although these operators have the potential to accelerate the evolution process and help the algorithm escape the local optima trap, there is a greater likelihood that they will produce offspring that are close to their parents, meaning their offspring population will be centered around the parents' location [55]. In such a case, the diversity of the solutions is diminished over the generations. Thus, standard crossover and mutation operators do not effectively produce a wide range of offspring in the decision space.

To resolve the above issues and expand exploration of the search space, niching methods were proposed to mitigate the fundamental challenge of population diversity loss within a population of solutions. These methods are introduced into EAs to assist with finding multiple optimum solutions for multi-modal optimization within a population. They do so by maintaining the diversity of certain properties within the population. The objective of this method is to achieve parallel convergence into a number of different basins of attraction within the multi-modal landscape within a single run of the algorithm.

2.6 Niching Techniques

Due to the term's origins in ecology, the niche refers to the peak or basin of attraction in the context of optimization, and the species are the subpopulations of individuals that occupy the niche [54]. Niche techniques are designed to extend standard EAs to multi-modal domains, taking care to avoid the EAs losing population diversity and converging to a solitary basin of attraction within a single population. It is possible that a niching strategy could reduce the likelihood of being trapped in a local optimum condition by searching for multiple optimum solutions simultaneously. Niche methods have been researched almost exclusively within the framework of genetic algorithm (GAs) in the past four decades. Several decades have been devoted to the development and study of EA niching methods in related fields, such as GAs and evolutionary strategie (ESs).

Throughout the course of this research, a range of algorithmic approaches were developed, drawing inspiration from both bio-inspired concepts related to organic speciation and ecological niches, as well as computational methodologies. The design of such approaches poses challenges from both theoretical and practical standpoints. From a theoretical perspective, two key considerations arise: first, the need to maintain diversity within a population-based stochastic algorithm in a computationally efficient manner, and second, the desire to gain insights from speciation theory or population genetics within the field of evolutionary biology. In the domain of applied optimization, there is a growing interest in providing decision-makers with a set of one or more solutions that effectively capture diverse conceptual designs across both single-criteria and multi-criteria search spaces. The academic literature on EA proposes numerous methods for niching, such as crowding [56], deterministic crowding [57], fitness sharing [38], parallelization [58], clustering [59], restricted tournament selection [41], and speciation [60], among others. A variety of PSO variants, such as NichePSO [61] and SPSO [61], have also incorporated niche methods to leverage their abilities to solve multi-modal optimization problems. However, the majority of such research focuses on multi-modal single objective optimization problems rather than MMOPs; the study of these methods and their implementation does not concern MMOPs. There have been few studies in this area that propose niching methods to manage multi-modality in relation to multi-objective optimization.

2.6.1 Diversity in the Search Space versus Objective Space

It is imperative to note that simply combining MOEAs with the above methods will not resolve the MMOPs problem. When it comes to MMOPs, there are two major issues that should be considered by evolutionary algorithms. To be successful, increased distribution of solutions is required, both within the search space and within the objective space, as is maintaining convergence toward the PF. It is therefore recommended to develop a selection strategy that addresses these three concerns and strikes the right balance between them.

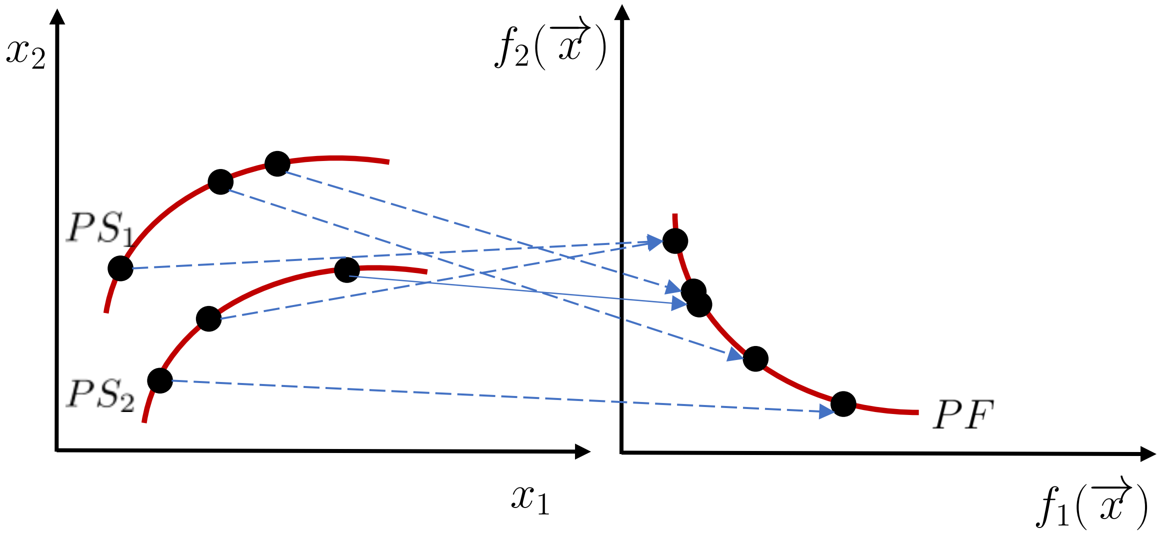


Figure 2.4: A representative example of MMOPs: improving diversity in the search space results in the deterioration of the diversity of solutions in the objective space.

Moreover, it is important to note that the objective space and decision space have distinct and specific requirements. While searching for solutions that are evenly distributed in the decision space may seem beneficial, it can result in inadequate diversity in the objective space when dealing with MMOPs. Thus, it becomes crucial to strike a balance and trade-off between achieving diversity in both the decision space and the objective space.

Figure 2.4 demonstrates how improving the distributions of the solutions in the search space can negatively affect the distributions of solutions in MOPs with multi-modal proportions. The solutions are capable of providing good coverage of the PS in one of the runs of the MMOEAs; however, because the diverse solutions in the decision space can map to similar or close solutions in the objective space, it can lead to the loss of solution diversity in the objective space. On the other hand, as shown in Figure 2.5, improvement in solutions diversity in the unimodal MOPs has been consistent with the improvement in solution diversity in the objective space.

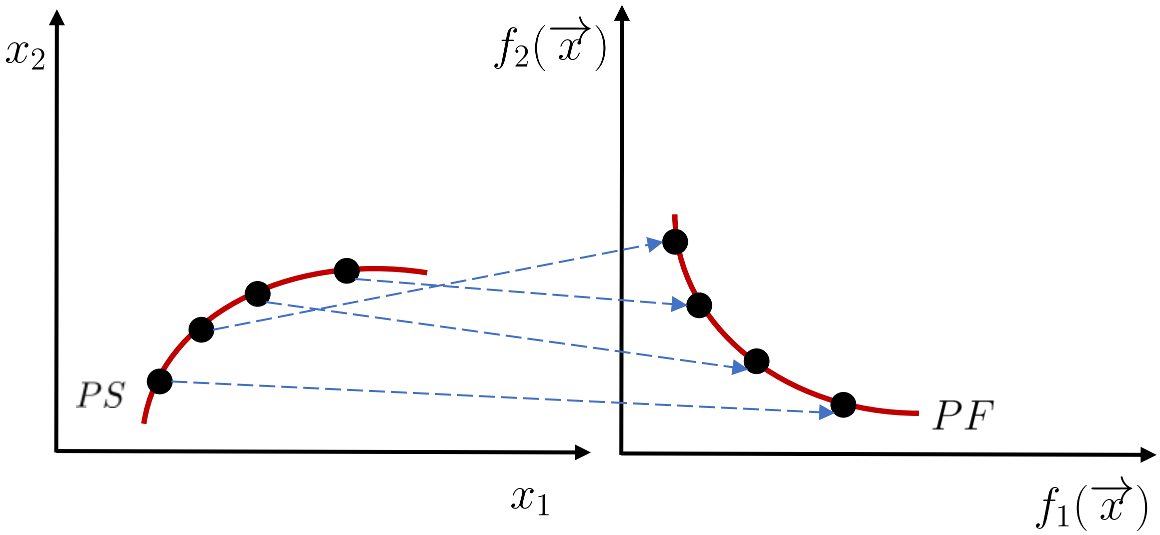


Figure 2.5: An example of MMOPs where maintaining diversity in the search space results in maintaining diversity of solutions in the objective space.

2.7 Benchmark Problems for Multi-modal Multi-objective Optimization

Each of these benchmark problems exhibits different characteristics, which are indicative of the level of difficulty. This is discussed in [62], according to which the degree of difficulty of the test problems varies depending on the PS and PF geometries, the overlap between PSs in each dimension, the number of PSs, and the coexistence of local and global PSs [63]. Some of these benchmark problems can be scaled up to higher dimensions to evaluate the effectiveness of MMOEAs in higher-dimensional search space and/or objective space.

Table 2.1 outlines five desirable characteristics of multi-modal, multi-objective benchmark test cases, which are briefly described in the following. The characteristics of the test cases, as outlined in Table 2.1, provide important information regarding their difficulties as well as to why certain algorithms outperform others in certain test cases and vice versa. For example, one of the indicators of the complexity of MMOPs would be the number of PSs associated with the problem since problems involving high numbers of PSs may be more difficult to solve [63].

Another characteristic, which is the geometry of the PF, is important when considering multi-objective problems involving three or more objectives in terms of convexity/concavity, linearity/nonlinearity, contentiousness or continuity, and so forth [64]. Several algorithms are designed to take advantage of convex PF, while others may be better suited to concave PF. A nonlinear PF is considered to be more difficult to determine than a linear PF. Some algorithms may fail to detect all regions of the PF as a result of the disconnected PF [62]. The MMF1-2 benchmark problems have been modified by symmetry and shift from unimodal multi-objective algorithms. However, they have some limitations, including unknown and complex local PSs.

The geometry of the PS is another important factor that can influence the performance of various algorithms in MMOPs. The characteristics of the PS, such as its connectivity, symmetry, linearity or nonlinearity, and complexity, can vary and lead to different levels of difficulty in solving these problems. Multi-objective evolutionary algorithms may only perform well when applied to certain PS shapes. As an example, a nonsymmetric PS (e.g., MMF1z) is more complex to solve and more similar to real-life problems [63]. Furthermore, it would be beneficial to examine how the MMOEAs operate in escaping the trap of local PS by using test cases in which both local and global PS exist simultaneously. Thus, benchmark problems that possess this characteristic can be used to evaluate the global search capabilities of a given algorithm.

Please note that we have not provided an extensive mathematical description of the benchmark problems in this work. For more specific details, the reader is referred to [62].

Table 2.1: Characteristics of Multi-modal multi-objective testing problems [65]

Test Problem	No. of PSs	Geometry in PF	Geometry in PSs	Local and Global PSs Coexist
Omni-test	27	convex	linear & symmetric	No
SYM-PART-simple	9	convex	linear & symmetric	No
SYM-PART-rotated	9	convex	linear & symmetric	No
MMF1	2	convex	non-linear & symmetric	No
MMF1z	2	convex	non-linear & non-symmetric	No
MMF2	2	convex	non-linear & symmetric	Yes
MMF3	2	convex	non-linear & symmetric	Yes
MMF4	4	concave	non-linear & symmetric	No
MMF5	4	convex	non-linear & symmetric	No
MMF6	4	convex	non-linear & symmetric	No
MMF7	2	convex	non-linear & symmetric	No
MMF8	4	concave	non-linear & symmetric	No
MMF9	4	convex	non-linear & symmetric	No
MMF10	4	convex	non-linear & symmetric	No
MMF11	2	convex	linear & symmetric	Yes
MMF12	2	convex, discontinuous	linear & symmetric	Yes
MMF13	4	convex	non-linear & symmetric	Yes
MMF14	2	concave	linear & symmetric	No
MMF15	2	concave	non-linear & symmetric	Yes

In addition, we will use the polygon-based problems described in [66] to assess the effectiveness of the approach. A polygon-based problem is considered a modified version of a map-based problem. A polygon-based, problem-based optimization algorithm takes advantage of the fact that the Pareto-optimal sets of these problems are distributed across a number of regular polygons. Consequently, it is relatively easy to examine an optimizer's behavior at the beginning of the search process [67]. In the polygon test problem, the proposed algorithms are compared to other competing algorithms to determine their scalability. SYM-PART is another test case we used in this thesis. In this test, each Pareto subset consists of two elements and the vertical and horizontal distances between adjacent Pareto subsets [68].

2.8 Performance Indicators

Compared to single-objective optimization problems, where the quality of the obtained solutions is assessed based on their fitness function values, in multi-objective optimization problems, it is necessary to measure performance by mapping them to a real number (referred to as a performance indicator) that allows us to evaluate the algorithms.

In this thesis, several performance indicators are employed to assess the diversity and convergence of the algorithm results. These measures include the Inverted Generational Distance Plus (IGD+) in the objective space [69], the Inverted Generational Distance indicator in the decision space (IGDx) [70], and the Pareto Set Proximity (PSP) [71]. The inverted gerneational distance indicator in the decision space (IGDx) and PSP indicators are meant to illustrate algorithms' performances in the search space, whereas Inverted generational distance plus in the objective space (IGD^+) indicators illustrate algorithms' functionality in the objective space [63].

Definition 2.6 (IGDx) [70] IGDx is a widely used indicator for evaluating

the performance of multi-modal, multi-objective optimization algorithms. It is required that sampling points be taken from the true PS to evaluate the quality of the obtained solutions set in the decision space.

Let P^* denote a set of solutions distributed uniformly across the true PS. Let that R represents the obtained set of solutions in the decision space. Accordingly, the IGDx metric is defined as follows:

$$\text{IGDx}(P^*, R) = \frac{\sum_{v \in P^*} |R - v|_2}{|P^*|} \tag{2.8}$$

where the distance $|R - v|_2$ is determined as the minimum Euclidean distance between a point v and any other point of R. The cardinality of P^* represented by $|P^*|$.

The IGDx indicator evaluates the diversity and convergence of the obtained solutions relative to the PS. Lower IGDx values indicate a more accurate approximation of the PS, indicating that the solutions have converged to the true PS and are well-distributed.

Definition 2.7 (PSP) [71] A Pareto-set proximilty (PSP) performance

indicator measures both the overlap ratio as well as the distance between the obtained PS and the true PS. It is desirable to have a larger PSP value. A PSP indicator for the solution set R regarding P^* can be defined as follows:

$$\text{PSP}(P^*, R) = \frac{CR(P^*, R)}{\text{IGDx}(P^*, R)} \tag{2.9}$$

$$\text{CR} = \left(\prod_{i=1}^{n} \delta_i \right)^{\frac{1}{2n}} \tag{2.10}$$

Where CR (cover rate) is a modified version of the Maximum Spread (MS) [72] for the search space:

$$\delta_i = \begin{cases} 0 & \text{if} \quad q_i^{max} \leqslant Q_i^{min} \quad or \quad q_i^{min} \geqslant Q_i^{max} \\ 1 & \text{if} Q_i^{min} = Q_i^{max} \\ \left(\frac{(min(Q_i^{max}, q_i^{max}) - max\left(Q_i^{min}, q_i^{min}\right)}{Q_i^{max} - Q_i^{min}} \right) & \text{otherwise} \end{cases} \tag{2.11}$$

where n denotes the number of decision variables, the terms Q_i^{max} and Q_i^{min} correspond to the upper and lower bounds of the true PS in dimension i. Similarly, the parameters q_i^{max} and q_i^{min} represent the maximum and minimum values of the obtained PS for the i_{th} decision variable.

Definition 2.8 (IGD$^+$) [69] The Inverted Generational Distance indicator in

the objective space (inverted generational distance indicator in the objective space (IGD)) [69] is one of the most commonly used in academic literature. The sampling points from the true Pareto front must be used to calculate the performance indicator. IGD values are calculated as the average of the distances between each sample reference point and the nearest solution. It is important to note that this performance indicator is not Pareto-compliant. This means that solutions that are better based on a Pareto dominance relationship may nevertheless be regarded as less desirable. Therefore, we utilized the IGD$^+$+ performance indicator [73], which is the extended version of the IGD performance indicator [74] that has been formulated so as to be weakly Pareto-compliant.

When computing the distance of the IGD$^+$ indicator, if a solution is dominated by a reference point, Euclidean distance is applied without change. If, on the other hand, the solution and the reference point do not dominate each other, we calculate the minimum distance between the reference point and the dominated region by the solution. This modification is derived from the fact that the Euclidean distance between the reference point and the solution indicates the degree of the solution's inferiority in comparison with the reference point, which should be minimized. Therefore, if the reference point dominates the solution, the Euclidean distance is used without change for the measurement. In contrast, if the solution and the reference point are not dominating one another, then reducing the distance by moving the solution toward the reference point may not necessarily improve all the objective functions of the solution. As a result, the usual Euclidean distance cannot be viewed as a measure of inferiority. The negative value of d_i in the formula for the IGD distance indicator would be deemed to have no inferiority (i.e., no insufficiency). Therefore, the IGD$^+$ performance indicator would be formulated as follows:

$$IGD^+(V,U) = \frac{1}{u}\sum_{u \in U} min \geqslant d^+(v,u) \tag{2.12}$$

Where $U \subset R^m$ is the sampling point from the true PF and V represents the approximation of the PF in this formula. The proposed modified Euclidean distance is calculated as follows:

$$d^+(v,u) = \sqrt{\sum_{k=1}^{m} max(v_i - u_i, 0)^2} \tag{2.13}$$

2.9 Summary

This chapter introduces the basics of multi-modal and multi-objective optimization, describing the concepts used in this dissertation. A formal definition of multi-objective optimization problems, multi-modality, and Pareto optimality is provided. Moreover, we describe the characteristics of those benchmark problems as well as the performance indicators used in this thesis to compare the performance of individual algorithms.

Chapter 3

Related Work and Foundations for Experimental Comparison

This chapter provides an examination of advanced algorithms used for addressing multi-modal multiple-objective optimization problems (MMOPs). These problems are generally categorized into two types: the first type focuses on multi-modal, multi-objective optimization problems with multiple global Pareto sets, while the second type deals with problems involving both local and global Pareto sets.

In Section 3.1, we provide an overview of the literature on multi-modal problems, including both multi-objective optimization algorithms and single-objective optimization algorithms. Section 3.2 addresses the algorithms introduced to solve the first class of these problems. To classify these algorithms, three major categories can be identified: dominance-based, decomposition-based, and indicator-based MMOEAs. In particular, we devote a significant portion of our attention to dominance-based search algorithms in Section 3.2, which is the focus of the remaining sections of this dissertation. Lastly, state-of-the-art algorithms for dealing with the second category of multi-modal, multi-objective optimization problems are discussed in Section 3.3.

3.1 Overview of State-of-the-art Algorithms

The concept of multi-modality has been studied for years in the context of single-objective optimization problems. However, multi-modal, multi-objective optimization is still a relatively unexplored field that requires a great deal of attention, which is why this thesis is primarily concerned with that area. In the following section, we present a brief overview of the existing methods for solving multi-modal, single-objective optimization problems, after which we provide an overview of multi-modal optimization methods in the context of multi-objective optimization.

3.1.1 Multi-modal Single-objective Optimization

Several approaches have been proposed to tackle the challenge of locating multiple global and local optima in multi-modal optimization problems. These include strategies like crowding populations [56] and niching techniques [75], among others. Niching techniques, discussed in Section 2.6, have played a crucial role in advancing multi-modal optimization algorithms. Over the years, numerous methods have been developed to address multi-modal, single-objective optimization problems, starting from the late 1970s [14].

As explained in Chapter 2.6, niche refers to the subdomain of the search space or region of the search space surrounding a particular peak in a fitness landscape, while species refers to the subset of populations that share the same characteristics. The different approaches to structuring species within niches employed by different niching methods illustrate how effective the methods are for exploring the search space aimed at discovering the peaks of fitness landscapes [76]. In some of the niching methods, such as clearing [77], fitness sharing [78], SPSO [79], and SCGA [60], which are based on distance between individuals, the definition of local neighborhood structures is required. Some of these methods consider the niche radius as a similarity measurement to group species whose distance is less than the neighborhood radius and which have a greater likelihood of forming a population of the same niche. It is, however, difficult to fine tune the niche radius, especially in cases of unevenly distributed optima or in a complex fitness landscape, since this requires prior knowledge; this is viewed as a disadvantage of these methods [76].

There are various niche methods, including clearing [80], speciation [60], fitness sharing [37], restricted tournament selections [81], probabilistic and deterministic crowding [82], and crowding [83]. The species in a population is comprised of individuals, including parents and offspring generated through the reproduction process in genetic algorithms. Within each niche, individuals with higher fitness function values are selected to propagate the next generation. Niche techniques offer the advantage of not needing niche parameters to be fine-tuned. However, there is a possibility of losing the discovered niche during the replacement process, as some members may belong to different niches.

In 2009, Li [84] proposed a new kind of niching method in which the ring topology between the individuals and their personal best is employed as a neighborhood structure in the PSO algorithm. By using particle local memory, this method creates a stable network while maintaining the best position found so far. As a consequence, the ring topology facilitates the creation of stable niches across different local neighborhoods and thus helps to find multiple local or global optima. As a follow-up to the previous study, Qu et al. [85] proposed a distance-based locally informed particle swarm (LIPS) in 2012. By integrating information from the Euclidean nearest neighbor of the personal best of each particle, the algorithm has been extended with local search capabilities and has updated the particle velocity with the PSO algorithm. The advantage of this approach is that it increases the possibility of considering neighboring individuals in the same niche, thereby raising the exploitation potential of the algorithm through interaction with local solutions.

In addition, a number of other niching-related methods have been proposed in the past few years to handle multi-modality in single objective optimization problems using Gravitational Search Algorithms (GSAs), which can be considered a powerful technique to solve a variety of these problems, including GGSA [86], QISGA [87], BQIGSA [88], and CoGSA [89]. Additionally, some other niching techniques have been suggested that may also be of interest to you. You may refer to some existing surveys on niching methods within the context of multimodal, single-objective optimization to gain a better understanding of these techniques [81, 90, 91].

3.1.2 Multi-modal Multi-objective Optimization

The use of MOEAs poses a considerable challenge in solving MMOPs, in part because they tend to increase the distribution of solutions in the objective space without taking the distribution of solutions in the search space into consideration. It is therefore essential to develop approaches for increasing diversity in the search population in order to capture Pareto-optimal solutions by covering all parts of the Pareto front to solve MMOPs. Although there has been a relatively limited amount of research conducted on multi-modal multi-objective optimization problems (MMOPs);

3.2 Classification of related approaches for Multi-modal Multi-objective Optimization with Multiple Global Pareto sets (Type I MMOP)

Regarding MMOEAs types, there are two major types: decomposition-based MMOEAs and Pareto-dominance-based MMOEAs. The decomposition-based MMOEAs [92] are an enhanced version of the MOEA/D algorithm [93, 63]. The two extended versions of MOEA/D used to solve MMOPs are described in [92] and [94]. According to the MOEA/D algorithm, an optimization problem with M objectives is divided into N single-objective problems, with each of these sub-problems being assigned to a separate individual [63]. As a result of the MOEA/D algorithm, N individuals evolved simultaneously. There are two variations of MOEA/D when it comes to aging MMOPs, which each assign one or more individuals with responsibility for dealing with equivalencies within each sub-problem [18].

MMOEAs from the second category are largely extended versions of the NSGA-II algorithm: the solutions are sorted into fronts in the environmental selection process based on the non-dominance sorting relationship. Then, secondary selection is performed by incorporating different niche selection techniques to preserve the solution distribution throughout the search space. An example of a developed algorithm that can be classified into this category can be seen in the studies [95, 96], which make use of clustering techniques to maintain multiple optimal solutions during development of the algorithm. As a result of this, multiple stable subpopulations within a population are maintained, allowing multiple optimal solutions to be preserved within the population. The more recent approach is to employ external archives to maintain a variety of non-dominated solutions in the decision space [97, 98, 99]. There have been several proposed MMOEAs that have implemented the crowding diversity measure in decision space in order to deal with MMOPs [71, 100, 101, 102, 103].

Moreover, discussions have been held regarding the implementation of diversity or convergence indicators into the set-oriented optimization procedure, a method that has shown some promise [63]. MMOEAs can be improved using these indicators to preserve the distribution of solutions in the search space as well as provide other interesting options [12]. The gap indicator (or the average distance to the nearest neighbor) [104] is a simple indicator that helps to increase the diversity of the optimal solutions obtained. A further example is the Rietze S-energy indicator, which produces a uniform distribution of points over a number of manifolds and is scalable when it comes to the number of decision variables [12, 63].

3.2.1 Decomposition-based Multi-modal Multi-objective Evolutionary Algorithms

Decomposition-based multi-modal multi-objective evolutionary algorithms (MOEAs) are a type of optimization algorithm for solving multi-objective optimization problems with multi-modal or local optimal solutions. MOEA/D is the most representative example of a decomposition-based algorithm [105]. In spite of the fact that MOEA/D has proven to be a promising candidate for solving a variety of MOPs, it has not been shown to be suitable for solving MMOPs, as reported in [94]. This is the result of only assigning one solution to each weight vector, which is responsible for directing the search in a Pareto-optimal direction. It is therefore impossible for MOEA/D to preserve multiple equivalent solutions.

As a general principle, these algorithms aim to decompose multi-objective problems into smaller sub-problems that are easier to solve. Multi-modal MOEAs based on decomposition typically consist of the following steps: initialization, in which a random population of solutions is generated; and decomposition, in which a multi-objective problem is split into several sub-problems, each representing a local optimum. Partitioning of the objective space is based on regions that correspond to individual sub-problems. The generation of solutions uses either a local search algorithm or a global optimization algorithm to generate a set of solutions for each sub-problem. Solution selection consists of evaluating and selecting solutions based on their fitness values for each sub-problem. Once the selected solutions have been combined, the new population is formed. It consists of all solutions that have not been dominated by any other solution in the Pareto Archive. The purpose of this structure is to store non-dominated solutions that have been identified so far in the search process. Upon meeting a particular stopping criterion, such as a specified maximum number of iterations or an acceptable level of accuracy, the algorithm will terminate.

A multi-modal MOEAs driven by decomposition has the advantage of providing a set of distributed solutions. A decomposition-based multi-modal MOEAs, however, has limitations as well. In particular, they may be sensitive to the choice of sub-problems and partitioning of the objective space, and they may not work well with complex or nonlinear problems. Below is a brief description of the proposed multi-modal MOEAs based on decomposition.

In 2018, **MOEA/D-AD** [106] was proposed as an alternative method of solving MMOPs based on MOEA/D . MOEA/D-AD aims to enhance the exploration-exploitation trade-off during opti. MOEA/D-AD aims to enhance the exploration-exploitation trade-off during optimization. By generating reference points representative of different regions in the space, the proposed algorithm provides a more profound understanding of the objective space. A more balanced exploration of the space and improved solutions are then achieved by guiding the optimization process using this information. With MOEA/D-AD, each weight vector has a subpopulation size that varies adaptively.

In the proposed MOEA/D-AD algorithms, there is an assignment of a weight vector w_i to a particular offspring y closest to it in the objective space, and the subpopulation of w_i is referred to as P_i. From all the solutions in the decision space, the nearest L solutions to y are selected (referred to as Q). In the event of either of the following two conditions being satisfied, the offspring 0 will be added to Pi: If $p_i \cap Q \neq \varnothing$, and at least one solution in $p_i \cap Q$ has the worse scalarizing function value than y. It will then be possible to remove those solutions from Pi that are worse than O. Whenever $p_i \cap Q$ reaches $\varnothing$, then there will be a maintenance of the niching structure in the decision space by following the techniques described earlier if $p_i \cap Q = \varnothing$. Offspring O competes with only those solutions that are its neighbors in the decision space. Therefore, multiple equivalent solutions may be kept in a subpopulation of a weight vector. The idea of assignment, deletion, and addition has been simplified into a framework of operations in [107], which can be used to enhance the performance of decomposition-based MMOEAs. MOEA/D-AD has the disadvantage of working with an unbounded population, resulting in a huge population after many iterations.

The proposed algorithm, **Tri-MOEA&TAR** [16], in 2019 offers a solution to Multi-Objective Optimization Problems MOP by utilizing two archives: the convergence archive and the diversity archive, combined with a recombination strategy. The algorithm starts by employing a decision variable analysis method to identify the decision variables linked to convergence and diversity. The information derived from this analysis is then utilized to enhance the convergence and diversity of solutions within the two archives. Finally, the recombination strategy is applied, resulting in the generation of a substantial number of multiple Pareto-optimal solutions.

3.2.2 Pareto-Dominance-based Multi-modal Multi-objective Evolutionary Algorithms

The majority of MMOEAs fall into the Pareto-dominance-based classification, which is based primarily on the NSGA-II algorithm, as illustrated in Figure 3.1. The mating pool consists of existing populations of solutions and their offspring solutions, which are the result of recombination and mutation. A subsequent environmental selection process determines which solutions from the mating pool will be passed on to the next generation. After sorting the solutions using the non-dominance sorting method, they are passed to the next generation, beginning with the first front (i.e., front_1). The process will continue until the number of solutions in front_i added to the next-generation population exceeds the number of the current population. The next step is to compute the crowding distance of each solution on the front_i and then to transfer those placed in sparser areas in the objective space to the next generation. Crowding distances are used by MOP strategies to increase the diversity of solutions in their objective space. To calculate the crowding value for each individual of front_i, it is necessary to sort the population according to the values of each objective function. A value of infinite distance is assigned first to the boundary solutions. The distance is computed by aggregating the absolute normalized difference between the function values of two consecutive

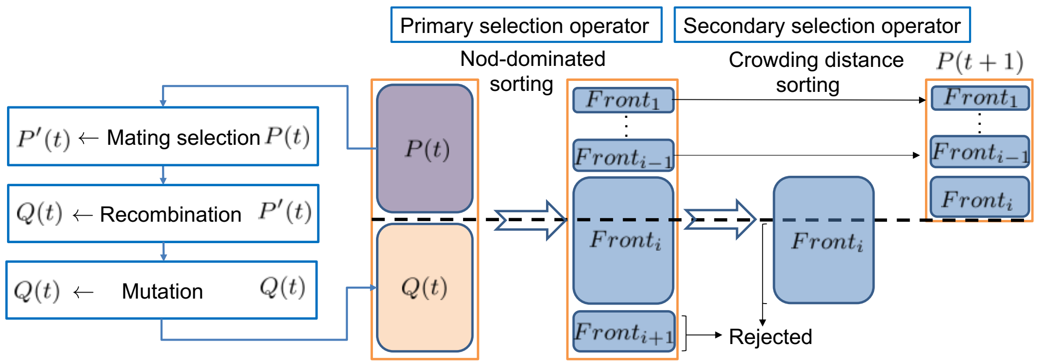


Figure 3.1: Environmental selection operation for the NSGA-II algorithm.

solutions. This is achieved by summing up the absolute normalized difference between the values of adjacent solutions, as described in [108].

As one of the first algorithms to address MMOPs, **Omni-optimizer** [100] was widely recognized in 2005 as one of the most representative MMOEAs [18]. The method can be employed to achieve multi-optima either for a single-objective problem or for a multi-objective problem. The method combines several optimization methods, including genetic algorithms, particle swarm optimization, and local search, to provide an extremely flexible and powerful optimization method. In this algorithm, the following approaches are employed. First, the Latin hypercube is used to generate the initial population uniformly. The second step consists of choosing the two individuals who will participate in the tournament selection process using the nearest neighbor approach (i.e., restricted selection). In addition, a two-tier fitness assignment scheme is employed, which calculates primary fitness according to phenotypes (objectives and constraints), while secondary fitness calculation is based on both phenotypes and genotypes (decision variables). Omni-optimizer has the advantage of being flexible; it is capable of handling a wide range of optimization problems and can be modified to meet different objectives and constraints. In addition, it has a hybrid nature that allows it to leverage the advantages of multiple optimization techniques, making it a powerful and efficient optimization tool. The nature of the problem and the parameters that are used can, however, have a significant impact on the algorithm's performance.

The decision space based **Niching Non-dominated Sorting Genetic Algorithm II (DN-NSGA-II)** is a multi-objective optimization algorithm that was proposed in 2016 [101]. This algorithm incorporates a niching method and a selection operator for creating the mating pool and selecting offspring. DN-NSGA-II includes a mating selection method based on solution distances. A brief summary of this algorithm can be summarized as follows. First, a niching method is utilized to create the mating pool. For solutions to compete with each other, they must all belong to the same niche. the process is as follows: the pro Initially, a solution is randomly selected from a population of candidates. In the second step, a constant number of solutions is chosen at random from the remaining population (crowding factor [CF]). In the third step, Euclidean distance is calculated between the current solution and the CF solution, and then the closest solution is selected. A final step is to add the best solution from the two options (current solution and closest solution) to the mating pool. The four steps described above should be repeated to fill the mating pool. In addition to the first modification, the selection operator has been modified in a second way. To improve the distribution of the solutions in the search space, the crowding distance technique was used in the decision space rather than the objective space for the secondary selection criteria.

A **Double-Niched Pareto Genetic Algorithm (DNEA)** algorithm was developed in 2018 [67], which is similar to Omni-optimizer except that it introduces two sharing functions between objective and solution spaces. Compared to standard generational evolutionary algorithms, DNEA is characterized by its environmental selection, which is a significant difference. In both the objective and decision spaces, this algorithm estimates the density of a solution. Using double-niching techniques, it is possible to take into account solutions that are very close to one another within the decision (objective) space while being very distant from one another within the decision (objective). This algorithm is capable of preserving diversity both in the decision space and the objective space. However, two parameters that are used in the sharing-niching algorithm must be fine-tuned.

In contrast to the aforementioned MMOEAs that utilize genetic variation operators like SBX crossover and polynomial mutation for inducing genetic variation, the subsequent MMOEAs are founded on diverse approaches.

The **MO-Ring-PSO-SCD** algorithm is based on Particle Swarm Optimization (PSO) [71], another representative MMOEAs that has received considerable attention since its introduction in 2018. The proposed approach optimizes particles locally by moving them toward the best positions that they have already found or those found by others in their ring. The ring topology [109] is then used to introdunce multiple niches. A MO-RING PSO finds the global optimum by interacting the particles in different rings with each other and exchanging information about their best positions among the particles. Afterward, the global best position is updated based on the best position found so far by each particle. By exploring new regions of the search space, the particles are able to find the global optimum. Furthermore, MO-Ring-PSO-SCD maintains diversity both in the objective and decision spaces by utilizing a density measurement approach similar to Omni-optimizer while handling boundary individuals differently in the objective space.

Following the previous study [71], in 2019 **MO-PSO-MM**, a self-organizing mechanism, was proposed [110], which involves the introduction of a self-organizing mechanism allowing the determination of population distribution and the construction of neighborhoods in the decision space during the evolution process. It is then possible to map solutions that are similar to each other into the same neighborhood. As in Omni-optimizer algorithms, a special crowding distance was utilized to preserve the diversity of solutions in the decision and objective spaces.

The **Niching-CMA** algorithm [111], introduced in 2010, extends the CMA-ES approach to address decision space diversity in evolutionary multi-objective algorithms. It employs an aggregate distance metric in both the objective and solution spaces to classify individuals into multiple niches. Individuals with higher non-domination levels within each niche have a survival advantage. The concept of space aggregation is also introduced to preserve diversity in the aggregated spaces (objective and decision spaces). This study demonstrates that the Niching-CMA method for multi-objective optimization can generate more diverse solutions. However, fine-tuning the user-defined parameters can be a challenging task.

Since its introduction in 2004, the SPEA2+ algorithm has been a significant improvement over the original SPEA2 algorithm [112] in terms of maintaining multiple diverse solutions [99, 98]. Various modifications and enhancements have been implemented to increase the algorithm's efficiency. These include revisions in the environmental selection process, adjustments in fitness calculation, and more efficient memory utilization. The algorithm incorporates two archives, one for the objective space and one for the solution space, to preserve the diversity of solution options. The archive in the objective space is determined based on the density of individuals within the objective space during the environmental selection process, while the archive in the decision space is determined by the density of individuals within the solution space. During mate selection in SPEA2+, individuals from the objective space archive are exclusively chosen from the neighborhood in the objective space.

Niching-CMA represents a significant improvement over the SPEA2 algorithm with regard to maintaining multiple diverse solutions. A number of changes and modifications are introduced to enhance the efficiency of the algorithm. These include revisions in the environmental selection process, modifications to the fitness calculation, and a more efficient use of memory. Two archives are used in this algorithm to maintain the diversity of solution options: one for the objective space and one for the solution space. According to SPEA2 [112], the archive in the objective space is determined by the density of individuals within the objective space for the environmental selection process, while the archive in the decision space is determined by the density of individuals within the solution. When selecting mates in SPEA2+, individuals within the neighborhood in the objective space are only selected from the archive in the objective space.

3.2.3 Indicator-based Multi-modal Multi-objective Evolutionary Algorithms

MMOEAs that rely on indicators tend to steer their evolutionary progress using reference vectors aligned with the chosen performance indicators. However, it should be noted that indicator-based designs do not guarantee a uniformly distributed optimal set of solutions, although they can guide the search process and yield favorable performance indicator values such as hypervolume. Nevertheless, the practical application of these approaches in real-world scenarios is challenging due to their high computational complexity.

Using a weighted indicator, a novel algorithm for assessing the convergence quality of solutions was developed in 2019 called the **MMEA-WI** algorithm, which utilizes a weighted indicator to evaluate the potential convergence quality of the solution. An indicator based on the IBEA algorithm [113] has been developed. By summing the fitness of other solutions depending on their distance from the solution, the weighted indicator of the solution is calculated. Thus, solutions tend to crowd the PS as a result of this process. We have introduced a convergence archive to maintain the uniformity of the solutions as well as improve the convergence capability.

3.3 Related work for Multi-objective Optimization with Multiple local Pareto-sets (Type II MMOP))

There have been several studies on solving MMOPs, but most of them have focused on the problem of finding multiple equivalent global Pareto solutions. MMOPs with local Pareto solutions are seldom addressed. Nevertheless, local Pareto solutions are of great importance when it is not possible to apply global ones. It is difficult to locate the local PS when the non-dominated sorting method is applied throughout the entire population. The reason for removing these local Pareto-optimal solutions from the selection process is due to the fact that, when considering environmental selection, local Pareto-optimal solutions tend to be dominated by global Pareto-optimal solutions. Therefore, these local Pareto-optimal solutions are usually discarded. Nevertheless, in many instances, decisions are made in favor of local Pareto solutions when global solutions are not feasible or are too expensive to be accessed. Consequently, it would be beneficial to develop strategies for locating local PSs. In this section, we give an overview of some of the recent research devoted to resolving these types of problems in a way that approximates their global PSs while maintaining some suitable local PSs as well.

To locate both global and local Pareto solutions, the **Multimodal Multi objective Genetic Algorithm (MMOGA)** was introduced in 2020 [114]. The Euclidean distance in decision space determines each individual's neighborhood in MMOGA. Individuals within their own neighborhood are limited to mating and reproducing with each other. In this way, the population has the opportunity to evolve into different niches. In this study, individuals are compared only with their neighbors rather than with the entire population as a whole. As a result, the local Pareto-optimal solution is not eliminated from the search process.

However, it is possible that certain individuals may have been mistakenly identified as local Pareto-optimal solutions. To address this issue, the algorithm combines the non-dominated solutions within each niche and removes the solutions that are dominated by their neighbors. The experimental results demonstrate that the proposed algorithm effectively solves MMOPs involving local Pareto sets. However, it is important to acknowledge that this study serves as a preliminary exploration in identifying both local and global Pareto sets for MMOPs. Despite the improved performance of the proposed algorithm for MMOPs with local Pareto sets, it is not as efficient as some algorithms designed for MMOPs with only global Pareto sets. Additionally, the algorithm entails a high computational complexity [114].

A 2021 research paper addressed MMOPs with local PSs, called **MMOEA/DC** [115]. The new MMOEA incorporates dual clustering in both the decision and objective spaces. As MMOP problems are solved, the focus is on two key goals: finding multiple good global PSs in the objective space that have the same quality and finding good local PSs of acceptable quality. MMOEA/DC is able to generate both global and local PSs through the application of the double clustering method, which is the hierarchical clustering method (HCM) and the neighborhood-based clustering method (NCM) within the decision space within the objective space. Averages of harmonic distances are utilized to evaluate the crowding distances between solutions in the decision space. To reduce the number of solutions within each cluster, an iterative process is employed where the most crowded solution in the decision space is successively eliminated from the most crowded cluster in the objective space until only one solution remains. The dual clustering approach adopted in this study effectively balances the preservation of good local Pareto sets of acceptable quality with the preservation of global Pareto sets throughout the evolutionary process. Based on the findings of this research, a crowding-based mating selection strategy is proposed, which employs a binary tournament selection method that considers the harmonic average distance values within the decision space. This strategy aims to select offspring that exhibit greater variation within the decision space, thus promoting diversity in the offspring population [115]. As a result, it is possible to select mating parents that are more uniformly distributed, thereby producing offspring that exhibit greater genetic diversity. This algorithm carries a number of advantages, including when solving MMOPs when both local and global PSs coexist by utilizing dual clustering to distinguish between good local PSs and good global PSs.

To solve MMOPs specifically to preserve local optimal solutions, **DNEA-L** is proposed as an enhancement to the DNEA algorithm [116]. Using a multi-front archive update method, both a global and local PS can be obtained. The framework of DNEA-L is based on the following approach: multiple non-dominated fronts are stored in a multi-front archive. Based on the proposed approach, the neighbors of each solution whose distance to the solution is less than the neighborhood radius identified in the decision space. Those solutions that are dominated by their neighbors in the population are eliminated because they are not locally Pareto-optimal. After sorting, non-dominated fronts are constructed from the remaining solutions. Solutions with the largest double-sharing functions are removed from the first K fronts when they exceed the population size, reducing the number of solutions on each front to N. Based on this algorithm, it would be possible to produce both global and local Pareto-optimal solutions with a high degree of diversity.

Using a **hierarchy ranking algorithm (HREA)**, an evolutionary algorithm was developed in 2022 [117] to determine the global and local preferences of decision-makers. An indicator of local convergence was used by HREA to assess local convergence. The indicator would allow both global and local PSs to be preserved during evolutionary processes. It is proposed that a hierarchy ranking method can be used to maintain the quality of the obtained local PF by balancing convergence and diversity. Moreover, IDMPe is proposed, which is based on IDMP and allows for the adjustment of the number of global and local PSs.

3.4 Foundation for Experimental Comparison

IIn this section, we describe the general experimental settings used in the majority of the experiments discussed in the following chapters. A specific experiment's settings may differ from those described in this section, in which case they are indicated accordingly.

Experiments were carried out using PlatEmo Platform Version 2.8.0 with Matlab R2020a running at a speed of 3 GHz with 16 GB of RAM in a 64-bit environment and an Intel Core i7 processor. Unless otherwise specified, we considered 100 as the population size. Each algorithm executes a maximum of 10,000 function evaluations for all experiments to meet the termination criteria [63]. In this study, we used the Mann-Whitney-U test or Wilcoxon rank sum test [118] to determine whether our results are statistically significant under the null hypothesis that the medians of both samples were equal. As a threshold value in this study, we used 0.01 as a test of whether there are statistically significant differences between algorithms when the p-value is less than 0.05. Additionally, we demonstrate the wins, losses, and ties based on statistical significance across all test suites for each algorithm versus the proposed algorithm.

As part of the experiment, we utilized the IGD$^+$ performance metric to assess whether the obtained optimal solutions in objective space resembled PF solutions. As a measure of the quality of optimal solutions obtained in an objective space, this indicator is designed to determine the degree of diversity and convergence. Furthermore, we used the IGDx performance indicator to assess the similarity between optimally obtained solutions and PSs in the search space. In this manner, we are able to examine both the diversity and convergence of optimally obtained solutions in the decision space. In addition, the PSP performance indicator is also used as a secondary metric calculated by dividing the cover-rate by the IGDx values, which illustrates the overlap between the PS and the derived results. Most commonly, the metrics mentioned above are used to assess the performance of MMOEAs. Notably, the lower value of the IGD$^+$ and IGDx values results in a better convergence of the obtained solution to the PF and PS, whereas the higher value of PSP indicates a better convergence and cover rate for the obtained optimal solution to the PS [119]. In Section 2.8, the above-mentioned indicators are described in detail.

We have computed the IGD$^+$, IGDx, and PSP values using a sample of the respective PF and PSs of the benchmark problems to evaluate the performance of various algorithms. In the reference point sampling approach, we used uniformly distributed reference points on each PF and PS to calculate performance metrics of the solutions obtained by MMOEAs.

Taking n as the number of decision variables, $P_c = 1$ and $P_m = 1/n$ represent the probabilities of SBX and polynomial mutation, respectively. $\eta_c = 20$ and $\eta_m = 20$ represent the distribution indices of these operators, respectively [63]. These configurations are implemented when any algorithm employs polynomial mutation and SBX crossover.

3.4.1 Configuration of the State-of-the-art Algorithms

Some of the related works have been implemented in PlatEmo, similar to their original publications. Furthermore, there are implementations of related works provided by the algorithm authors. Our own contributions have also been incorporated into the PlatEmo framework in addition to these contributions. Based on the examination of the available source codes for the algorithms, the author of the thesis identified some algorithms that were employed in the study. Specifically, the MMOEA/DC, Mo-Ring-PSO-SCD, and DNEA algorithms were utilized in this thesis.

To compare the state-of-the-art algorithms with the proposed algorithms, we used algorithms representative of the two major categories. The Mo-ring-PSO, Omni-optimizer, DNEA, and DN-NSGA-II algorithms are the most representative of the Pareto-dominance-based acpmmoea, the source code for which was available. Furthermore, our proposed algorithm for solving MMOPs of type II was compared with the MMOEA/DC algorithms, which are the most representative of the second category of decomposition-based MMOPs. The NSGA-II algorithm is used as the baseline algorithm for comparison with the other proposed algorithms.

The algorithm parameters have been configured in accordance with the specifications provided in the original publications by the respective authors. In the following list, we provide the settings for the parameters of the related methods. Any changes to parameters are noted in the corresponding section.

- NSGA-II source code was used as it is implemented on the PlatEmo platform, and parameters were configured based upon those reported in the original literature [33].

- The Mo-Ring-PSO-SCD source code was obtained from the corresponding author's website, and the parameters were configured according to the literature [71]. Consequently, $C_1 = C_2 = 2.05$ and $W = 0.7298$ were set.

- For the MMOEA/DC algorithm, the parameter λ, which determines the radius of the neighborhood in NCM, was set to 0.1 while the parameter β, which sets the minimum number of solutions in each cluster, was set to 5.

- In the proposed DNEA algorithm, Niche radius in the objective space was set to $\delta_{obj} = 0.06$ and in the decision space to $\delta_{var} = 0.02$.

3.4.2 Specification of Benchmark Problems

To assess the quality of the solutions for the proposed approaches, we tested the algorithms on a total of 20 different multi-modal multi-objective problem instances, including MMF1–15 (which are taken from the CEC 2019 competition on multi-modal multi-objective optimization [120, 101]), SYM–PART [121], as well as Omni-test problems [100] and polygon-based problems [66]. These multi-modal test problems have the important characteristic of every problem including multiple distinct subsets of the PS, each covering the PF independently. These test cases all fall under the first category of MMOPs, where there are multiple global PSs. To test the functionality of the proposed algorithms, each test case contains a range of complexity levels based on the number and shape of PSs and PFs. Additionally, we tested the algorithm's scalability in large-dimensional search spaces and objective spaces using the MMF14 and polygon test problems [122]. Considering the fact that this research area is relatively new, there are a limited number of test cases that have been conducted in the field. Our expectation is that over time, this number will increase. It is important to note that MMF1-14, SYM-PART, and Omni-test multi-modal multi-objective benchmark functions were implemented on the PlatEmo platform by the author. The polygon test problems are provided by the corresponding author of the proposed benchmark problems. Below are details about the benchmark problems used in this thesis:

- Benchmarks such as MMF1-MMF-13, MMF1-z, SYM-PART, and Omni-test have two decision variables and two objective functions, and they cannot be scaled.

- Using the MMF14 benchmark problem, it is possible to increase the number of decision variables from two to six with two or three different objective functions, which produces 10 different benchmarks.

- Three instances of the Polygon benchmark suite are presented using variables of size two, four and six, with $m = 6$ being the number of objective functions.

Chapter 4

Exploratory and Preservative Methodologies

This chapter analyzes two different types of developed approaches to address multi-modal multi-objective optimization problems. Combining these two types of approaches enhances the ability of existing algorithms to deal with MMOPs. The first types of proposed approaches consist of development techniques that enable the algorithm to explore the search space in greater depth, identifying multiple distinct solutions within the search space simultaneously by modifying the tournament selection, reproduction, and mutation operators. Further details concerning this type of approach are provided in Section 4.1. The second type of proposed approach develops methods for preserving these diversely obtained solutions, which map to the same or similar solutions in the objective space, for transmission to the next generation using modifications to the environmental selection process. Furthermore, the proposed environmental selection procedures can be divided into two categories so that they can be used for solving multi-modal, multi-objective optimization problems involving multiple global Pareto sets (MMOPs of type I) as well as problems involving local and global Pareto sets that must be maintained (MMOPs of type II). In Section 4.2, the proposed approaches for dealing with MMOPs of type I are further discussed, while Section 4.3 details the proposed approaches for dealing with MMOPs of type II. The performance of each proposed method in this chapter was also evaluated in comparison with some of the most recent and prominent related algorithms and previous proposed methods in the chapter, following the introduction of each proposed approach.

The following descriptions of the proposed approaches and the experiments are taken from the original publications of the author of this thesis [123, 124, 119, 125].

4.1 Proposed exploratory methods

Using explorative methods, the algorithm can be assisted in exploring the search space more deeply by modifying the tournament selection, reproduction, and mutation operators, thereby identifying multiple distinct solutions within the search space at the same time. The exploratory method searches for the best possible solution in a multidimensional search space when solving multi-objective optimization problems. Searches can be conducted locally or globally. A local search method focuses only on improving nearby solutions, while a global search method considers the entire search landscape.

4.1.1 Tournament Selection Mechanism

The author of the thesis has previously published the proposed tournament selection in [119], and the main content of this section derives from that publication.

Basically, tournament selection is the process of selecting a group of individuals out of a generation that will survive and reproduce in the following generation [41]. In a binary tournament, the two participants are chosen at random, and the individual with the lower crowding value and front number is added to the parental population. In this repeatable process, N parental solutions are produced, which may lead to duplicate solutions. This increases computational costs by evaluating duplicated solutions that have already been evaluated and decreases the full potential for finding the optimal set of solutions based on N selected diverse solutions.

Additionally, tournament selection is typically based on dominance mechanisms and density measurements within the objective space, which tends to favor solutions that are not dominant and located within the separated area of the objective space. For the algorithm to discover multiple solutions, it must modify the classical tournament selection (such as the tournament selection used in NSGA-II) as needed.

To eliminate the likelihood of duplicate solutions being generated, we propose a new tournament selection mechanism that excludes duplicate solutions from selection and incorporates diversity-based metrics into the search space to select a mate pair from the population for each solution. We propose a new tournament selection mechanism where a mate pair from the population is selected for each solution by excluding it from selection. All offspring that are identical to their parents are subjected to a second mutation if there is no change after crossover and mutation. In this way, we prevent the emergence of identical solutions in a population by ensuring that offspring from a particular generation are not identical to their parents [119].

Furthermore, we suggest that diversity-based metrics be incorporated into the search space instead of objective metrics in order to improve mating selection. Through the use of this strategy, we intend to increase the likelihood of finding non-dominated solutions in areas of the search space that are not well explored and to identify individuals that contribute significant diversity to the search space.

As a means of estimating the average number of duplicated solutions produced in a binary tournament selection mechanism, we conducted an experiment with the NSGA-II algorithm (with standard tournament selection) on a test problem (MMF1) containing 100 individuals and 10,000 function evaluations. Our findings show that, on average, 4.145 of the offspring of each generation are identical to their parents [119].

The proposed mating selection process aims to select mating parents that possess a high degree of diversity in the search space and convergence toward the PF. Algorithm 2 provides the pseudocode for our proposed tournament selection method. To select a mate, each individual in a population is evaluated based on two criteria. First, there is the population front number Line 1), for which a lower number indicates a better convergence of the solution. Second, there is the harmonic average distance (HAD) value of the solution among the entire population (Line 2), with a higher value indicating a less congested area around the solution. Next, two random solutions are selected for each solution in the population, excluding the solution itself (Line 3). As the mate for the solution $P_i(t)$, the individual with the lowest front number and the

```
   Input: Population P(t)
   Output: Population of Parents Q(t)
1  Front ← fast-non-dominated-sort(P(t)) HAD ← harmonic-average-distance(P(t))
2  for i ← 1 to |P| do
3      Choose two solutions x and y randomly from P(t), excluding the individual P_i(t);
4      if Front_x < Front_y then
5          mate_i(t) ← Select x as a mate for P_i(t);
6      else if Front_x > Front_y then
7          mate_i(t) ← Select y as a mate for P_i(t);
8      else
9          mate_i(t) ← Select the one with the higher HAD value;
10     end
11     P'_i(t) ← Recombine(P_i(t),mate_i(t));
12     Q_i(t) ← Mutate(P'_i(t));
13     if Q_i(t)==P_i(t) || Q_i(t)==mate_i(t) then
14         Q_i(t) ← Mutate(P'_i(t));
15     else
16         Do nothing;
17     end
18 end
19 return Q(t)
```

Algorithm 2: ModifiedMatGenSelect($P(t)$) [119]

largest HAD is selected (Lines 5–11)[119]. The HAD value for the solution i and its l nearest neighbors can be calculated as follows:

$$HAD(i) = \frac{k}{\sum_{j=1}^{l} \frac{1}{d_{ij}}} \tag{4.1}$$

The d_{ij} are the Euclidean distances from the solution i and the solution j within its neighborhood, and k denotes the neighborhood size.

Following the selection of a mate for the solution $P_i(t)$, the genetic operators are applied to the solution $P_i(t)$ and its mate to create an offspring (Lines 12–13). A second chance is given to offspring if recombination and mutation do not result in changes (Line 15). This process prevents duplicate solutions from being produced (Lines 14–18) [119].

4.1.2 Neighborhood-based Mutation Operator

The approaches introduced in this section are derived from the publications of the author of this thesis, which appear in [126, 102].

Due to the fact that most population-based metaheuristics employ a mutation operator, extending this operator to include a deeper exploration of the search space can facilitate finding more diverse solutions to dealing with MMOPs. This is accomplished by proposing a neighborhood-based mutation operator as the author's second contribution to multi-modal, multi-objective optimization.

We propose a modification of this operator based on the neighborhood mutation concept proposed by Qu et al. [127] in order to make it more applicable to multi-modal, multi-objective optimization problems. There is a description of a neighborhood-based mutation (NBM) approach in Algorithm 3. As part of the proposed method, neighboring solutions are constructed, the mutation probability of each solution is multiplied by the number of times each solution has appeared in its neighborhood, and a polynomial mutation operator is applied to each solution based on its adjusted mutation probability.

The intention behind the proposed operator indicates that a solution appearing in a neighborhood with other solutions has a greater chance of being mutated, leading to a greater possibility of generating new solutions outside the crowded areas and escaping local optimization traps. In this way, we are able to explore the search space in a deeper way and generate multi-modal solutions.

Figure 4.1 presents a demonstration of the detection of the unexplored area in the search space using the neighborhood-based mutation operator. Per Figure 4.1, PS_1 and PS_2 are two Pareto sets of solutions. These solutions are mainly located adjacent to PS_1. Implementing the neighborhood-based mutation will likely facilitate the relocation of the most crowded solution, highlighted in red, toward PS_2 as well as facilitate further exploration of the search space.

The concept of neighborhoods has been widely utilized in evolutionary algorithms. Neighborhoods can generally be classified into two types: index-based neighborhoods and distance-based neighborhoods. Index-based neighborhood optimization is commonly employed when dealing with single global peak optimization problems. On the other hand, distance-based neighborhoods are

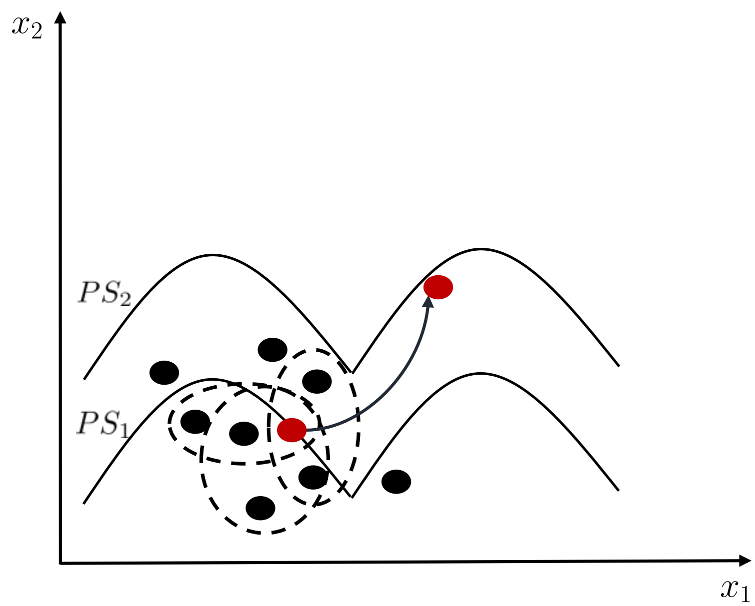


Figure 4.1: A neighborhood-based mutation illustration.

typically used in multi-modal optimization scenarios where the objective is to locate multiple optima and diversity is crucial [127]. To promote diversity within the population, we incorporated a distance-based neighborhood approach using the Euclidean distance metric in combination with the mutation operator. This integration aimed to enhance the algorithm's capacity to generate a wider range of diverse solutions.

In Algorithm 3, the Euclidean distances between all solutions in the decision space are first computed (Lines 1–4). Each solution's neighborhood consists of the individual itself and its K closest neighbors, as computed by the Euclidean distances (Lines 6–7). Next, polynomial mutations are used to mutate each individual in the population along with its neighboring solutions (Lines 9–26), and this operator is explained in more detail in the following chapter. The mutated offspring are then returned (Line 27).

In multi-objective evolutionary algorithms, the polynomial mutation operator is often used and has been shown to be effective [128]. It was originally proposed by Deb and Goyal [46, 129]. A polynomial distribution is used in this operator to sample a new value around the original value for real-valued variables. The polynomial mutation operator is composed of two parameters: the mutation probability p_m and the distribution index η_m. Specifically, a

Input: List O of offspring of solutions of current generation with $o := |O|$, Neighborhood Size$=K$
Probability of Mutation$=p_m$,
Distribution Index$=\eta_m$
Upper and lower bounds x_k^u and x_k^l for each variable k
Output: Mutated Individuals O

for $i \in \{1, .., o\}$ **do**
    for $j \in \{1, .., o\}$ **do**
        $Euc(i, j) = \|O[i].\vec{x} - O[j].\vec{x}\|_2$ //calculate Euclidean distances between solutions
    end
end
for $i \in \{1, .., o\}$ **do**
    $N(i)$ = list of indices of $K + 1$ smallest values in $Euc(i)$ //Set the neighborhood of each solution i as itself and its K nearest neighbors
end
for $i \in \{1, .., o\}$ **do**
    for $j \in N(i)$ **do**
        for $k \in \{1, .., n\}$ **do**
            $b = U(0, 1)$
            if $b \le p_m$ **then**
                $\delta_1 = \frac{O[j].x_k - x_k^l}{x_k^u - x_k^l}$
                $\delta_2 = \frac{x_k^u - O[j].x_k}{x_k^u - x_k^l}$
                $b = U(0, 1)$
                if $b \le 1/2$ **then**
                    $\delta_q = [(2b) + (1 - 2b)(1 - \delta_1)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}} - 1$
                else
                    $\delta_q = [1 - (2(1 - b)) + 2(b - 0.5)(1 - \delta_2)^{\eta_m + 1}]^{\frac{1}{\eta_m + 1}}$
                end
                $O[j].x_k += \delta_q.(x_k^u - x_k^l)$
            end
        end
    end
end
return O

Algorithm 3: NBMutation(O, K, p_m, η_m, x_k^u, x_k^l) [126]

probability parameter p_m is used for each variable $x_i \; \forall i = 1, ..., n$ within a solution $\vec{x}$ to determine whether that variable is subject to mutation. In this case, $n \cdot p_m$ represents the expected number of variables that will be mutated for each solution. It is often recommended that $p_m = \frac{1}{n}$ in the literature [33, 129].

Hence, in the expected situation, only one variable should be mutated per solution. As for the second parameter, the distribution index η_m determines the distribution from which the new, modified value is chosen. Distribution indexes η_m can be used to balance exploitation and exploration in the search process by influencing how much change a mutation operator produces. If the change is larger, it indicates that it is more likely that the drawn value from the distribution will be similar to the original x_i value. According to the literature, η_m is often set to 20 [33, 129].

Using this mutation operator implies that a neighboring solution, which appears in the neighborhood of many solutions, has the chance to be mutated more often than other solutions. In this way, the solutions that are located in crowded areas of the search space have a higher chance of being mutated. As a result, this might lead to better exploration in the decision space [126].

4.2 Proposed Preservative Methods

Using modifications to the environmental selection process, the second type of proposed approach focuses on developing methods for preserving these diverse solutions that are mapped to similar or the same solutions within the objective space to ensure that these diverse solutions are preserved for future generations. We present our proposed approaches in the following sections to deal with two types of problems that require different selection operators to solve. As the first category consists primarily of finding multiple global PSs, the environmental selection process is required to maintain the globally obtained optimal solutions. While the second category involves preserving both global and local optimal solutions, it is imperative to develop selection approaches that do not exclude locally identified optimal solutions from the search.

4.2.1 Weighted sum crowding distance density measurement approach

Throughout this section, the main content is drawn from the author's publications [126, 102].

To improve diversity in the objective space, the commonly used approach is to employ the classical crowding distance (CD) method, which enhances the distribution of solutions [33]. However, having a well-dispersed population in the objective space does not necessarily guarantee a high level of diversity in the decision space. To address this issue, similar to the Omni-optimizer algorithm [100], we incorporated the CD method in the search space to maintain solution diversity. Additionally, we devised a strategy called weighted sum crowding distance (WSCD), where we combined the crowding distance values from both the decision and objective spaces. This strategy aimed to achieve a more balanced distribution of solutions in both spaces.

Based on the proposed WSCD approach, CD is calculated in the objective space (CD_{obj}) in a similar manner to the NSGA-II algorithm, for which the pseudocode is shown in Algorithm 4. There is a large CD value (infinity) assigned to the extreme solutions in the objective space (Lines 1–4). For the remaining solutions, CD values are calculated by summing up the normalized distances between the neighbors on the left and right sides of each solution in the objective space (Lines 8–14) [33].

A description of the WSCD method can be found in Algorithm 5. The proposed approach begins by calculating the maximum and minimum values of all solutions for each decision variable (Lines 2–5). For each solution, the CD values in decision space and WSCD values are set to zero (Lines 6–9). After that, the solutions are sorted based on the values of the decision variables for each variable (Line 11). Using the normalized distance between the boundary solution and its adjacent neighbor, CD values are calculated for boundary solutions (Lines 12–13). By normalizing the distances between the left- and right-side neighbors of the solutions in the decision space, the CD values for the remainder of the solutions can be determined (Line 14). In our work, we normalize CD values in the decision space (CD_{dec}) and objective space (CD_{obj}), utilizing the max-min normalization method, so that the scores of CD values can be compared across different dimensions in the decision and objective spaces (Lines 19–20). Our objective was to ensure an adequate diversity of solutions in both the decision and objective spaces by assigning a final weighted sum for CD based on the assigned weights w1 and w2 for the CD in both the decision and objective spaces.

```
Input: List P of non-dominated solutions with p := |P|,
Number of Objectives M
Output: List P with added Crowding Distance (CD) values for each
        solution
for i ∈ {1, .., m} do
    f_{i,max} = maximum of values for i-th objective in P
    f_{i,min} = minimum of values for i-th objective in P
end
for j ∈ {1, .., p} do
    P[j].CD = 0 //initialize CD of j-th solution in P
end
for i ∈ {1, .., m} do
    P' = sort P ascending based on i-th objective
    P'[1].CD = ∞
    P'[p].CD = ∞
    for j ∈ {2, .., p − 1} do
        P'[j].CD += (P'[j+1].f_i − P'[j−1].f_i) / (f_{i,max} − f_{i,min})
    end
end
return P
```

Algorithm 4: CrowdingDistanceObj(P, m)- Pseudocode based on [126].

4.2.1.1 Configuration of the Proposed Methods

We evaluated the effectiveness of the approaches proposed in Sections 4.2.1 and 4.1.2 by implementing these approaches separately on NSGA-II algorithms, which are called NSGA-II-weighted-sum-crowding-distance (NSGA-II-WSCD) and NSGA-II-neigborhood-based mutation (NSGA-II-NBM). We combined NBM and WSCD approaches to capture and preserve the solutions discovered and implemented them on the NSGA-II using an algorithm we called NSGA-II-weighted-sum-crowding-distance-neigborhood-based mutation (NSGA-II-WSCD-NBM) Mutation (NSGA-II-NBM). We compared the various configurations of these approaches with each other, as well as the NSGA-II algorithm (as the baseline algorithm) and the Mo-Ring-PSO-SCD algorithm (as the state-of- the-art algorithm).

Input: List S of non-dominated solutions with added Crowding Distance (CD_{obj}) values for each solution in objective space according to NSGA-II algorithm with $s := |S|$, ; `// Algorithm 4`
, Number of Objectives: m, Number of Decision Variables
: n
Output: List S with added Weighted Sum Crowding Distance (CD_{ws}) values for each solution

for $i \in \{1, .., n\}$ **do**
  $x_{i,max}$ = maximum of values for i-th decision variable in S
  $x_{i,min}$ = minimum of values for i-th decision variable in S
end
for $j \in \{1, .., s\}$ **do**
  $S[j].CD_{dec} = 0$ //initialize CD_{dec} of j-th solution in S
  $S[j].CD_{WS} = 0$ //initialize CD_{WS} of j-th solution in S
end
for $i \in \{1, .., n\}$ **do**
  S' = sort S ascending based on i-th decision variable
  $S'[1].CD_{dec} += 2 \cdot \frac{|S'[j+1].x_i - S'[j].x_i|}{|x_{i,max} - x_{i,min}|}$
  $S'[s].CD_{dec} += 2 \cdot \frac{|S'[j].x_i - S'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
  for $j \in \{2, .., s-1\}$ **do**
    $S'[j].CD_{dec} += \frac{|S'[j+1].x_i - S'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
  end
end
for $j \in \{1, .., s\}$ **do**
  $S[j].CD.obj = norm(S[j].CD_{obj})$ //normalize CD_{obj} of j-th solution using max and min values of CD_{obj} in S
  $S[j].CD.dec = norm(S[j].CD_{dec})$ //normalize CD_{dec} of j-th solution using max and min values of CD_{dec} in S
  $S[j].CD_{WS} = w_1 \cdot S[j].CD_{dec} + w_2 \cdot S[j].CD_{obj}$;
end
return S

Algorithm 5: WSCD(CD_{obj}, m, n) - Pseudocode based on [126].

The following list is a description of the configuration of the proposed approaches.

- As explained, the proposed neighborhood-based mutation approach has been incorporated into the polynomial mutation, the configuration of which is described in the section 3.4 on foundation for experimental comparison setup. When a neighborhood-based mutation is performed, there is a single parameter K, which represents the size of the neighborhood. The number of individuals selected for each neighborhood area of the

Table 4.1: Analysis of performance using the IGDx indicator on the MMF1–MMF6 benchmarks using 10,000 function evaluations. Comparison of WSCD and NBM approaches conducted using NSGA-II. The best-performing algorithms are highlighted in bold and gray shading, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns [126].

Problems	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	**0.06321 (1.817E-3)**	0.0.07552 * (9.316E-3)	0.07923 * (7.412E-3)	0.0.07235 * (7.26E-3)	0.1051 * (151E-2)
MMF2	**0.01688 (2.885E-3)**	0.01771 (3.956E-3)	0.08949 * (7.2725E-2)	0.0.03088 * (8.32E-3)	0.1021 * (853E-2)
MMF3	**0.01486 (1.309E-3)**	0.015017 (2.318E-3)	0.0.05839 * (3.49894E-2)	0.02478 * (5.73E-2)	0.07854 * (314E-2)
MMF4	**0.01486 (1.309E-3)**	0.015017 * (2.318E-3)	0.0.05839 * (3.49894E-2)	0.02478 * (5.73E-2)	0.07854 * (314E-2)
MMF5	**0.01486 (1.309E-3)**	0.015017 * (2.318E-3)	0.0.05839 * (3.49894E-2)	0.02478 * (5.73E-2)	0.07854 * (314E-2)
MMF6	**0.01486 (1.309E-3)**	0.015017 (2.318E-3)	0.0.05839 * (3.49894E-2)	0.02478 * (5.73E-2)	0.07854 * (314E-2)
Best/All	6/6	0/6	0/6	0/6	0/6

solution can be controlled using this parameter. It is generally recommended that K be chosen between 1/20 and 1/5 of the population size [127]. A neighborhood's size can be made proportional to its population, which makes the choice of the neighborhood's size easy. As a result, our preliminary experiments suggest that the neighborhood size should be 1/5 of the population size, at which point it will reach its maximum performance.

- The weights for crowding distances in the objective space and in the decision space are evenly divided in both WSCD variations (NSGA-II-WSCD-NBM and NSGA-II-WSCD) as $w_1 = 0.5$ and $w_2 = 0.5$.

4.2.1.2 Discussion of Results

The following paragraphs were extracted from the original paper by the author [102]. In order to assess the effectiveness of the proposed algorithms in solving MMOPs, we applied these algorithms to the MMF1–MMF6 benchmark problems. The experimental results, presented in Tables 4.1, 4.2, and **??**, compare the performance of the algorithms based on the IGDx, IGD$^+$, and PSP indicators, respectively. Lower IGDx and IGD$^+$ values and higher PSP values indicate better performance [126].

It is concluded from the analysis of Tables 4.1, and 4.2 that when both explorative and preservative approaches are implemented together on the NSGA-II

Table 4.2: Analysis of performance using the PSP indicator on the MMF1–MMF6 benchmarks using 10,000 function evaluations. Comparison of WSCD and NBM approaches conducted using NSGA-II. The best-performing algorithms are highlighted in bold and gray shading, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns [126].

Problems	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	**15.70245 (5.23527E-1)**	13.14347 * (1.59755)	12.50492 * (1.24596)	13.50486 * (1.37166)	9.29939 * (1.41609)
MMF2	**59.21458 (9.79915)**	56.24433 (12.59192)	9.36013 * (7.80405)	30.63185 * (9.39582)	9.42434 * (8.32393)
MMF3	**67.17031 (5.65764)**	66.58847 (9.97048)	14.95064 * (11.64698)	38.33345 * (9.46867)	12.73248 (0.0314)
MMF4	**23.97306 (2.88038)**	16.89303 * (3.19544)	17.15216 * (3.1857)	21.85007 * (2.94561)	8.20806 * (3.14241)
MMF5	**8.72255 (3.8104E-1)**	7.66519 * (9.6644E-1)	6.8738 * (5.9585-1)	7.95066 * (5.7642E-1)	4.89867 * (1.33815)
MMF6	**10.03783 (6.3865E-1)**	9.20888 * (9.6738E-1)	7.92392 * (8.2706)	9.24543 * (8.2126E-1)	5.14445 * (8.2706E-1)
Best/All	6/6	0/6	0/6	0/6	0/6

Table 4.3: Analysis of performance using the IGD$^+$ indicator on the MMF1–MMF6 benchmarks using 10,000 function evaluations. Comparison of WSCD and NBM approaches conducted using NSGA-II. The best-performing algorithms are highlighted in bold and gray shading, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns [126].

Problems	NSGA-II-WSCD-NBM	NSGA-II-NBM	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	5.53E-3 * (6.2E-4)	**4.6E-3 (1.2E-4)**	5.441E-3 * (3.22E-4)	6.49E-3 * (7.6e-4)	5.32E-3 * (2.6E-4)
MMF2	1.455E-2 * (2.516E-3)	**1.4452E-2 (2.497E-3)**	1.6955E-2 * (1.4602E-2)	1.877E-2 * (5.89E-3)	1.995E-2 (1.253E-2)
MMF3	1.2298 E-2 * (2.12E-3)	**1.098E-2 (2.007E-3)**	1.527E-2 * (1.3291E-2)	1,656E-2 * (0.00485)	1.497E-2 (9.72E-3)
MMF4	5.347E-3 * (7.6E-4)	**4.762E-2 (2.4E-4)**	5.425E-3 * (2.5E-4)	7.02E-3 * (9.2E-4)	5.17E-3 * (1.9E-4)
MMF5	5.37E-3 * (3.9E-4)	**4.6E-3 (1.7E-4)**	5.59E-3 * (3.2E-4)	6.52E-3 * (5.3E-4)	5.34E-3 * (3.2E-4)
MMF6	5.43E-3 (4.57E-4)	**4.59 E-3 (2.01E-4)**	5.49E-3 * (2.83E-4)	6.43E-3 * (7.5E-4)	5.31E-3 * (2.6E-4)
Best/All	0/6	6/6	0/6	0/6	0/6

algorithm, which is referred to as the NSGA-II-WSCD-NBM algorithm, it outperforms the NSGA-II-NBM algorithm in all the test cases. Compared with the results of the other algorithms, it also shows significant superiority in all test problems. In other words, combining both exploratory and preservative approaches will lead to a better approximation of PS in terms of diversity and convergence [102].

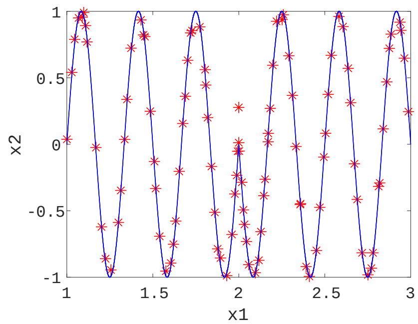
To analyze the performance in the objective space, Table 4.3 shows the IGD$^+$ values for the different algorithms. As can be observed from the results, NSGA-II-NBM obtains a better IGD$^+$ value than the others, while both the NSGA-II and NSGA-II-WSCD-NBM algorithms gain IGD$^+$ values similar to each other. It can also be observed that the proposed NSGA-II-WSCD-NBM algorithm provides significantly superior performance over both the original NSGA-II and the state-of-the-art Mo-Ring-PSO-SCD methods. In terms of IGDx, the proposed NSGA-II-WSCD-NBM algorithm provided significantly better results on all six test problems compared to the previous two algorithms from the literature. On all but one of the benchmarks used, NSGA-II-NBM outperforms the state-of-the-art in the objective space, as measured by the IGD^+ performance indicator.

According to our analysis of the results, WSCD variants result in the preservation of distinct solutions with the same objective function values. Therefore, the NSGA-II-WSCD shows improvement compared to the NSGA-II in terms of the decision-space-related metric. Furthermore, it is also assumed that neighborhood mutation contributes to the discovery of more Pareto-optimal solutions during the search as a result of boosting the proportion of solutions within the search process [102].

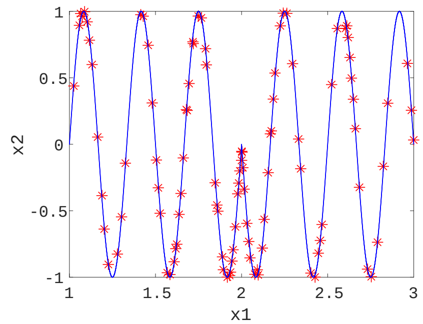
To better understand the similarity between the obtained solutions in both the decision and objective spaces, we present the obtained solutions for the NSGA-II-WSCD-NBM, NSGA-II-WSCD, NSGA-II-NBM, and Mo-Ring-PSO-SCD in Figures 4.2 and 4.3. In the figures below, we display the runs that achieved the median IGDx indicator for each algorithm [102].

As an example, in Figure 4.3 we illustrate the obtained solutions in the decision space for the MMF3 problem of the algorithms. The same is shown for the objective space. In the objective space, all algorithms obtain an evenly distributed solution set along the PF. However, when we take a closer look at the decision space, we notice that there are differences. Based on Figures 4.2 and 4.3, it is evident that the obtained solutions in decision space for NSGA-II-WSCD-NBM are evenly distributed along the PS while covering a greater number of points in each of its subsets. This is due to the fact that both the NBM and WSCD methods can assist the algorithm in identifying and maintaining the optimal solutions captured in the decision space.

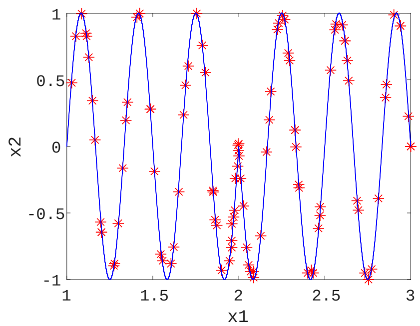
It has been observed that the solutions obtained in NSGA-II-NBM tend to fall into one of the subsets. In this case, this algorithm would not be able to preserve the solutions in different subsets since the accd is only used in objective space. Although the solutions in decision space are distributed over



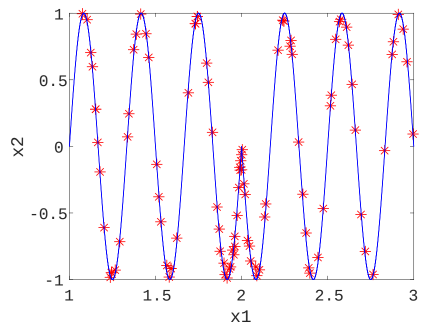
(a) PS for NSGA-II-WSCD-NBM



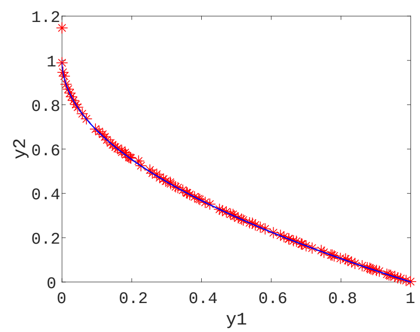
(b) PS for NSGA-II-NBM



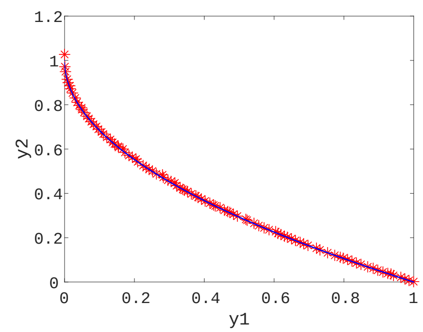
(c) PS for NSGA-II-WSCD



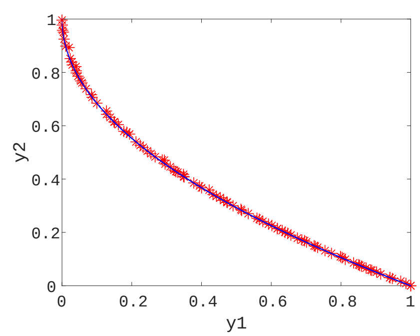
(d) PS for Mo-Ring-PSO-SCD



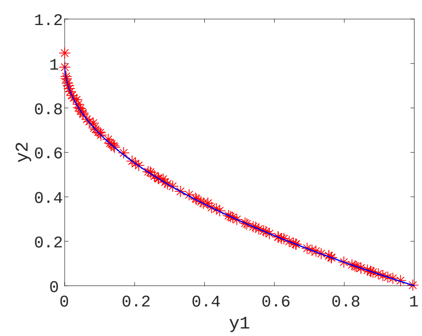
(e) PF for NSGA-II-WSCD-NBM



(f) PF for NSGA-II-NBM

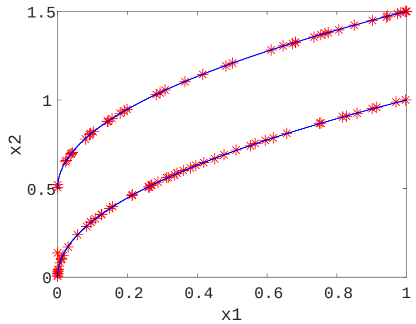


(g) PF for NSGA-II-WSCD

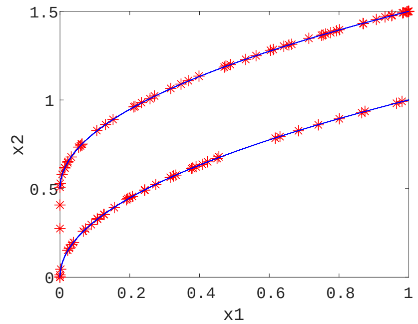


(h) PF for Mo-Ring-PSO-SCD

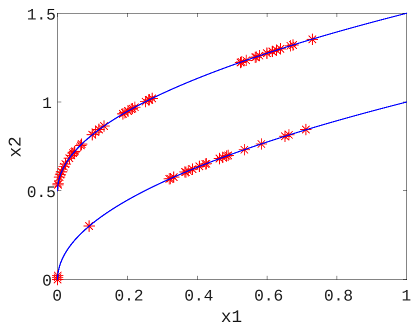
Figure 4.2: Obtained solutions in decision and objective space for MMF1 test problem [102]



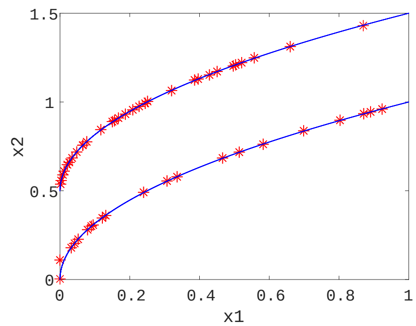
(a) PS for NSGA-II-WSCD-NBM



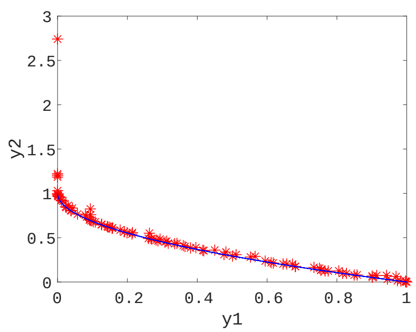
(b) PS for NSGA-II-NBM



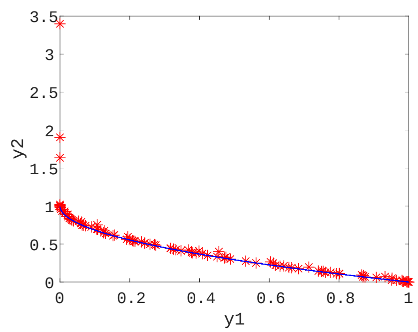
(c) PS for NSGA-II-WSCD



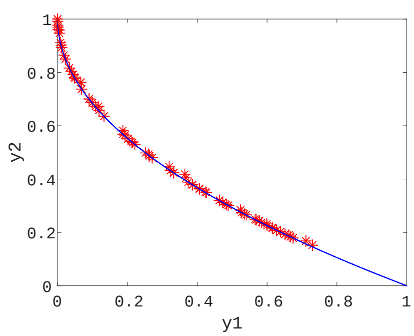
(d) PS for Mo-Ring-PSO-SCD



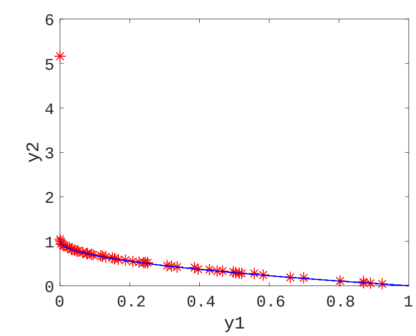
(e) PF for NSGA-II-WSCD-NBM



(f) PF for NSGA-II-NBM



(g) PF for NSGA-II-WSCD



(h) PF for Mo-Ring-PSO-SCD

Figure 4.3: Obtained solutions in decision and objective space for MMF3 test problem [102]

all equivalent subsets of the PS in the NSGA-II-WSCD algorithm, an even distribution is still lacking (Figures 4.1[c] and 4.2[c]). Accordingly, we conclude that NSGA-II-WSCD, which utilizes accd in the decision space, maintains the majority of the current determined solutions in the decision space. Due to the lack of a neighborhood mutation process, it was not possible to find most of the solutions over the PS.

The results of the Mo-Ring-PSO-SCD shown in Figures 4.2 and 4.3 also reveal that the PS could not be fully covered by the algorithm and that the solutions were not evenly distributed along the PS.

4.2.1.3 Influence of the Weight Values in the Weighed Sum Crowding Distance Approach

Our study also examines the impact of weight values (w) in different variants of the WSCD approach, in addition to evaluating its overall performance. Our preliminary findings indicate that increasing the weight value in either the objective space or decision space improves the distribution of solutions in the respective space, but it leads to a degradation in the distribution of solutions in the other space. Figure 4.4 presents a comparison of the performance of NSGA-II-WSCD for different weight values. The horizontal axis represents various weight values for the CD in the decision space (ranging from 1 to 0), while the vertical axis shows the corresponding values for IGDx, PSP, and IGD^+ based on the assigned weights [126].

The performance of the NSGA-II-WSCD algorithm, as shown in Figures 4.4(a) and 4.4(c), is sensitive to the weight vectors on multi-modal, multi-objective test problems. We expected that decreasing the CD in decision space would have a negative effect on the accuracy of the NSGA-II-WSCD algorithm when approximating the PS in decision space. In contrast, by increasing the weight value in objective space, we are able to obtain a better estimate of PF. According to Figure 4.4(b), the PSP values obtained support the idea that as the weight of the CD increases in decision space, the diversity of approximations of PS will generally increase [126].

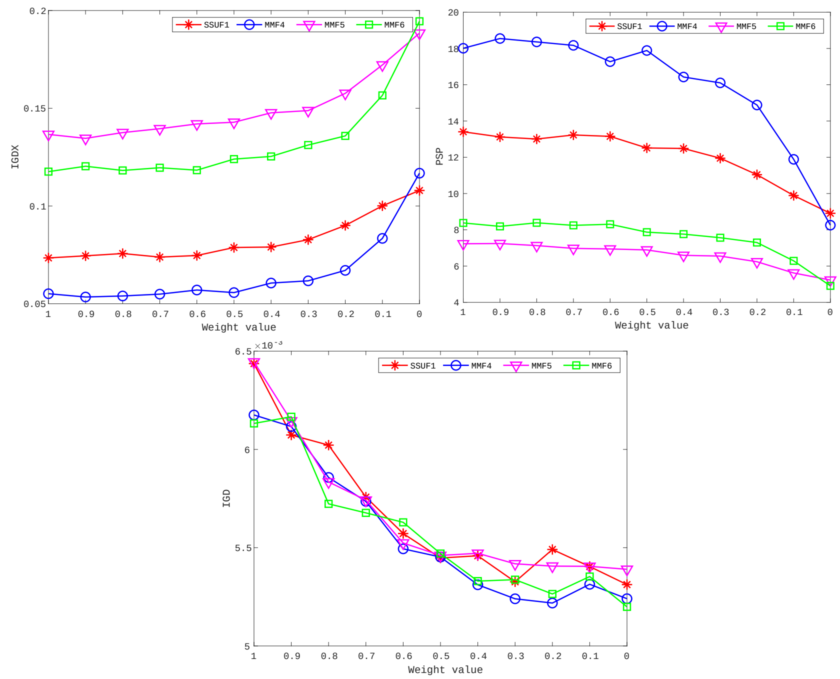


Figure 4.4: Achieved (a) IGDx values, (b) PSP values and (c) IGD$^+$ values by NSGA-II-WSCD using different weight values for the crowding distance in the decision space [126].

4.2.1.4 Influence of the Population Size in the Weighed Sum Crowding Distance Approach

Moreover, we examine the impact of population size on the performance of the NSGA-II-WSCD-NBM algorithm. In most algorithms, increasing the population size results in better approximations of optimal solutions [130]. However, increasing the population size also comes with an increase in computational costs [126]. We performed experiments with different population sizes for the NSGA-II-WSCD-NBM algorithm on the six different test problems to evaluate the effects of the population size on the approximation of the PS and PF. The number of function evaluations is set to 10,000 and the used population sizes are 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000 [126]. To make the trade-off between both approximations of optimal solutions in decision and objective spaces, we equally divided the weight values in both decision and objective spaces [126].

The results of these experiments are shown in Figure 4.5, where the median IGDx, PSP, and IGD$^+$ values of the experiments based on 31 independent runs are shown on the vertical axis [126]. As expected, we observe in Figures

4.5(a) and 4.5(b) that larger population sizes lead to a higher probability of locating more diverse solutions. Therefore, the algorithm provides better approximations of the PS with larger population sizes. However, as we can observe in Figure 4.5.(c) the rate of improvement regarding the IGD$^+$ indicator decreases for larger population sizes. This shows that locating more solutions in the decision space does not guarantee well-distributed solutions in the objective space [126].

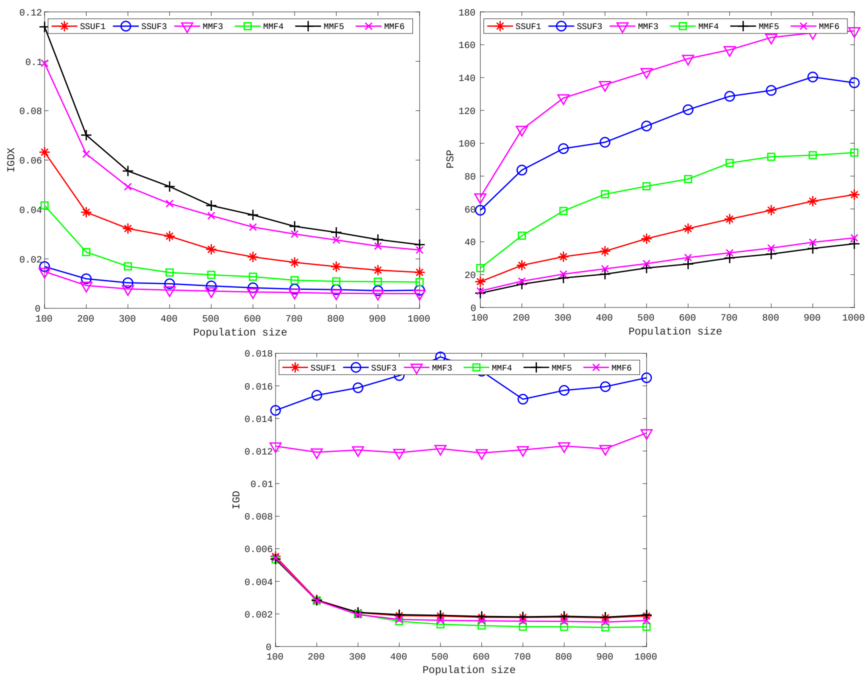


Figure 4.5: A comparison of the effects of population size differences on the median (from left to right) values of the (a)IGDx, (b)PSP, (c)and IGD$^+$ using NSGA-II-WSCD algorithm [126].

4.2.2 Manhattan Distance-based Density Measurement Approach

This section contains content from the publication by the author of this thesis in [124, 126].

To enhance density estimation in the environmental selection process, this section introduces another distance-based density measurement technique based on the Manhattan distance metric calculated in the decision space. We used the Manhattan distance metric (also known as the p_1 metric) to compute the distances between solutions using the proposed method. This was due to the natural capability of grids to represent the distribution of solutions.

Our method calculates the Manhattan distance between each solution and all other solutions on the current front. After summing all of these distances between each solution and the rest of the solutions, our global Manhattan distance metric is calculated as follows:

$$MD_{global}(\vec{a}) = \sum_{p \in P} \|\vec{a} - \vec{p}\| = \sum_{p \in P} \sum_{i=1}^{n} |a_i - p_i| \tag{4.2}$$

where P is the current front of solutions, n is the dimension of decision variables, and a_i and p_i represent the grid index values of solutions $\vec{a}$ and $\vec{p}$ in dimension i.

4.2.2.1 Combination of Manhattan and Crowding Distances in the Search Space

With the aim of enhancing the diversity of solutions, we multiplied the Manhattan distance metric value in decision space with its CD value (as defined in [100]), in which this distance only takes into account its nearest neighbor for boundary definitions.

Figure 4.6 illustrates how Manhattan distance and CD distance measurements, when combined (which we refer to as the MDCD diversity measurement approach), can lead to a wide range of solutions in decision space.

In Figure 4.6, the global Manhattan distance values for the solutions S_1 and S_2 are both equal to 20. Both of the solutions are located far from the rest of the solutions, and both provide good coverage of solutions in the decision space. In this example, solution S_1 is located in a more crowded neighborhood area than solution S_2. Therefore, the CD value for S_1 is smaller than for the other solutions. By multiplying both Manhattan and CD values, S_2 obtains a larger value than the other solutions. Therefore, we could guarantee a better diversity estimation of solutions by combining both distance metrics [124].

In algorithm 6, we present our proposed method (NSGA-II-MDCD). We modify NSGA-II by replacing the CD with our MDCD metric. First, we calculate the global Manhattan distance value (Lines 1–12). Then, the CD value for all solutions in the decision space is calculated (Lines 8–21). The final MDCD value for each solution is calculated by multiplying the two distances [124].

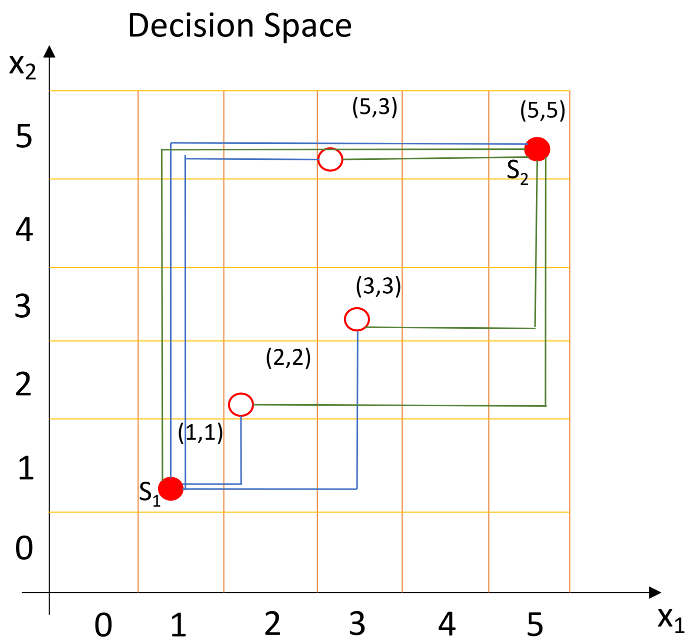


Figure 4.6: An example of the computation of MDCD, and its influence on the diversity of solutions in the decision space [124].

4.2.2.2 Configuration of the Proposed Method

For the purposes of evaluating the performance of the proposed approach, the approach is implemented on the NSGA-II algorithm as an example, which is called the NSGA-II-MDCD algorithm. The configuration of the proposed algorithm is as follows:

- As implemented in the previous chapter on the distributions of weights, we determined the WSCD by dividing all weights equally in both the decision and objective spaces.
- To determine the optimal grid size for MDCD, an experiment was conducted that compared the performance of different grid sizes, which are presented in the following sections. Based on the results of the experiment, a grid size of 10 was selected as the optimal grid size.

4.2.2.3 Discussion of results

Experimental comparisons were conducted with 1, 5, 10, 15, 20, 25, and 30 grids to examine the impact of grid size on the performance of the proposed algorithm

Input: Number of Objective functions: M,
Number of Decision Variables: n,
List P of solutions of current front (with GridIndex values for each dimension), of size $p = |P|$
Output: List P of solutions of current front with extra property of Combined Manhattan Distance and CD (MDCD) for each solution

for $j \in \{1, .., p\}$ **do**
  $P[j].MD_{global} = 0$
  $P[j].CD_{dec} = 0$
  $P[j].MDCD = 0$
end
for $i \in \{1, .., p\}$ **do**
  for $j \in \{1, .., p\}$ **do**
    for $k \in \{1, .., n\}$ **do**
      $P[i].MD_{global} += P[i].GridIndex(k) - P[j].GridIndex(k))$
    end
  end
end
for $i \in \{1, .., n\}$ **do**
  $x_{i,min}$ = minimum of values for i-th decision variable in P
  $x_{i,max}$ = maximum of values for i-th decision variable in P
end
for $i \in \{1, .., n\}$ **do**
  P' = sort P ascending based on i-th decision variable
  $P'[1].CD_{dec} += 2 \cdot \frac{|P'[j+1].x_i - P'[j].x_i|}{|x_{i,max} - x_{i,min}|}$
  $P'[p].CD_{dec} += 2 \cdot \frac{|P'[j].x_i - P'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
  for $j \in \{2, .., p-1\}$ **do**
    $P'[j].CD_{dec} += \frac{|P'[j+1].x_i - P'[j-1].x_i|}{|x_{i,max} - x_{i,min}|}$
  end
end
for $i \in \{1, .., p\}$ **do**
  $P[j].MDCD = P[j].CD_{dec} \cdot P[j].MD_{global}$
end
return P

Algorithm 6: Combined Manhattan Distance and Crowding distance Approach (MDCD) .Pseudocode based on [124].

(i.e., NSGA-II-MDCD). We perform this experiment with a population size of 100 and parameter values as described in the section of foundation for experimental comparison 3.4. An analysis of the grid size comparison is presented in Figure 4.7.

Increasing the size of the grids from one to five; results in a decrease in IGDx and IGD values and a dramatic increase in PSP values, as shown in Figures (1) and (2). In contrast, increasing the grid size from five to 30 does not significantly affect IGD and IGDx on most problems. Although some changes can be noticed between different sizes, there are no clear increases or decreases when the grid size is increased. Those observations, which demonstrate that The proposed algorithm maintains a steady state behavior with an increase in grid size, which can be explained by the fact that with increasing grid size, the Manhattan distance between all possible solutions in decision space increases. Consequently, when the Manhattan distance and the crowding distance are multiplied, the same solutions are selected for the selection process for the various grid sizes [124].

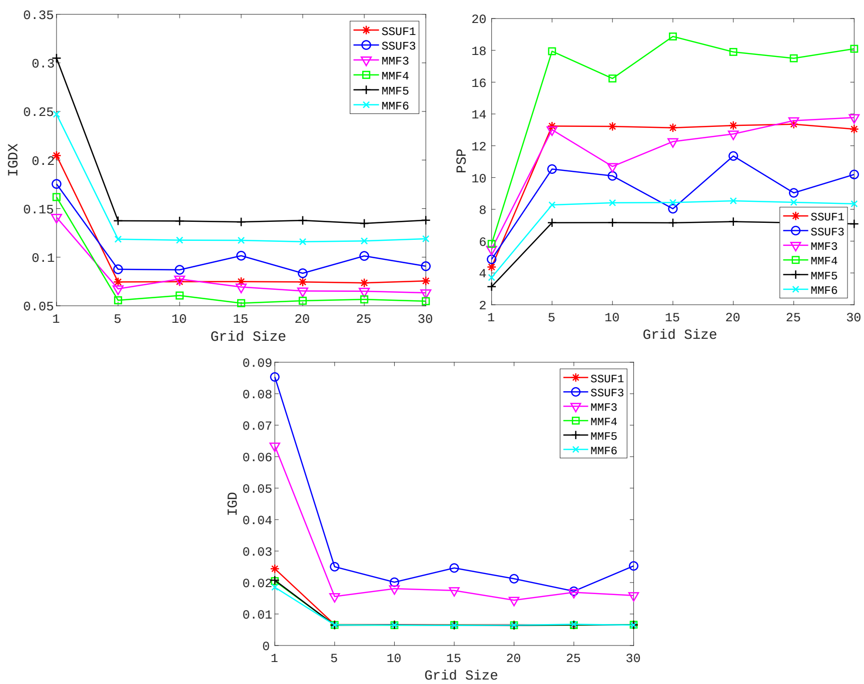


Figure 4.7: Achieved (a) IGDx values, (b) PSP values and (c) IGD values by NSGA-II-MDCD algorithm with different grid size [124].

The IGDx, IGD$^+$ results for the different algorithms compared are presented in Tables 4.4, 4.5 and 4.6. As can be observed in Tables 4.4, 4.5, NSGA-II-MDCD performs the best in terms of IGDx and PSP compared to the rest of the algorithms for four out of six test problems. Thus, the proposed algorithm provides a better distribution of solutions in the decision space. Although the NSGA-II-WSCD algorithm produced better results for MMF3 and MMF4 compared to the proposed method, no statistical significance was observed between these two algorithms. The observation indicates that in MMF3 and MMF4, optimal solutions are more concentrated in concrete grids, which results in a lower performance for the NSGA-II-MDCD algorithm compared to other problems where optimal solutions are distributed across a greater number of grids.

As expected from Table 4.6, the IGD$^+$ value of the NSGA-II algorithm shows its superiority in comparison with the proposed algorithm. This is because the main focus of the NSGA-II algorithm is to obtain a better diversity of solutions in objective space while neglecting decision space; therefore, a lower IGD$^+$ value is expected. With further analysis of the results, we could claim that the NSGA-II-MDCD algorithm provides a better approximation of PS while not disturbing the approximation of PF [124].

Table 4.4: Analysis of performance using the IGDx indicator on the MMF1-MMF6 benchmarks using 10,000 function evaluations. A performance evaluation of the MDCD approach was conducted using NSGA-II. The best performing algorithms are highlighted in bold and gray shaded, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	NSGA-II-MDCD	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	**0.07478 (0.00849)**	0.07923 * (7.412E-3)	0.0.07235 * (7.26E-3)	0.1051 * (151E-2)
MMF2	**0.08699 (0.072)**	0.08949 * (7.2725E-2)	0.0.03088 * (8.32E-3)	0.1021 * (853E-2)
MMF3	0.07747 (0.03521)	**0.0.05839 (3.49894E-2)**	0.02478 * (5.73E-2)	0.07854 * (314E-2)
MMF4	0.06053 (0.01059)	**0.0.05839 (3.49894E-2)**	0.02478 * (5.73E-2)	0.07854 * (314E-2)
MMF5	**0.13723 (0.01042)**	0.0.05839 * (3.49894E-2)	0.02478 * (5.73E-2)	0.07854 * (314E-2)
MMF6	**0.11752 (0.00682)**	0.0.05839 * (3.49894E-2)	0.02478 * (5.73E-2)	0.07854 * (314E-2)
Best/All	4/6	2/6	0/6	0/6

Moreover, the solutions obtained in the median execution for the three algorithms over the different datasets are represented in Figures 4.8 and 4.9 both in decision and objective spaces. In these figures, the true PS and PF are represented in blue to underline the clustered solutions. As seen in Figure 4.8, the solutions of NSGA-II-MDCD are more evenly distributed in decision space than the solutions of acnsga-ii-wscd and NSGA-II algorithms. As for objective space, the proposed algorithm still obtains a good approximation of the PF, but some parts of it are less crowded than others compared with NSGA-II [126].

Table 4.5: Analysis of performance using the PSP indicator on the MMF1-MMF6 benchmarks using 10,000 function evaluations. A performance evaluation of the MDCD approach was conducted using NSGA-II. The best performing algorithms are highlighted in bold and gray shaded, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns [102].

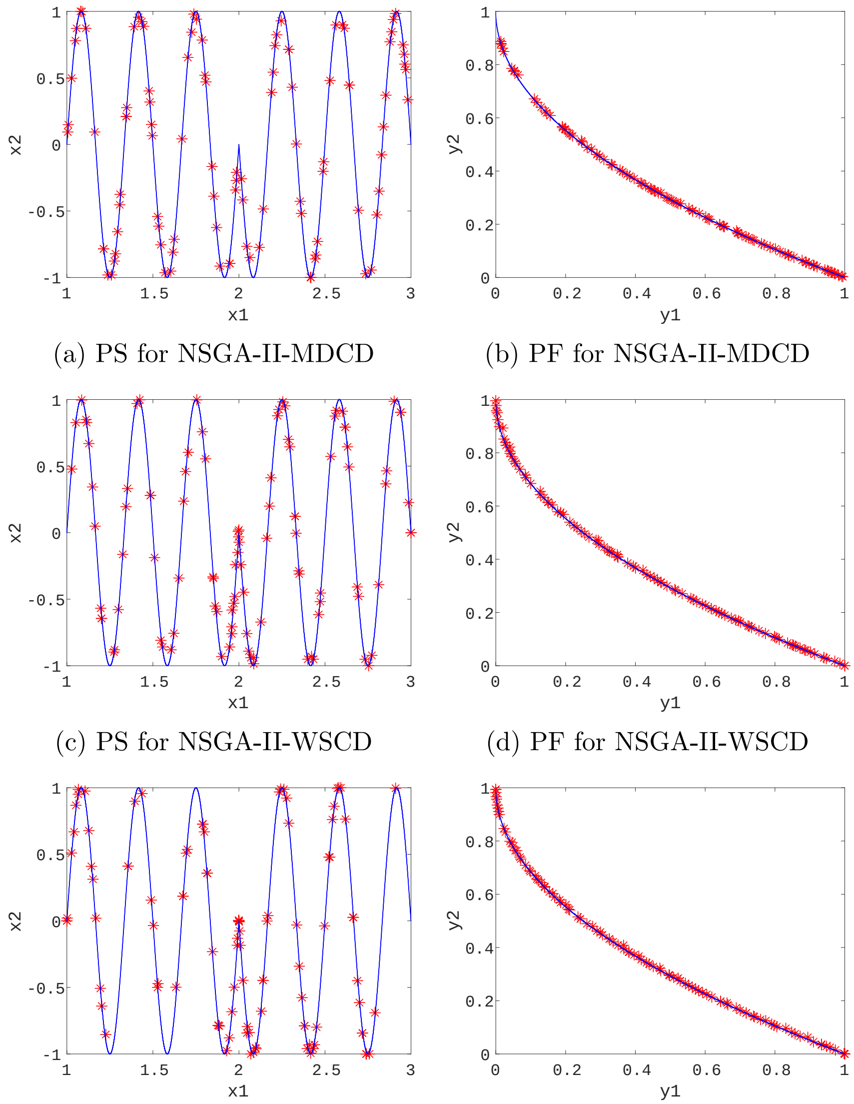
Problems	NSGA-II-MDCD	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	**13.2156 (1.48997)**	12.50492 * (1.24596)	13.50486 * (1.37166)	9.29939 * (1.41609)
MMF2	**10.10445 (7.29567)**	9.36013 * (7.80405)	30.63185 * (9.39582)	9.42434 * (8.32393)
MMF3	10.69778 (0.03521)	**14.95064 (11.64698)**	38.33345 * (9.46867)	12.73248 (0.0314)
MMF4	16.23044 (3.10691)	**17.15216 (3.1857)**	21.85007 * (2.94561)	8.20806 * (3.14241)
MMF5	**7.16681 (0.45601)**	6.8738 * (5.9585-1)	7.95066 * (5.7642E-1)	4.89867 * (1.33815)
MMF6	**8.41297 (0.53495)**	7.92392 * (8.2706)	9.24543 * (8.2126E-1)	5.14445 * (8.2706E-1)
Best/All	4/6	2/6	0/6	0/6

Table 4.6: Analysis of performance using the IGD$^+$ indicator on the MMF1-MMF6 test problems, using 10,000 function evaluations. A performance evaluation of the MDCD approach was conducted using NSGA-II. The best performing algorithms are highlighted in bold and gray shaded, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	NSGA-II-MDCD	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	0.00662 * (0.00053)	5.441E-3 (3.22E-4)	6.49E-3 * (7.6e-4)	**5.32E-3 2.6E-4)**
MMF2	0.02011 (0.02278)	1.6955E-2 (1.4602E-2)	1.877E-2 * (5.89E-3)	**1.995E-2 (1.253E-2)**
MMF3	0.01805 (0.01474)	1.527E-2 (1.3291E-2)	1,656E-2 * (0.00485)	**1.497E-2 (9.72E-3)**
MMF4	0.00645 * (0.00035)	5.425E-3 * (2.5E-4)	7.02E-3 * (9.2E-4)	**5.17E-3 (1.9E-4)**
MMF5	0.00655 * (0.00034)	5.59E-3 * (3.2E-4)	6.52E-3 * (5.3E-4)	**5.34E-3 (3.2E-4)**
MMF6	0.00647 * (0.0005)	5.49E-3 * (2.83E-4)	6.43E-3 * (7.5E-4)	**5.31E-3 (2.6E-4)**
Best/All	0/6	0/6	0/6	6/6

4.2.3 Grid Distance-based Density Measurement Approach

According to the algorithm proposed in the previous section, which we referred to as NSGA-II-MDCD, all the solutions in the current fronts directly influence the global Manhattan distance calculation for each solution. Thus, when calculating the Manhattan distance of a solution, if there is a solution located much further from this solution on the same front, it creates an illusion that



(a) PS for NSGA-II-MDCD (b) PF for NSGA-II-MDCD

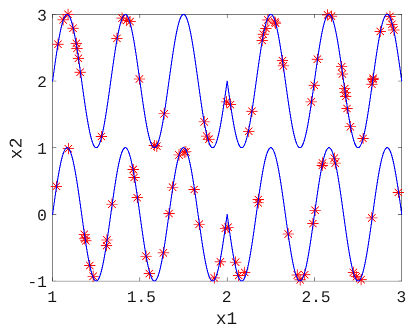
(c) PS for NSGA-II-WSCD (d) PF for NSGA-II-WSCD

(e) PS for NSGA-II (f) PF for NSGA-II

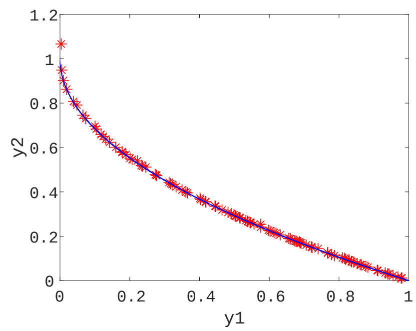
Figure 4.8: Obtained solutions in decision and objective space for MMF1 test problem [124]

the solution is located in a sparser area, even if it may be located in the neighborhood of a more populated area. By addressing this issue, we proposed an improved grid-based distance method (Gr_{dec}) that accounts for the effects of neighboring solutions on the density estimation for each solution. The main contents of this section are taken from the publication by the author in [65].

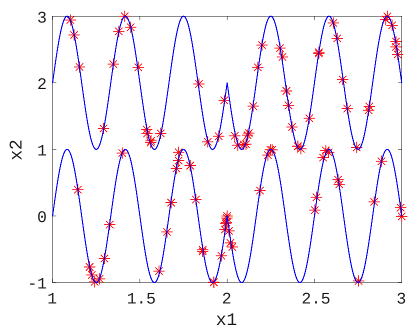
Calculating crowding distance values can assist in identifying crowded areas. However, it is also necessary to consider the density of solutions nearby each solution when computing the crowding distance values. Our proposed method



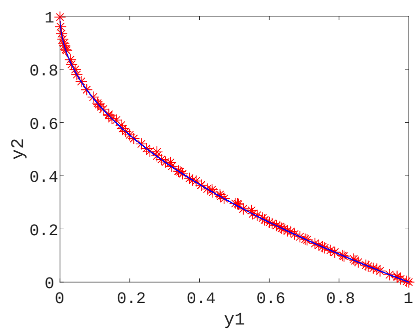
(a) PS for NSGA-II-MDCD



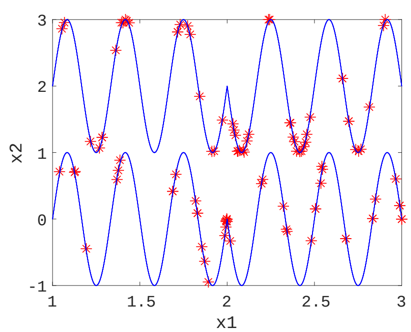
(b) PF for NSGA-II-MDCD



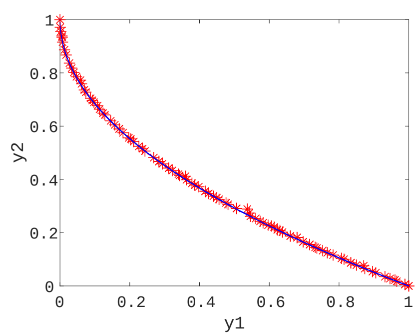
(c) PS for NSGA-II-WSCD



(d) PF for NSGA-II-WSCD



(e) PS for NSGA-II



(f) PF for NSGA-II

Figure 4.9: Obtained solutions in decision and objective space for MMF5 test problem [124]

was inspired by an algorithm called Grid-Based Evolutionary Algorithm (GrEA) [131], which divides solutions into grids in objective space, introducing a novel notion of neighborhood. To increase the selection pressure toward the PF, the GrEA algorithm partitions the solutions in the objective space and calculates the grid difference between the solutions, which improves the distribution of the solutions.

Since grids are naturally capable of expressing solution distributions, we adapted the neighborhood concept for the GrEA algorithm from objective space to decision space. To measure the distance between solutions in the search space, we partitioned the decision space into grids and developed a novel grid-based distance measurement mechanism. The neighborhood area for each solution S_i in the decision space is limited by a maximum grid difference of n (the number of decision variables). For each solution S_i, the neighborhood contains the solutions S_i with a grid difference less than n to S_i, which is formulated as follows:

$$NB(S_i) = \{S_j | S_i \neq S_j \wedge \ GD(S_i, S_j) < n\} \tag{4.3}$$

Where $S = (S_1, ..., S_N)$ is the current front of solutions and NB denotes the set of solutions that are in the neighborhood of solution S_i. $GD(S_i, S_j)$ is the grid distance between pairs of solutions S_i and S_j.

Our proposed grid-based crowding distance (Gr_{dec}) value for each solution is calculated using the following equation to favor solutions located in sparse areas:

$$Gr_{dec}(S_i) = \sum_{S_j \in NB(S_i)} (n - GD(S_i, S_j)) \tag{4.4}$$

In this manner, a solution located in a congested area will be assigned a larger Gr_{dec} value. The solution with smaller Gr_{dec} values is preferred in our proposed approach. The proposed method for grid-based crowding distance in decision space is described in Algorithm 7. In the first step of the calculation, after the parameters have been set up (Lines 1–6), the grid distance between each pair of solutions on the same front will be calculated (Lines 7–13). Following this, a grid-based crowding distance (Gr_{dec}) value is calculated for each solution based on Equations 4.3 and 4.4 (Lines 14 to 18).

Input: Number of decision variables: n,
List S of solutions of current front (with Grid index Gr_{Ind} values for each dimension)
Output: List S with the extra property Grid-based crowding distance in decision space (Gr_{dec}) for each solution

1 . **for** $i \in \{1, .., |S| - 1\}$ **do**
2     $S[i].Gr_{dec} = 0$;
3   **for** $j \in \{1, .., |S|\}$ **do**
4         $GD(i, j) = 0$;
5   **end**
6 **end**
7 **for** $i \in \{1, .., |S| - 1\}$ **do**
8   **for** $j \in \{i + 1, .., |S|\}$ **do**
9   **for** $k \in \{1, .., n\}$ **do**
10             $GD(i, j) += S[i].Gr_{Ind}[k] - S[j].Gr_{Ind}[k]$;
11             $GD(j, i) += S[j].Gr_{Ind}[k] - S[i].Gr_{Ind}[k]$;
12   **end**
13   **end**
14 **end**
15 **for** $i \in \{1, .., |S|\}$ **do**
16   **for** $j \in \{1, .., |S|\}$ **do**
17         $S[i].Gr_{dec} += max(n - GD(i, j), 0)$;
18   **end**
19 **end**
20 $S = norm_{Gr_{dec}}(S)$;   // normalization of Gr_{dec}
21 **return** S

Algorithm 7: Grid-based crowding distance in decision space (Gr_{dec}) approach. Pseudocode based on [65]

To normalize the grid-based crowding distance values (to combine them with other distance metrics), a max-min normalization is applied to ensure that the results fall within the same range (0,1] (Line 19). To avoid obtaining zero values for the solutions that have the minimum value, the min value in the equation should be substituted with a small value (e.g., 0.001) as follows:

$$norm(V_i) = \frac{V_i - (min(V) - 0.001)}{max(V) - (min(V) - 0.001)} \quad (4.5)$$

where $V = (V_1, ..., V_N)$ is a set of values (the Gr_{dec} values in this work) and $norm(V_i)$ is the i_{th} normalized value.

with the lower Gr_{dec} values are selected and transferred to the next generation. Solutions located in the vicinity of sparse areas are represented by the lower values.

To evaluate the performance of the grid-based crowding distance approach for approximation of PS, we have incorporated our proposed algorithm into the NSGA-II algorithm and replaced its environment selection with the proposed Grdec method, which we call the NSGA-II-Gr algorithm. To address this shortcoming, the proposed method Grdec is further improved by dividing the Grdec value for each solution by the Gr_{dec} value applied to the decision space (Gr_{dec}). This variant of the proposed algorithm is termed NSGA-II-Gr-CD_{dec}.

In spite of the fact that the proposed method (NSGA-II-Gr) is able to approximate the density of solutions, there are some cases in which this method fails to highlight solutions that are located in crowded areas. Due to the fact that the Gr_{dec} value of solutions is highly dependent on grid resolution, in some cases, even if solutions are not located in equally crowded areas, the Gr_{dec} values of some solutions can be the same. To address this shortcoming, the proposed method Gr is further improved by dividing the Gr_{dec} value for each solution by the CD_{dec} value applied to the decision space (CD_{dec}). This variant of the proposed algorithm is termed NSGA-II-Gr-CD_{dec}. To address this shortcoming, the proposed method Gr is further improved by dividing the Gr_{dec} value for each solution by the CD_{dec} value applied to the decision space (CD_{dec}). A description of this calculation can be found in Section 4.2.1. This variant of the proposed algorithm is termed NSGA-II-Gr-CD_{dec}.

A representative example of the importance of this combination can be found in Figure 4.10. Using this example, solutions D and E share the same Gr_{dec} values but have different values for CD_{dec}. As a result of dividing the Gr_{dec} values with the CD_{dec} values, solution E gets a smaller value than D in terms of GrCD_{dec} (i.e.$\mathrm{GrCD}_{dec} = \frac{\mathrm{Gr}_{dec}}{\mathrm{CD}_{dec}}$). The combination of these two variables indicates that solution E is located in a less crowded area than solution D.

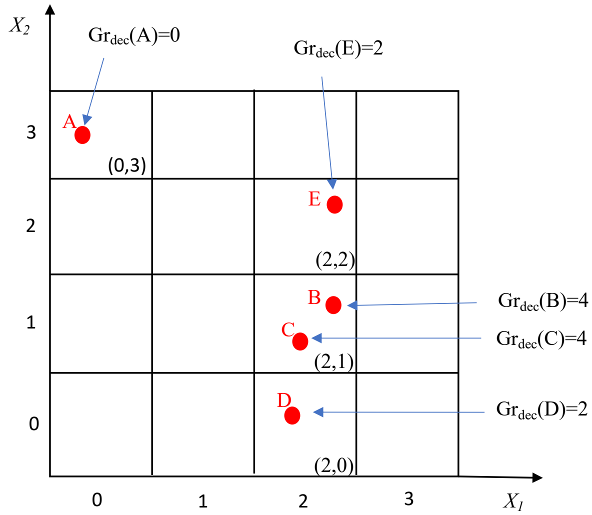


Figure 4.10: An instance to demonstrate the importance of combining the grid-based crowding distance method with the crowding distance in decision space [65].

The proposed Gr_{dec} method is further modified by two additional modifications. To retain the diversity of the solutions in the objective space, this method is combined with the usual CD approach used in NSGA-II ((CD_{obj}) [33]). This algorithm is referred to as NSGA-II-Gr-CDobj. In another variation, we attempt to take advantage of the crowding approach in both decision and objective spaces. By combining the Gr_{dec} approach with our proposed WSCD approach (Section 4.2.1), we introduce a new algorithm, NSGA-II-Gr-WSCD, which increases the diversity of the decision space without significantly reducing the distribution of the solutions in the objective space. Considering that lower Gr_{dec} values and higher CD_{dec} CD_{dec}, CD_{obj}, and CD_{ws} values are preferred, these modifications are proposed by dividing the calculated Grdec values with the respective crowding distance. Following this, the final distance values are sorted in ascending order, and the lowest values (representing solutions located in the vicinity of sparser areas) are transferred to the next generation.

4.2.3.1 Configuration of the Proposed Method

In preliminary experiments, we tested all the proposed algorithms on each test problem with varying grid sizes, and the results indicated that a grid size increase from 1 to 20 improved all algorithms. In evolutionary algorithms, the curve of improvement remains stable after grid size 20, with some fluctuation due to the stochastic nature of the algorithms. We therefore conducted all experiments with a 20-grid size.

4.2.3.2 Discussion of Results

The median value and interquartile range (IQR) of the corresponding IGDx, IGD^+, and PSP performance indicators are provided in Tables 4.7, 4.8, and 4.9. From the analysis of the results, it is evident that by combining the crowding distance approach with the grid-based approach, the proposed algorithms can improve their search power compared to the baseline algorithm and the presented state-of-the-art algorithms, which will lead to improved quality of the optimal solutions obtained in the decision space.

As seen with the results of Table 4.7, in all cases, the combination of the grid-based approach with crowding distance in both decision (NSGA-II-Gr-CDdec) and objective (NSGA-II-Gr-CDobj) spaces leads to an improvement in the quality of the optimal solutions obtained in decision space. In light of these results, it appears that there is no statistically significant difference between the two proposed algorithms for NSGA-II-Gr-CDdec and NSGA-II-Gr-CDobj. It is interesting to note that both algorithms performed the best in every test problem compared to other MMOEAs. The pairwise comparison between NSGA-II-Gr-CDdec and NSGA-II-CDobj demonstrated that the first algorithm outperformed the second in all test problems, as indicated by the IGDx and PSP values, which highlights the importance of applying Gr_{dec} to improve solution distributions in the decision space.

NSGA-II-Gr demonstrated better performance than the state-of-the-art algorithm MO-Ring-PSO-SCD in terms of IGDx and PSP values across all test problems. The PSP values indicate that both NSGA-II-Gr-CDdec and NSGA-II-Gr-CDws consistently achieved superior results in each test problem, effectively balancing the diversity and convergence of the obtained optimal solutions.

It is evident that the manipulation of the diversity of solutions in decision space may result in a deterioration of the diversity in objective space when dealing with MMOPs. There are several reasons why this can occur, including the fact that solutions that are added to the decision space are dominated by existing solutions or located near each other in the objective space, so they do not contribute to the diversity of the objective space. In the proposed Gr-WSCD method, a trade-off is made between the distribution of solutions in these two spaces. Per Table 4.9, the improvement in the IGDx value is much greater than the deterioration of the IGD^+ value for the proposed methods. For example, the improvement in IGDx value for the MMF5 and MMF8 test problems on NSGA-II-Gr-CDws over NSGA-II-CDws is approximately 1.2e-02 and 1.05e-02, respectively. In contrast, the IGD+ value decreased by 8.36e-04 and 2.97e-04. These findings demonstrate that the proposed algorithms are successful in making a reasonable compromise between the diversity of decision spaces and objective spaces. Taking a closer look at Tables 4.7, 4.8, it is evident that Gr_{dec} methods contribute to the distribution of solutions in the decision space for the most complex test case (such as MMF1z with asymmetrical PS).

Table 4.7: Analysis of performance using the IGDx indicator on the MMF1-MMF9 benchmarks using 10,000 function evaluations. A comparison of different variation of Gr_{dec} and WSCD approaches has been conducted using NSGA-II. The best performing algorithms are highlighted in bold and gray shaded, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	NSGA-II-Gr	NSGA-II-Gr-CDDec	NSGA-II-Gr-CDobj	NSGAII-Gr-CDws	NSGA-II-CDDec	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	0.067124(0.008875)*	0.061202(0.003417)	0.062595(0.002853)*	**0.060603**(0.003125)	0.061958(0.002577)*	0.062318(0.002726)*	0.073865(0.005652)*	0.11003(0.02105)*
MMF1z	0.052395(0.003493)*	**0.044782**(0.00213)	0.047838(0.002698)*	0.045687(0.002583)	0.045896(0.001208)*	0.046543(0.002298)*	0.055018(0.004518)*	0.12034(0.025452)*
MMF2	0.018418(0.004607)	0.018879(0.003991)	0.018996(0.004189)	**0.017765**(0.003369)	0.019397(0.004749)	0.018523(0.003559)	0.031923(0.012669)*	0.10784(0.087009)*
MMF3	0.015085(0.002805)	**0.014636**(0.001038)	0.016876(0.00313)*	0.014908(0.002532)	0.015515(0.00159)*	0.015936(0.002998)*	0.024832(0.008584)*	0.066651(0.031745)*
MMF4	0.045234(0.003421)*	0.038038(0.002652)	0.038727(0.002337)	**0.037983**(0.002413)	0.039865(0.00547)*	0.041913(0.003713)*	0.04535(0.003056)*	0.10256(0.047135)*
MMF5	0.11035(0.005561)*	0.1034(0.003865)	0.10417(0.006102)	**0.10222**(0.005236)	0.11292(0.005694)*	0.11425(0.010062)*	0.12693(0.011658)*	0.20553(0.048088)*
MMF6	0.095281(0.008412)*	**0.089501**(0.00341)	0.091343(0.002659)	0.091507(0.003796)	0.098542(0.006039)*	0.09933(0.007392)*	0.10804(0.011067)*	0.1892(0.073865)*
MMF7	0.044455(0.003232)*	0.038889(0.002492)*	**0.036447**(0.002983)	0.03801(0.002741)*	0.03704(0.002444)	0.036946(0.002157)	0.043546(0.003618)*	0.067719(0.021536)*
MMF8	0.088645(0.005978)*	0.0791(0.006105)	0.088348(0.007404)*	**0.076404**(0.004284)	0.083033(0.010748)*	0.086917(0.007588)*	0.10699(0.011313)*	0.8061(0.63768)*
MMF9	0.012333(0.000966)*	**0.010691**(0.000806)	0.011113(0.001116)*	0.010808(0.000977)	0.011338(0.001384)*	0.013275(0.001564)*	0.013383(0.002343)*	0.24982(0.20045)*
Best/All	0/10	4/10	1/10	5/10	0/10	0/10	0/10	0/10

As an additional means of evaluating the performance of the proposed algorithms regarding the approximation of PF, IGD^+ values were analyzed. In nine of 10 test cases, the NSGA-II-Gr-CDobj algorithm performs significantly

Table 4.8: Analysis of performance using the PSP indicator on the MMF1-MMF9 benchmarks using 10,000 function evaluations. A comparison of different variation of Gr_{dec} and WSCD approaches has been conducted using NSGA-II. The best performing algorithms are highlighted in bold and gray shaded, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns [102].

Problems	NSGA-II-Gr	NSGA-II-Gr-CDDec	NSGA-II-Gr-CDobj	NSGAII-Gr-CDws	NSGA-II-CDDec	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	14.8073(1.9046)*	16.2504(0.87617)	15.8527(0.72453)*	**16.4489**(0.78003)	15.9796(0.68082)*	15.9255(0.73988)*	13.2792(0.98137)*	8.8227(1.6945)*
MMF1z	19.0266(1.2466)*	**22.2021**(1.1178)	20.8208(1.3242)*	21.8624(1.3449)	21.6625(0.5915)*	21.3178(1.1579)*	17.9275(1.5125)*	8.0487(1.837)*
MMF2	54.1538(14.3124)	52.9243(13.3026)	52.6005(10.6054)	**55.1813**(9.7281)	51.541(12.4749)	53.9273(9.5183)	29.1782(10.7133)*	7.9555(6.7582)*
MMF3	66.0964(11.5562)	**67.9741**(4.8889)	59.2371(11.6847)*	66.7617(10.7485)	64.4272(6.5669)*	62.7239(11.1474)*	38.3825(12.4391)*	14.5135(7.267)*
MMF4	21.8348(1.7856)*	26.1288(1.9195)	25.6534(1.579)	**26.1896**(1.6784)	24.941(3.2196)*	23.7223(2.1679)*	21.5821(1.454)*	9.5321(4.0232)*
MMF5	9.0458(0.46834)*	9.6169(0.33912)	9.5596(0.57918)	**9.7424**(0.51451)	8.8265(0.42334)*	8.7221(0.69202)*	7.7577(0.71897)*	4.7613(1.1202)*
MMF6	10.467(0.91076)*	**11.0811**(0.46192)	10.9065(0.33621)	10.8866(0.48719)	9.9693(0.62399)*	10.0447(0.69786)*	9.1072(1.0346)*	5.1003(2.3573)*
MMF7	22.4605(1.4481)*	25.351(1.9344)*	**27.1832**(2.1236)	26.1666(1.7906)*	26.7337(1.678)	26.7984(1.5704)	22.674(2.0256)*	14.1855(4.2449)*
MMF8	11.1843(0.69393)*	12.576(0.97508)	11.2055(0.9299)*	**12.9898**(0.6339)	12.0055(1.6526)*	11.4086(0.88867)*	9.2193(1.0427)*	0.99354(0.51834)*
MMF9	81.0639(6.131)*	**93.4914**(6.7324)	89.9869(9.4493)	92.5107(8.4157)	87.9327(10.8428)*	75.2867(9.1232)*	74.2549(12.4973)*	0.71175(19.3066)*
Best/All	4/10	1/10	5/10	0/10	0/10	0/10	0/10	0/10

Table 4.9: Analysis of performance using the IGD$^+$ indicator on the MMF1-MMF9 test problems, using 10,000 function evaluations. A comparison of different variation of Gr_{dec} and WSCD approaches has been conducted using NSGA-II. The best performing algorithms are highlighted in bold and gray shaded, and significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	NSGA-II-Gr	NSGA-II-Gr-CDDec	NSGA-II-Gr-CDobj	NSGAII-Gr-CDws	NSGA-II-CDDec	NSGA-II-WSCD	Mo-Ring-PSO-SCD	NSGA-II
MMF1	0.007599(0.00073)*	0.006187(0.000424)*	**0.005055**(0.000386)	0.00584(0.000639)*	0.005754(0.000336)*	0.005418(0.00055)*	0.006533(0.000504)*	0.005331(0.000233)*
MMF1z	0.007038(0.000836)*	0.005919(0.000341)*	**0.004838**(0.000185)	0.005703(0.000548)*	0.005782(0.000281)*	0.005298(0.000462)*	0.006554(0.00057)*	0.005145(0.000222)*
MMF2	**0.014274**(0.003097)	0.015225(0.00409)	0.015057(0.003806)	0.015032(0.002328)	0.014505(0.002001)	0.014852(0.002164)	0.019609(0.006034)*	0.020704(0.019092)*
MMF3	0.011869(0.002062)	0.011685(0.002171)	0.011666(0.002669)	0.012051(0.002231)	**0.011566**(0.001405)	0.012024(0.002053)	0.015091(0.003205)*	0.014244(0.006389)*
MMF4	0.008182(0.00144)*	0.006185(0.00058)*	**0.005068**(0.000225)	0.00619(0.000593)*	0.005954(0.000284)*	0.00545(0.00056)*	0.006793(0.000809)*	0.005221(0.000259)*
MMF5	0.007493(0.000737)*	0.006361(0.000506)*	**0.005031**(0.000332)	0.006111(0.000515)*	0.005691(0.000347)*	0.005275(0.000651)	0.006285(0.000481)*	0.00537(0.000226)*
MMF6	0.007036(0.001198)*	0.006074(0.000735)*	**0.004996**(0.000283)	0.005967(0.00066)*	0.005702(0.000356)*	0.005445(0.000493)*	0.006374(0.000755)*	0.005198(0.000334)*
MMF7	0.008725(0.00095)*	0.007058(0.00083)*	0.005098(0.000304)	0.006122(0.000588)*	0.006702(0.000767)*	0.005271(0.000542)*	0.007676(0.001388)*	**0.005016**(0.000264)
MMF8	0.007482(0.000784)*	0.006989(0.000668)*	**0.005144**(0.000178)	0.00552(0.000312)*	0.006857(0.00046)*	0.005223(0.000338)*	0.007582(0.000663)*	0.005289(0.000292)*
MMF9	0.033322(0.005984)*	0.027648(0.002816)*	0.019675(0.00109)*	0.01997(0.001073)*	0.02757(0.003082)*	**0.019284**(0.000882)	0.028642(0.003486)*	0.020262(0.001267)*
Best/All	0/10	0/10	6/10	0/10	1/10	1/10	0/10	1/10

better than the original NSGA-II algorithm. The reason for this observation is that the algorithm is mostly focused on providing evenly distributed solutions in objective space rather than in decision space.

To further evaluate the performance of the proposed algorithms on the approximation of the PF, IGD$^+$ values were studied. These results, as shown in Table 4.9, indicate that the NSGA-II-Gr-CDobj algorithm significantly outperforms the original NSGA-II algorithm in nine out of 10 test cases. The obvious reason for this observation is that the focus of this algorithm is on providing better-distributed solutions in objective space while ignoring decision space. On the other hand, the NBM operator enhances the exploration of the search space, leading to a better identification of Pareto-optimal solutions during the search process.

It is evident that the obtained IGDx and PSP values for both NSGA-II-Gr-CDdec and NSGA-II-Gr-CDws performed better than the previous algorithms without considering the Gr_{dec} approach without degrading the IGD^+ values much. Based on this analysis, the proposed NSGA-II-Gr-CDdec and NSGA-II-Gr-CDws agorithms are successful in preserving the discrete solution in a search space as a result of the proposed Gr_{dec} method.

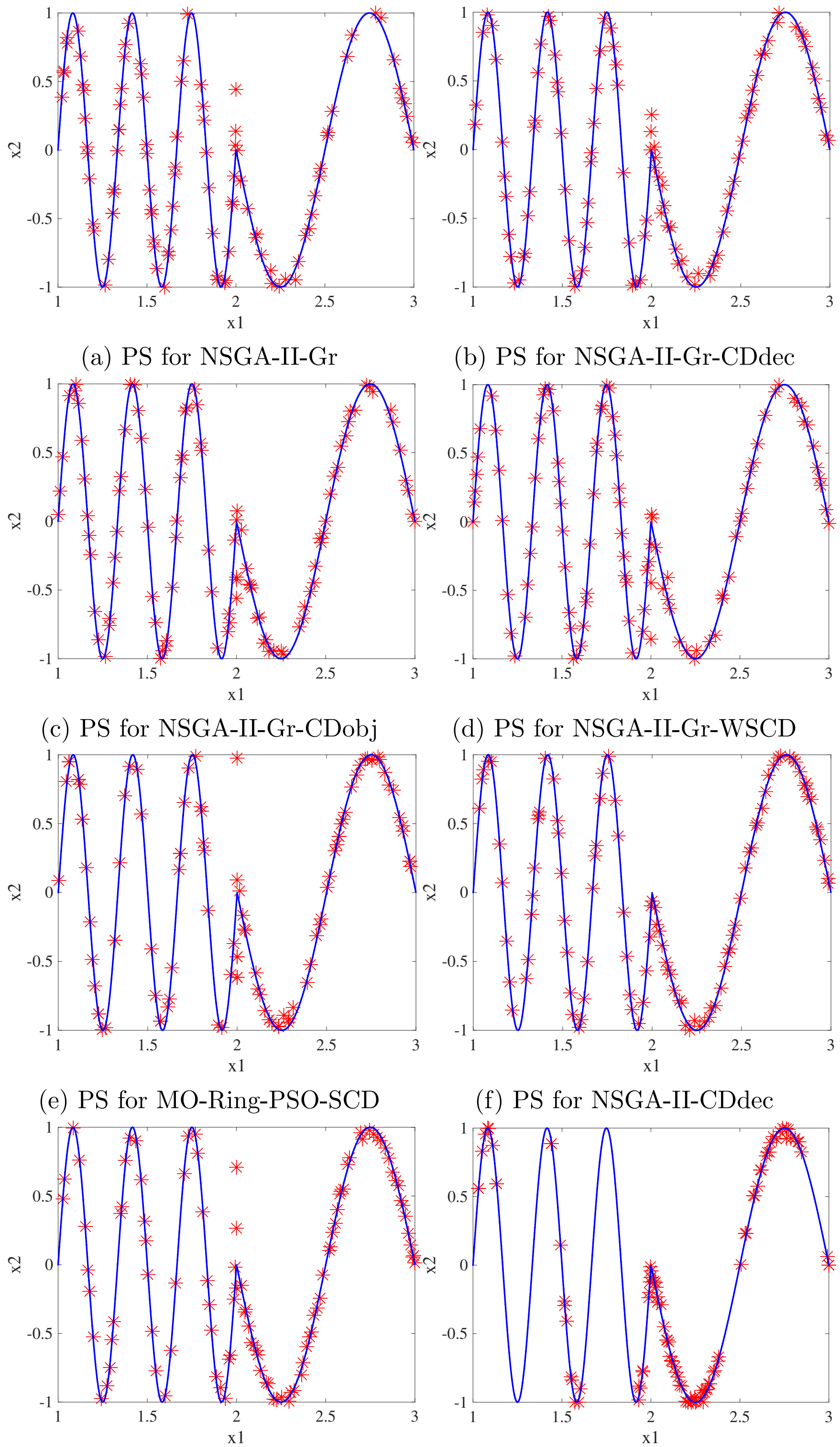
The overall reason for the better performance of both of the proposed NSGA-II-Gr-CDdec and NSGA-II-Gr-CDws over the rest of the compared MMOPs is that in early generations, solutions may be randomly distributed over the search space. In some cases, grid-based crowding distances are zero; however, after normalizing the value, they become non-zero. By dividing the obtained values with the crowding distance values, the solution is chosen primarily based on the crowding distance values, which compensate for the absence of grid-based crowding distance values. In the later generations, as the solutions converge toward the PS, they are closer to each other, and therefore their Gr_{dec} values differ. Thus, the combination of this distance with the crowding distance outperforms crowding distance alone by considering a wider range of solutions than the two-nearest-neighborhoods considered by crowding distance. As a result of this combination, more diverse solutions are selected along the PS as the selection pressure on the decision space increases in comparison to the NSGA-II-CDdec.

To demonstrate the similarities between the obtained solutions in both the decision and objective spaces, some results are presented in Figures 4.11, and 4.12 for the median run of the IGDx performance indicator. The solutions obtained by all the competitors are presented in both the decision and objective spaces. Solid blue lines illustrate the true PS and PF, while red markers illustrate the obtained solutions. From Figures 4.11, and 4.12, for example, it can be observed that the obtained solutions for both NSGA-II-Gr-CDdec and

NSGA-II-Gr-CDws cover a larger area and are more evenly distributed over the PS, corresponding to the two PSs of the problem, located respectively in the ranges of $x_1 \in [1,2]$ (left side) and $x_1 \in [2,3]$ (right side). However, in NSGA-II-WSCD (Figure 4.9[d]), the optimally obtained solutions are unevenly distributed over the PS, meaning they are denser on the right side than on the left. As a result, the optimal solutions for NSGA-II cover more points on the right side of the PS, and these points are positioned closely together. Due to the straightforward shape of the right side of this specific problem, both NSGA-II and NSGA-II-WSCD found more Pareto-optimal solutions in this particular area. Hence, the proposed grid-based methods are capable of covering more points in both PSs, and the solutions are distributed more uniformly across them.

4.3 Proposed Approaches for Preserving both the Local and Global Pareto-sets of Solutions

In the past few years, a considerable amount of research has been undertaken to develop approaches to deal with MMOPs of the first type that have multiple equivalent PSs; however, only limited research has been done on approaches to deal with MMOPs of the second type that have both global and local PSs. Decision-makers are often interested in having information about local PSs that are of inferior quality but are acceptable in case the global PSs are no longer feasible in practice due to changes in the environment or because global optimal solutions are too costly to implement. However, when it comes to capturing and preserving local optimal solutions, the main challenge arises because local Pareto-optimal solutions with acceptable quality are dominated and replaced by non-dominated solutions using non-dominated sorting methods implemented across the entire population of solutions. Moreover, since the primary objective of the MOEAs is to provide a better distribution of optimal solutions over the PF, multi-modal solutions with similar objective function values but located in different regions of the search space may be eliminated from the search process in these algorithms. Thus, to address this type of MMOPs, we propose a method that preserves both local and global optimal solutions, thus providing better approximations of local and global PSs. With the help of niching methods, we are able to retain solutions that are both distant in the search space and close to the objective space. In the next section,



(a) PS for NSGA-II-Gr

(b) PS for NSGA-II-Gr-CDdec

(c) PS for NSGA-II-Gr-CDobj

(d) PS for NSGA-II-Gr-WSCD

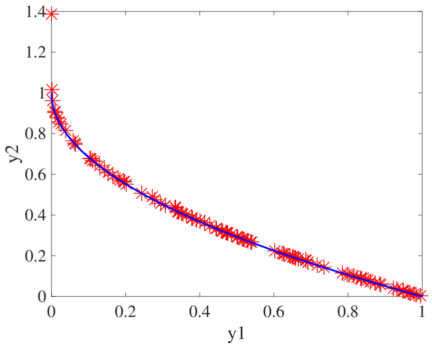
(e) PS for MO-Ring-PSO-SCD

(f) PS for NSGA-II-CDdec

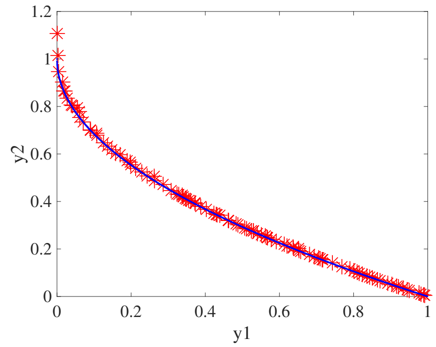
(g) PS for NSGA-II-WSCD

(h) PS for NSGA-II

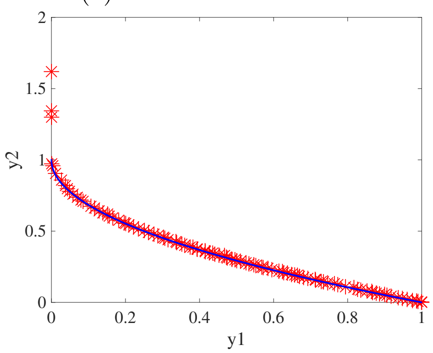
Figure 4.11: Obtained solutions in decision space for MMF1z test problem [65]



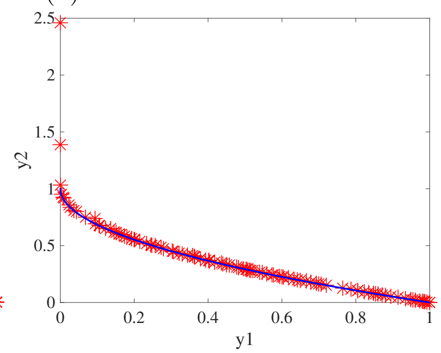
(a) PF for NSGA-II-Gr



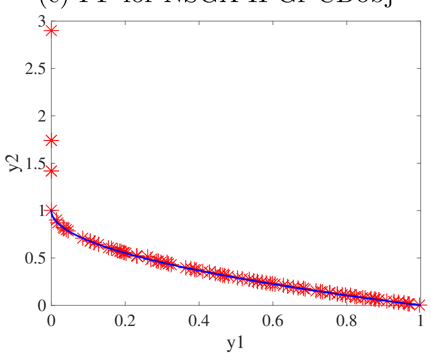
(b) PF for NSGA-II-Gr-CDdec



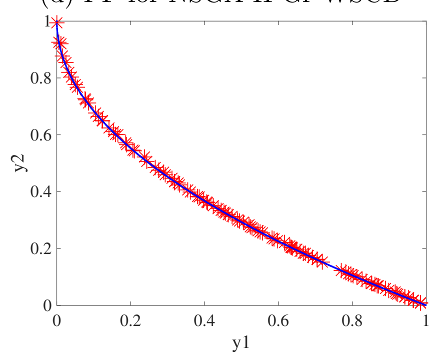
(c) PF for NSGA-II-Gr-CDobj



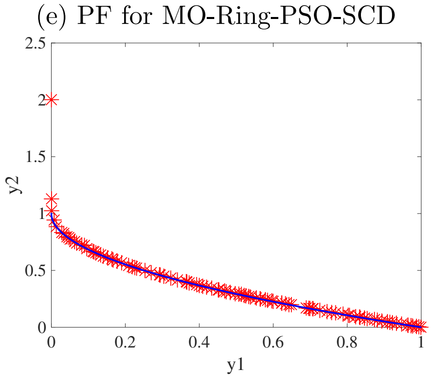
(d) PF for NSGA-II-Gr-WSCD



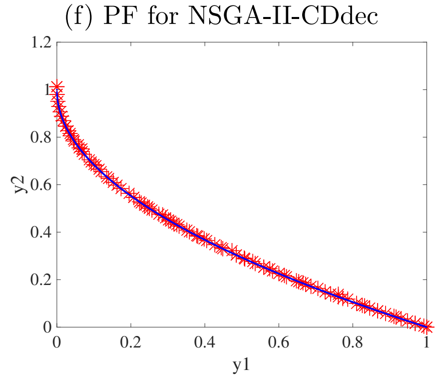
(e) PF for MO-Ring-PSO-SCD



(f) PF for NSGA-II-CDdec



(g) PF for NSGA-II-WSCD



(h) PF for NSGA-II

Figure 4.12: Obtained solutions in objective space for MMF1z test problem [65]

we will provide a detailed explanation of our suggested approach for addressing these specific types of problems.

4.3.1 DBSCAN Clustering Approach

The density-based spatial clustering of applications with noise (DBSCAN) algorithm, first proposed in 1996 by [132], is a clustering algorithm that relies on neighborhood radius (Eps) and $MinPts$, which is the minimum number of points that allow them to be considered as one cluster. A variety of situations can be tackled with this technique, including multi-objective optimization, which utilizes the strengths of clustering algorithms to solve complex optimization problems. Multi-objective optimization problems can be solved effectively by DBSCAN because it is a clustering algorithm based on density. In this algorithm, regions of high density in the objective space are identified, and solutions within those regions are clustered together. It is thus capable of identifying clusters with irregular shapes or varying densities, making it suitable for complex optimization problems in which there are multiple local optima. Nevertheless, to solve MMOPs, we used this clustering technique to find the dense areas in the search space so as to generate a greater diversity of solutions.

A description of the DBSCAN clustering technique is given in Algorithm 8. The $MinPts$ parameter indicates the minimum number of points to take into consideration when building a cluster. If the Euclidean distance between two points is equal to or less than the value of Eps, they are regarded as neighbors. For a solution b that belongs to the solution set U, the neighborhood of that point is indicated by $N_{Eps}(q)$:

$$N_{Eps}(q) = [a \in U | dist(p, q) \leq Eps) \tag{4.6}$$

There are two conditions to satisfy for a point q to directly reach a point p: 1) the data point q must satisfy the core point condition,

$$|N_{Eps}| \geq MinPts \tag{4.7}$$

Input: $\tilde{P}$: union of the population and offspring; $MinEps$; Eps: neighborhood raduis
Output: C_1,C_2,...,C_k

$Core \leftarrow \emptyset$;
for $x \in \tilde{P}$ **do**
    $Neigbor(x) = \left\{ y \in \tilde{P} \mid \|y - x\| \leq Eps \right\}$; `// Find direct density-reachable solutions for solutions` x
    if $|Neigbor(x)| \geq MinEps$ **then**
        $Core \leftarrow Core \cup x$; `// Consider` x `as a core solution`
    end
    if $x \in core$ **then**
        $i = 1$;
        while $\tilde{P} \neq \emptyset$ **do**
            select an arbitrary solution x from $\tilde{P}$;
            $U = C_i \leftarrow \{x\}$;
            while $U \neq \emptyset$ **do**
                Select an arbitrary solution from U;
                $B \leftarrow Neigbor(x) \setminus C_i$;
                $U \leftarrow B \cup B$ and $C_i = C_i \cup B$;
                $U \leftarrow U \setminus \{x\}$
            end
            $\tilde{P} \leftarrow C_i$;
            $i \leftarrow i + 1$
        end
    else
        Consider $\{x\}$ as noise
    end
end
return C_1,C_2,...,C_k

Algorithm 8: Clustering Algorithm: DBSCAN-Algorithm($\tilde{P}$, $MinEps$, Eps)

2) the data point p must be in the $N_{Eps}(q))$ neighborhood of the point q:

$$p \in N_{Eps(q)} \tag{4.8}$$

A point a is densely reachable from a point b if the following conditions are met: 1) There exists a chain of points, denoted as $p_1, p_2, \cdots, p_n$, that connects the two points. Specifically, p_1 is point a, and p_n is point b. 2) All the points in the chain, from p_1 to p_{n-1}, must also be core points. In other words, each intermediate point in the chain must have enough neighboring points within a specified radius to qualify as a core point.

Any point in the solution set U that cannot be reached by any core point is considered noise and is excluded from forming any cluster. To control the amount of noise in the dataset, a parameter called $MinPts$ is utilized.

4.3.2 EMMOA – $\mathbf{XY_{local}}$ Algorithm

This section provides a detailed description of our algorithm with the goal of maintaining both local and global PSs simultaneously. The suggested algorithm is called EMOA-XY-Euc. As part of our suggested approach, we employed $DBSCAN$ to cluster the solutions in the search space.

The clustering algorithm mentioned above is applied in the search space using a similar approach as described in [115]. In line with the methodology presented in the algorithm proposed by [115], we incorporate a non-dominated sorting algorithm within each cluster. This enables us to identify non-dominated solutions within the neighborhood of each cluster and maintain them during the search for additional global optima in the search space.

In the next step, our proposed density measurement is applied to the non-dominated solutions from each cluster as the secondary selection criteria. The density measurement is adapted to both the search and objective spaces. By preserving the diversity of solutions in the search space as they are close to their objective spaces, the algorithm can locate and preserve more diverse solutions in the search space.

4.3.3 General Framework

The primary structure of the proposed algorithm, named EMMOA – $\mathrm{XY_{local}}$, is depicted in Algorithm 9. Through this algorithm, non-dominated solutions obtained from individual local regions are collected and merged with the global non-dominated solutions. A diversity estimator metric is subsequently used to rank them. Finally, the N first solutions, ranked in best-to-worst order based on their diversity in both decision and objective spaces, constitute the next generation of population.

On the basis of Algorithm 9, the inputs are the MMO problem (MMOP[n,m]); the noise elimination parameter; the population size (N); the minimal number of solutions for each local cluster (num); and the size of the neighborhood of the DBSCAN cluster (Eps), which is determined in the following way:

$$Eps = \alpha \cdot (X_i^u - X_i^l) \tag{4.9}$$

where α is considered as a tuning factor for neighborhood size, the X_i^u and X_i^l values are the upper and lower bounds for the decision variable i, and the output represents the final population. The major components of our proposed algorithm are outlined as follows.

Following the initialization of the population (Line 2), solutions are then analyzed in Line 3. In Line 5, the parental population is selected using the mating selection operator. In Lines 6–7, the SBX and polynomial mutations are applied to the parental population, yielding an offspring named Q. After evaluating the offspring population in Line 8, the current population and its offspring are combined in Line 9 to form a union population. In Line 10, a modified environment selection is implemented to create the next population. In the next subsection, the procedure of the proposed environmental selection is explained in detail in Algorithm 13. The process will continue until the termination criterion is satisfied, and then the final population in Line 12 will be returned.

4.3.4 Proposed Environmental Selection

In Algorithm 10, we demonstrate the modified environmental selection, which is detailed below. Algorithm input parameters include the combination of the population and the offspring ($\tilde{P}$), the neighborhood radius (Eps), and (num) is the minimum number of solutions in each cluster. The population for the next generation is the output of this algorithm, $P(t+1)$. Initially, the solutions are sorted using a non-dominated sorting method and divided into fronts (Front $= [Front = (Front_1, Front_2, ...)]$; Line 2).

To implement the $DBSCAN$ clustering method to cluster solutions in the search space, described in detail in [132], the solutions must first be normalized in the search space using Equation 4.10 (Line 5). This is due to the $DBSCAN$ clustering algorithm's sensitivity to the scale of the decision variables [133].

Input: MMOP(n,m): An optimization problem with n dimensions of search space (i.e. [X^l, X^u] lower and upper boundaries for each dimension), and m dimensions of objective space; N: Population Size; Eps: neighborhood raduis ; $N_{eps(b)}$; num: Minimun number of solutions for each cluster

Output: P: Final population

1 $t = 0$;
2 $P(t) \leftarrow$ Initialize($P(t)$);
3 Evaluate($P(t)$);
4 **while** *Termination critera not met* **do**
5 $P_{mate} \leftarrow$ Tournoment-Selection($P(t)$);
6 $\tilde{P(t)} \leftarrow$ SBX-Corossover(P_{mate});
7 $Q \leftarrow$ Ploynomial-Mutatation($\tilde{P}$);
8 Evaluation($Q(t)$);
9 $\tilde{P} \leftarrow P \cup Q$;
10 $t = t + 1$;
11 $P(t) \leftarrow$ Modified-Environmental-Selection($\tilde{P}$, Eps, N, num) ; `// Algorithm 13`
12 **end**
13 **return** P

Algorithm 9: General Framework EMMOA – $\mathrm{XY}_{\mathrm{local}}$ Algorithm

Using the $DBSCAN$ clustering algorithm, the number of clusters k is not predetermined; it is determined automatically by the algorithm based on the density and distribution of the data points.

$$x^{'(j)} = \frac{x^{(j)} - x^{(j)}_{min}}{x^{(i)}_{max} - x^{(i)}_{min}} \quad j = 1, \cdots, n \tag{4.10}$$

The $x^{(j)}_{min}$, $x^{(j)}_{max}$ values represent the min and max value for the decision variable j, and $x^{(j)}$ value represents the j_{th} decision variable for the vector x. In multimodal problems, it is important to preserve those equivalent distinct solutions located in sparse areas of the search space and not consider them as noise. This retains the diversity of solutions in the search space. Parameter A for the elimination of noise must therefore be considered 1, which allows even single solutions to be considered clusters in the sparse areas of the search space and to be kept within the search process.

After the solutions have been clustered (C_1,C_2,...,C_k), the fast non-dominant sorting method is applied to each cluster (Lines 7–11). Next, when the number

Input: $\tilde{P}$; N; Eps; $MinEps$; num
Output: $P(t+1)$: Next population
1 $BestPop \leftarrow \emptyset$;
2 $Front \leftarrow$ Fast-Non-Dominated-Sort($\tilde{P}$) ; // $Front = (Front_1, Front_2, ...)$
3 $i = 1$;
4 $j = 1$;
5 Normalize $\tilde{P}$ in the search space;
6 // Cluster the solutions within the search space using DBSCAN
7 C_1,C_2,...,$C_k \leftarrow$ DBSCAN-Algorithm($\tilde{P}$, $N_{eps(b)}$, Eps) **for** $i = 1, ..., k$ **do**
8   **if** $|C_i| > num$ **then**
9         $Front_{C_i} \leftarrow$ Fast-Non-Dominated-Sort(C_i) ; // $Front_{C_i} = (Front1_{C_i}, Front2_{C_i}, ...)$
10         $BestPop \leftarrow Front1_{C_i} \cup BestPop$
11   **else**
12         $i = i + 1$
13   **end**
14 **end**
15 $BestPop \leftarrow BestPop \cap Front_1$;
16 **while** $|BestPop| + |Front_j| \leq N$ **do**
17     $BestPop \leftarrow BestPop \cup |Front_j|$;
18     $j = i + 1$
19 **end**
20 **while** $|BestPop| \geq N$ **do**
21     $P(t+1) \leftarrow$ Density-Estimator($BestPop$, N) ; // Algorithm 11
22 **end**
23 **return** $P(t+1)$

Algorithm 10: Environmental Selection: Modified-Environmental-Selection($\tilde{P}$, Eps, N, num)

of solutions within a cluster is greater than the minimum cluster size (num), the non-dominated solutions from these clusters are added to the *Bestpop* (Line 10), with the intention of preserving the local PSs. Following excluding global non-dominated solutions from the solution set in *Bestpop* (Line 15), the solutions from $front_1$ and $front_2$ are added to $front_i$ in turn until the total number of solutions in *Bestpop* exceeds the population size (N; Lines 16–19). Following this, our modified diversity measurement method is implemented for the population in the *Bestpop* to preserve more diverse solutions in the search space (Algorithm 11), which we describe in greater detail below. As a result of this secondary selection criteria, the solutions for the next population are passed on to the next generation (Line 22).

Algorithm 11 illustrates our modified version of density preservation as a

```
   Input: BestPop ; N
   Output: P(t+1)
1  for x ∈ BestPop do
2      Identify the solution y ∈ BestPop with minimum euclidean distance to the solution x in
        the objective space ;
3      dist(x_obj) = ‖x − y‖;
4      Find the coresponding solution y in the decision space represented by y' that maps to
        the solution y in the objective space (y = f(y'));
5      dist(x_dec) = ‖x' − y'‖;
6      dist(x) = dist(x_obj) · dist(x_dec)
7  end
8  while |BestPop| ≥ N do
9      for x ∈ BestPop do
10         Find the neighborhood size for the solutions by Equation 4.12 ; Calucluate the
            Harmonic Average Distance value of the solution x (HAD(x))in the search space by
            ; density(x) = dist(x) · HAD(x)
11     end
12     Findout the solution x ∈ BestPop with the minimun density value ;
        BestPop = BestPop \ {x};
13 end
14 P(t+1) = BestPop
15 return P(t+1)
```

Algorithm 11: Modified secondary selection criteria: Density-Estimator($BestPop$, N)

secondary selection criteria. In Line 2 of the first step for each solution x, we find the solution y that is closest in the objective space using the Euclidean distance metric. In Line 3, we calculate the Euclidean distance between the solutions x and y in the objective space (i.e. $dist(x_{obj}) = \|x - y\|$). Then we locate the solutions y' and x' in the search space that correspond to the solutions y and x in the objective space (Line 4).

$$dist(x) = dist(x_{obj}) \cdot dist(x_{dec}) \tag{4.11}$$

While the size of the remaining solutions in *Bestpop* is larger than the population size, we perform the following steps (Lines 8–16). Using the following equation, the neighborhood size of the solutions is determined in Line 10:

$$Neighborhood - size = \sqrt{|bestpop|} \tag{4.12}$$

Then, we calculate the harmonic average distance (HAD) between each solution and its K nearest neighbor (i.e. $K = \lfloor Neighborhood - size \rfloor$):

$$HAD(x_i) = \frac{K}{\sum_{j=1}^{K} \frac{1}{d_{ij}}} \tag{4.13}$$

Where the Euclidean distance between the solution i and its j_{th} nearest neighbor is represented by d_{ij}. According to the example visualized in [34], the advantage of using harmonic average distance over crowding distance is that it is able to more accurately indicate whether solutions reside in the crowded area. Consequently, in Line 12, to calculate the density of the solution x in its neighborhood area in the search space, its obtained HAD distance value to the k nearest neighbors ($HAD(x)$) is multiplied by the distance $dist(x)$ obtained for the solution x ($density(x) = dist(x) \cdot HAD(x)$). Considering that these distances are measured at various scales, we have multiplied them. This is because we aim to remove the need for normalization of these distance values and to consider the effect of each distance value on the density of the solutions equally.

Figure 4.13 illustrates an example of a proposed density measurement method. For the solution A', the nearest neighbor solution is represented by B' in the objective space. Next, we calculate the distance between these two solutions ($dist_{obj}(A', B')$). Within the search space, we identify solutions denoted as A and B, while mapping them to their respective counterparts, A' and B', in the objective space. In the search space, we calculate the distance between these two solutions ($dist_{dec}(A, B)$). Then, we compute the harmonic average distance (HAD) between A and its K-nearest neighbor ($K = \sqrt{N} = \sqrt{9} = 3$) in the search region. As a result, the density of solution A ($density(A)$) is calculated by multiplying these distances with each other.

4.3.5 Specification of Benchmark Problems

Apart from the test cases outlined in Section 4.4.2, we also performed the following tests to evaluate the coexistence of the global and local PSs on the MMOPs: MMF11, MMF12, MMF13, and MMF15 benchmarks have two decision variables and two objective functions, which are scalable with the number of local PSs. However, in these test cases, we consider the coexistence

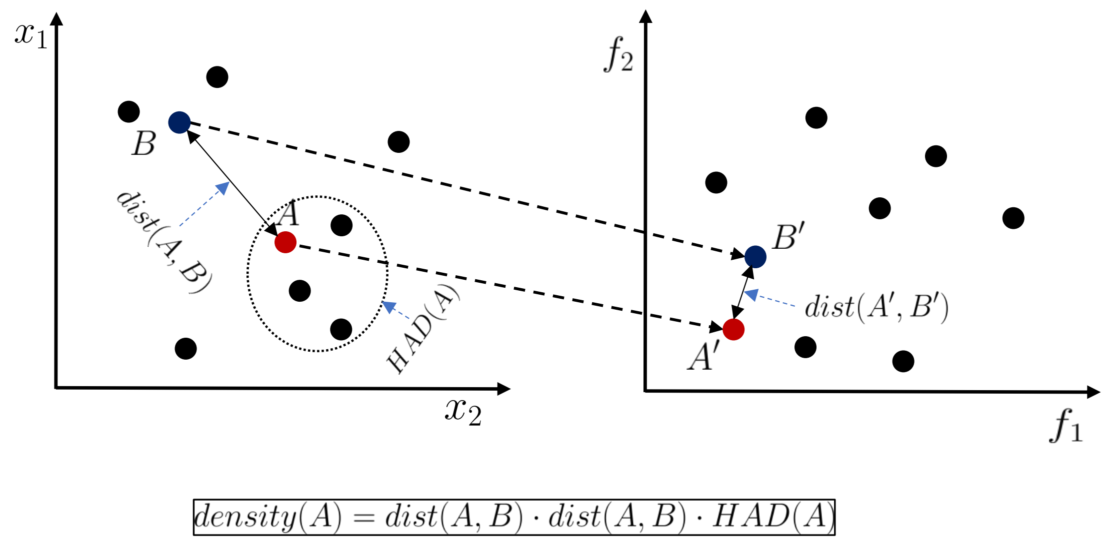


Figure 4.13: An example of the calculation for the proposed density estimation for solution A.

of a global PS and a local PS, and, accordingly, one global PF and one local PF.

4.3.5.1 Configuration of the Proposed Method

The configuration of the proposed method is as follows: for the DBSCAN clustering algorithm, the parameter α, which controls the neighborhood radius (i.e. EPs) and the num value are set to 0.1 and 5 accordingly, as it is stated in [115]. A HAD distance measurement using the NxEMMO algorithm has a neighborhood size of $\sqrt{\sum_{i=1}^{k-1} Front_i}$, where k in this case refers to the front number that required to be used as a front number to select $N-(\sum_{i=1}^{k-1} |Front_i|)$ remaining solutions (i.e. N is the population size).

4.3.5.2 Discussion of results

In order to assess the effectiveness of the proposed algorithm in comparison to other competing algorithms, we present comprehensive results for IGDx, PSP, and IGD$^+$ in Tables 4.10, 4.11, and 4.12. These tables include the median values and interquartile ranges (IQR) calculated from 31 independent runs of the algorithm for each performance indicator.

To examine the performance of the proposed $\mathrm{EMMOA}-\mathrm{XY}_{\mathrm{local}}$ algorithm, we compare it with two other state-of-the-art algorithms: the NxEMMO (detailed explanation provided in Section 5.2.1) and the MMOE/AC [115] algorithms (the state-of-the-art algorithms for solving type II MMOPs), as well as the NSGA-II as the baseline algorithm. Per the results in Table 4.10, in general, the proposed algorithm performed the best among the other algorithms on seven out of 15 test problems. Based on the performance of the algorithms over the test cases that contained both local and global PSs, it was observed that in two out of four test cases, the proposed algorithm outperformed the MMOE/AC algorithm, and in one test case where the MMOE/AC algorithm obtained superior results, no statistical significance was reported between the two algorithms. As seen from the comparison of this algorithm with the NxEMMO algorithm, the proposed algorithm in all the test cases containing local PSs was capable of preserving more spots on the local PS while preserving the global PSs. Three out of 11 tests containing only global PSs yielded the best results when using the NxEMMO algorithm. As the results of the tests show, the proposed algorithm is not only superior to NxEMMO when it comes to preserving the local PSs but also performs better when dealing with just global PSs in 7 out of 11 test cases.

To maintain the local optimal solutions, it is essential to integrate a clustering method into the search space. By retaining the non-dominated solutions within each cluster, the local optimal solutions are preserved throughout the search process. By employing the proposed density measurement technique in the search space, the optimizer ensures the preservation of local Pareto-optimal solutions and subsequently proceeds to locate a global Pareto-optimal solution during the evolutionary search.

As shown in Table 4.12, when using the IGD^+ performance indicators to compare algorithms regarding their performance on the approximation of the PF, the proposed algorithm was outperformed by others on all test cases. Conversely, MMOE/AC is the best-performing algorithm in five out of 11 test cases that include the global PSs. This is due to the hierarchy clustering technique that is employed in the objective space to further preserve the diversity of the solution and therefore approximate the PF more accurately. While the main focus of the algorithm in this study is on the distribution of solutions in the search space, in multi-modal problems there is a trade-off between the diversity of the solutions in the search and objective spaces. As

Table 4.10: Analysis of performance using the IGDx indicator on the MMF1-MMF15 test problems using 10,000 function evaluations A comparison of the $NSGAIIXY_{local}$ algorithm and the other state-of-the-art algorithms The best-performing algorithms are highlighted in bold and gray shading, and their significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	$EMMOA - XY_{local}$	NxEMMO	MMOEAC	NSGAII
MMF1	**0.063703(0.004499)**	0.06826*(0.011957)	0.065891*(0.00304)	0.11478*(0.023295)
MMF1z	0.05036(0.006777)	0.054599*(0.013128)	**0.049536(0.003536)**	0.11967*(0.023685)
MMF2	**0.019686(0.006711)**	0.086632*(0.060714)	0.020344(0.003601)	0.085467*(0.064877)
MMF3	**0.019131(0.003288)**	0.072191*(0.042195)	0.020345(0.006536)	0.068035*(0.046307)
MMF4	0.041289*(0.003202)	**0.034715(0.003061)**	0.042502*(0.00379)	0.10556*(0.035845)
MMF5	0.57059(0.012583)	**0.56786(0.013278)**	0.591*(0.00927)	0.61874*(0.015645)
MMF6	**0.091691(0.004483)**	0.09557*(0.011502)	0.097371*(0.003753)	0.19387*(0.057345)
MMF7	* 0.036333(0.003384)	0.036965(0.003481)	0.040795*(0.005045)	0.069254*(0.017539)
MMF8	0.12503*(0.039628)	0.26061*(0.11824)	**0.09801(0.020285)**	0.7981*(0.33379)
MMF9	0.009969*(0.000473)	**0.007537(0.000432)**	0.011029*(0.00077)	0.093028*(0.2199)
MMF10	**0.01522(0.12058)**	0.20128(0.022631)*	0.016695(0.002183)	0.20203(0.016937)*
MMF11	**0.011373(0.000838)**	0.24902(0.000563)*	0.012554(0.000693)*	0.25045(0.001166)*
MMF12	0.005189(0.000515)	0.24476(0.000744)*	**0.005097(0.000543)**	0.24626(0.001106)*
MMF13	0.15705(0.030822)*	0.25794(0.00219)*	**0.14938(0.015624)**	0.30889(0.03992)*
MMF15	**0.12458(0.04274)**	0.2689(0.003997)*	0.13602(0.041955)*	0.2763(0.007056)*
Best/All	8/15	3/15	4/15	0/15
Wins/Losses/Ties	–	2/13/4	2/13/5	3/10/2

such, improving the distribution of the solutions in the search space might adversely affect how well the algorithm performs in the objective space.

Table 4.11: Analysis of performance using the PSP indicator on the MMF1-MMF15 test problems using 10,000 function evaluations A comparison of the $NSGAIIXY_{local}$ algorithm and the other state-of-the-art algorithms The best-performing algorithms are highlighted in bold and gray shading, and their significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	$EMMOA - XY_{local}$	NxEMMO	MMOEAC	NSGAII
MMF1	15.6378(1.0943)	14.632*(2.8334)	15.0204*(0.67793)	8.4532*(1.7456)
MMF1z	19.6325(2.8707)	18.1833*(4.5993)	20.1523(1.4141)	8.1724*(1.5981)
MMF2	49.5431(17.7406)	10.7635*(5.6645)	49.1464(8.9314)	9.9459*(7.8377)
MMF3	50.5629(10.9215)	11.3401*(10.1311)	48.4516(13.0693)	12.4462*(9.9079)
MMF4	24.1959*(1.9821)	28.6876(2.5845)	23.4269*(2.0486)	9.1544*(3.1222)
MMF5	1.2362(0.030535)	1.2386(0.030162)	1.1947*(0.017932)	1.1292*(0.06761)
MMF6	10.8594(0.63715)	10.3903*(1.2134)	10.2032*(0.3937)	4.9954*(1.6075)
MMF7	26.5736*(3.0536)	26.5611(3.6799)	24.2768*(2.9238)	13.775*(4.1412)
MMF8	7.845*(2.6586)	3.5652*(1.5098)	10.0144(2.1187)	1.0041*(0.39043)
MMF9	100.3115*(4.9371)	132.6703(7.5858)	90.6094*(6.3278)	10.7495*(34.3256)
MMF10	65.7022(51.6623)	0.15029(2.4653)*	59.8909(8.1726)	0.12076(2.048)*
MMF11	87.9305(6.5103)	0.71708(0.1368)*	79.6373(4.3744)*	0.5974(0.14036)*
MMF12	192.6812(19.0085)	0.55476(0.11163)*	196.1908(20.7349)	0.4769(0.15071)*
MMF13	5.5002(2.2617)*	1.8423(0.039078)*	6.2589(0.67636)	1.1662(0.58644)*
MMF15	8.0269(2.5592)	1.7018(0.2187)*	7.3265(2.4356)*	1.6514(0.31767)*
Best/All	8/15	3/15	5/15	0/15
Wins/Losses/Ties	–	2/13/4	2/13/5	3/10/2

4.4 Summary

To address multi-modal, multi-objective optimization problems, this chapter proposes two types of approaches. By combining these two types of approaches, existing algorithms are enhanced in their ability to handle MMOP. Using the

Table 4.12: Analysis of performance using the IGD^+ indicator on the MMF1-MMF15 test problems using 10,000 function evaluations A comparison of the $NSGAIIXY_{local}$ algorithm and the other state-of-the-art algorithms The best-performing algorithms are highlighted in bold and gray shading, and their significance relative to the best algorithms is indicated by an asterisk (*) in the respective columns.

Problems	$EMMOA - XY_{local}$	NxEMMO	MMOEAC	NSGAII
MMF1	0.004124*(0.000174)	0.004407*(0.000192)	0.003321(0.000236)	0.0039*(0.000136)
MMF1z	0.004607*(0.000506)	0.004409*(0.000496)	0.0034(0.000357)	0.003841*(0.000304)
MMF2	0.011688(0.002865)	0.013729*(0.00909)	0.011506(0.002206)	0.013448(0.015575)
MMF3	0.010876(0.001863)	0.011404(0.005682)	0.010655(0.001427)	0.009451(0.007619)
MMF4	0.004128*(0.000248)	0.004546*(0.000452)	0.003528(0.000206)	0.00347(0.000179)
MMF5	0.008723(0.001089)	0.009195*(0.000985)	0.008471(0.001436)	0.008859*(0.00101)
MMF6	0.004388*(0.000357)	0.004868*(0.000527)	0.003382(0.000306)	0.003975*(0.00029)
MMF7	0.005022*(0.000422)	0.004802*(0.000363)	0.003427(0.000242)	0.003858*(0.000161)
MMF8	0.071565*(0.00031)	0.071495*(0.000205)	0.071143*(0.000201)	0.070967(0.000126)
MMF9	0.008512*(0.000645)	0.005995(0.000403)	0.007847*(0.000863)	0.006528*(0.000437)
MMF10	0.005839(0.11131)*	0.00183(0.15196)	0.00457(0.000437)	0.002665(0.15081)
MMF11	0.006243(0.000565)*	0.00274(0.000218)	0.006959(0.000723)*	0.003098(0.000272)*
MMF12	0.001861(0.000246)*	0.00074(4.3e-05)	0.00181(0.000281)*	0.000777(7.1e-05)*
MMF13	0.008824(0.004057)*	0.003475(0.000897)*	0.007785(0.001474)*	0.003002(0.000612)
MMF15	1.0352(0.002908)*	1.0294(0.001342)	1.0379(0.00395)*	1.0293(0.000666)
Best/All	0/15	4/15	6/15	5/15
Wins/Losses/Ties	–	4/13/4	2/13/5	3/10/2

first approach, the algorithm is able to explore the search space more deeply, identifying multiple distinct solutions within the search space simultaneously by changing the tournament selection, reproduction, and mutation operators. Additionally, the proposed environmental selection procedures can be divided into two categories so that they can be applied to multi-modal, multi-objective optimization problems that involve multiple global Pareto sets (MMOP of type I) and problems that require both local and global Pareto sets to be maintained. After the introduction of each proposed method, the performance of each method was compared with several of the most recent and prominent related algorithms and previous methods proposed in the chapter. Several different MMOP with varying levels of complexity and dimensions in the decision space were used to assess the algorithms' functionality and suitability for handling MMOP with different properties. In both type I and type II MMOP, the results obtained from the proposed approaches demonstrate that the proposed algorithms are more efficient and effective in exploring and preserving the diverse solutions in the search space than the state-of-the-art algorithms.

Chapter 5

Inter- and Intra-Front Selection Operations

In this chapter, we propose a novel classification of environmental selection operators in a Pareto-dominance-based MMOEAs. The development of this classification involved considering whether to use solutions from other fronts to calculate crowding values, which we classified as intra- and inter-front selection operations. To overcome some of the shortcomings associated with existing crowding methods based on intra-selection operations, two algorithms are proposed in which the selection mechanism is determined by inter-front selection operations.

Since most MMOEAs measure solution quality in terms of Pareto dominance, we focus our research primarily on this type of MMOEAs. In the past few years, it has been possible to estimate the density of crowds in the search space by using several crowding distance methods. It is our intention to examine some of the drawbacks of the crowding distance method in this chapter.

We propose a novel classification of Pareto-dominance-based algorithms for environmental selection into two categories: (I) **intra-front selection operations** and (II)**inter-front selection operations** [63].

(I) It is possible to measure density using the neighboring solutions located on the same front by determining the crowding distance between each solution based on the neighboring solutions located on the same front of the solution. According to the crowding value calculation, a solution can appear to be far away from other solutions in the decision space when it may actually be located near many other solutions on the previous front [63]. Thus, even if the area has already been discovered by solutions on previous fronts, the solution has a better chance of surviving and passing on to the next generation.

(II) Another approach to measuring diversity involves taking into account not only the solutions within the same front of the search space but also the solutions in the vicinity of preceding fronts. By considering the solutions in both the current front and its neighboring fronts, this method provides a more accurate calculation of the crowding values of the solutions. This enhanced diversity measurement approach is described in detail in [63].

The experimental results obtained from the proposed algorithms were compared with other state-of-the-art algorithms and other algorithms proposed by the author that are implemented for many MMOPs. The main content of this chapter is taken from three publications of the author of this thesis: [63] [34], and [119].

5.1 Intra-front Selection Operations in Multi-modal Multi-objective Optimization Problems

Most Pareto-dominance-based MMOEAs are based upon the NSGA-II algorithm [33], which is considered one of the most commonly used Pareto-dominance-based MOEAs [134]. In the current population, Pareto dominance is used to evaluate fitness as the primary criterion. A non-dominated solution has a higher fitness value than dominated solutions. Thus, it is more likely to survive and be passed on to future generations. Following this point, diversification is considered a secondary criterion for selection [63].

The Pareto-dominance-based MMOEAs calculate the crowding distance value of the solutions in the search space in order to maintain a good distribution of solutions. This value is calculated using a similar general structure as in NSGA-II in the objective space: for each solution in the Front_i, the mean distance between two adjacent solutions on the left and right sides of the solution is calculated. Calculating the crowding distance for each solution is done by summing these distances [63].

This method of calculating the crowding value, which solely considers solutions within the same front and ignores the influence of solutions from other fronts, is referred to as the **intra-front selection operation**. Many existing MMOEAs utilize the crowding distance approach in the decision space to enhance the diversity of the population, as observed in studies such as [126, 101, 100].

Figure 5.1 depicts an example of a visualized calculation of the crowding distance for intra-front selection operations to better demonstrate the concept.

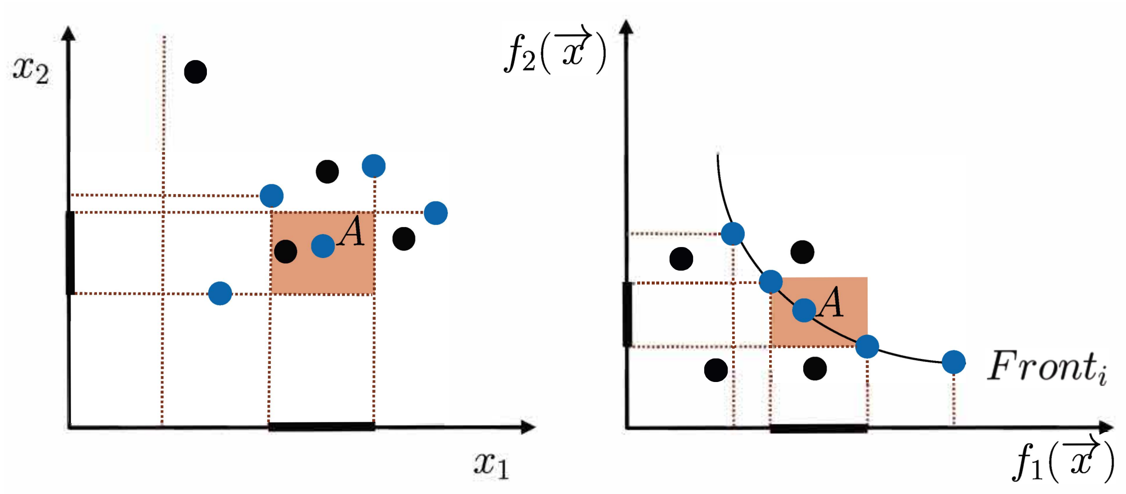


Figure 5.1: An illustration of the measurement of crowding distance values in the intra-front selection procedure [63].

According to Figure 5.1, the crowding value of the solution A is calculated by considering the effects of the closest solutions on both sides of the solution in the same front rather than other neighboring solutions from other fronts, both in the search space and objective space. The volume of the orange-highlighted regions indicates the crowding value of solution A in the search space as well as the objective space. This example clearly illustrates the concept of an intra-front selection operation [63].

5.1.1 The Challenges of Crowding Distance Approach

The first limitation is that there are more neighboring solutions in the decision space than in the objective space, making calculating the crowding distance in the decision space more challenging. Consider two objectives and two decision variables in a problem. The crowding distance along a non-dominated front in the objective space can be calculated using two neighboring solutions. In the search space, however, there can be up to four (i.e., 2^n) neighboring solutions

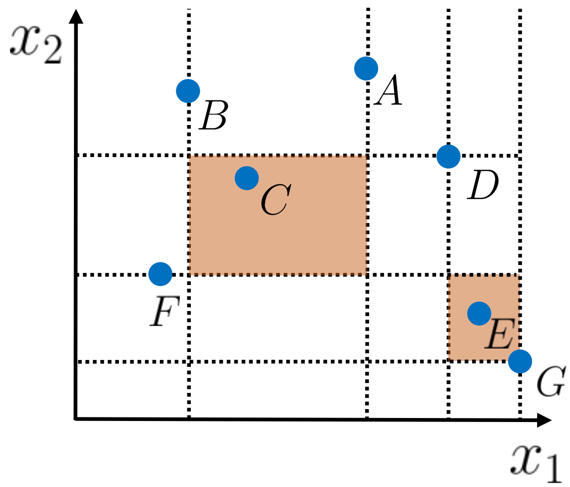


Figure 5.2: An example of how crowding distance is measured, for two solutions within a decision space [63].

for the same non-dominated solution An example of measuring the crowding distance in the search space is illustrated in Figure 5.2. The crowding distance calculated for C is calculated using four solutions: F, B, D, and A, and the crowding distance calculated for E is calculated using three solutions: F, D, and G. As we can see, C has a higher crowding distance value than E, even if solution C is near solution B; this is because the crowding distance for C is heavily influenced by the solution A [63].

Another shortcoming of the crowding distance calculation in the decision space is that the overlap of distinct PSs in the search space creates the illusion that the solution is located in a dense area, so it is excluded from selection using crowding distance, even though it is essential to preserve solutions that enable us to explore uncovered areas in the decision space. Figure 5.3 shows the aforementioned drawback when using the MMF4 test problem [63]. Despite being located in a sparse space and being the only solution covering PS_2's left side, solution A is still considered near the other solutions when crowding distances are calculated. This issue arises from the fact that the PSs are overlapped in both dimensions of the search space, making it appear crowded. As a consequence, if we use crowding distance as the secondary selection criterion, we eliminate these solutions from the search process and lose the opportunity to search for solutions that are optimal in the local area [63].

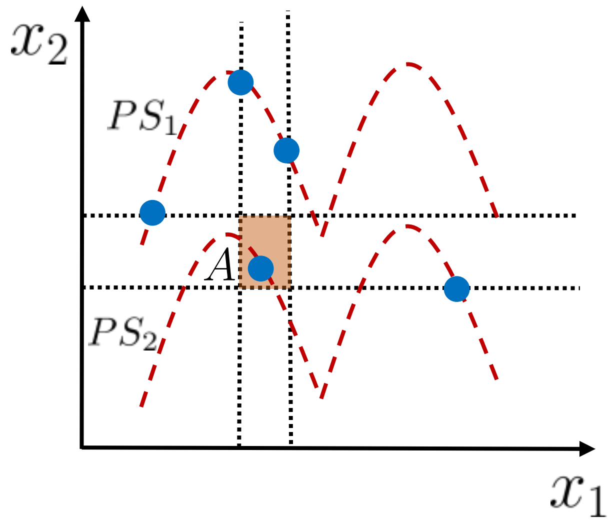


Figure 5.3: An example of the limitation of crowding distances when estimating density [63].

An alternative solution to these problems involves developing a selection methodology that uses Euclidean distances among neighboring solutions on the same front to determine the best solution. The presented selection strategy is called the Euclidean-based Selection Evolutionary Multi-Modal Multi-Objective (ES-EMMO) algorithm.

Another disadvantage of crowding distance measurements—or any selection method that measures density by looking at neighboring solutions on the same front—is that density calculations do not consider the neighboring solutions on previous fronts and therefore are inaccurate.

We can see an example of this problem in Figure 5.4. Crowding distance is calculated based on neighboring solutions located on the same front. According to the figure, solution A has a larger crowding distance value since it has a greater distance from the other solutions in the same front in the search space (i.e. B, C, and D), and the volume of the orange highlighted region denotes the crowding value of solution A. Using the crowding distance metric therefore increases the chance of selecting the solution A that will survive and transfer to the next population. In contrast, it does not improve the distribution of the solutions since the same area is already covered by some solutions from Front_1 to Front_{i-1}.

An alternative solution to the mentioned problems involves developing a selec-

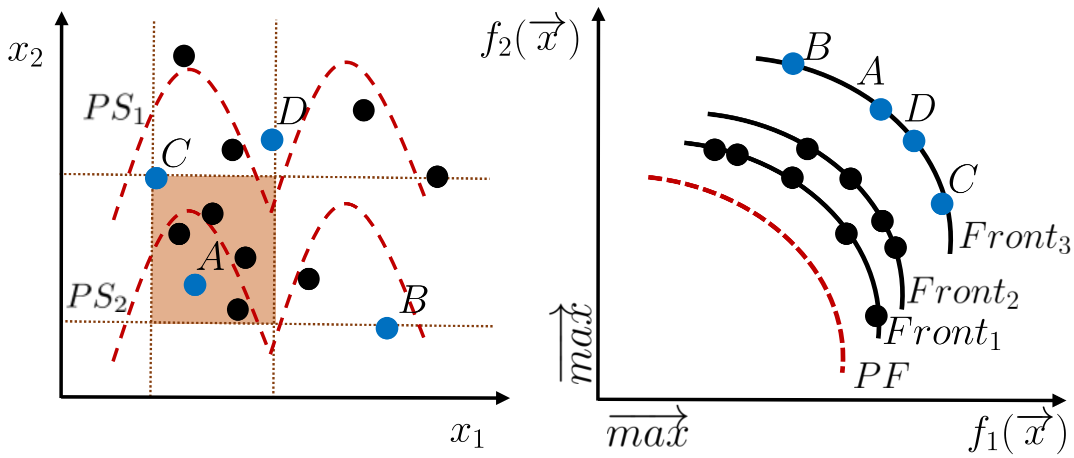


Figure 5.4: An example of crowding distance and making the illusion that the solution A is located in a sparse area while it actually is not, by ignoring the effects of other nearby solutions on previous fronts [63].

tion methodology that uses Euclidean distances among neighboring solutions on the same front to determine the best solution. The presented selection strategy is called the Euclidean-based Selection Evolutionary Multi-modal Multi-objective (ES-EMMO) algorithm.

5.1.2 ES-EMMO Algorithm

In this section, we propose the ES-EMMO algorithm, which has a modified environmental selection from the original NSGA-II. The goal is to preserve solutions that cover several PSs. This is because in MMOEAs, we aim to avoid removing solutions that are near each other in objective space but far away in search space, which can represent different PSs. Considering this concept and aiming to give higher chances to solutions located near one another in the objective space but far enough apart in the search space, we employ the Euc_{xy} metric to measure each solution, This metric serves as a basis for the environmental selection procedure [63].

Figure 5.5, illustrates an example of the Euc_{xy} measurement for solution A to make it easier to comprehend the concept. In the right figure, d_{AB} and d_{AD} represent the Euclidean distances between solution A and its neighboring solutions B and D, respectively. The left figure shows the distance between solution A and these solutions in the decision space, with d'_{AB} and d'_{AD}

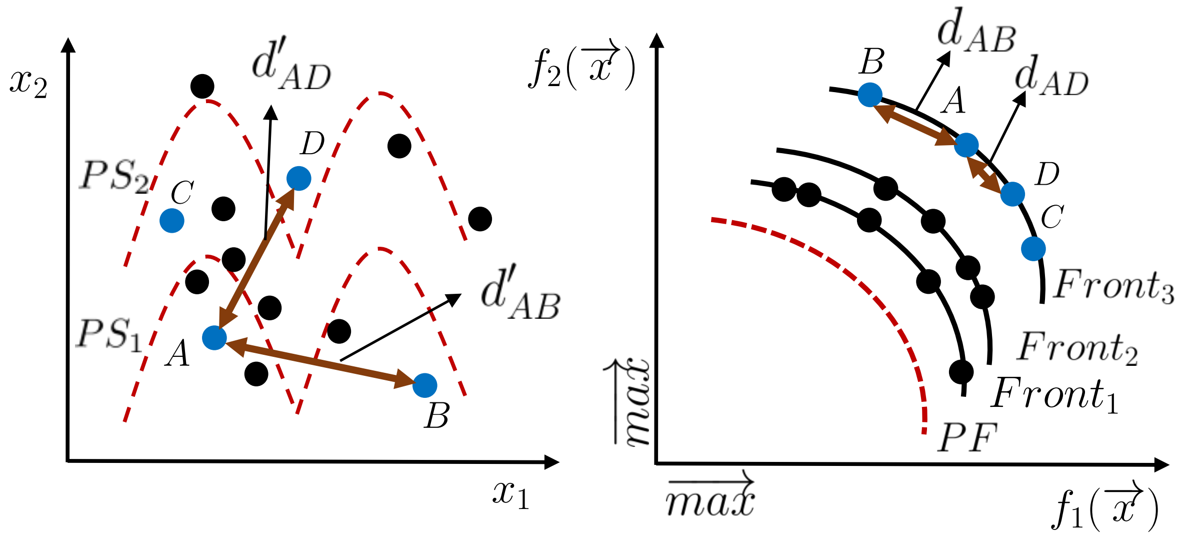


Figure 5.5: An example of the concept of Euc_{xy} approach [63].

representing the corresponding Euclidean distances. A solution's final crowding value is determined by multiplying its distance from its neighboring solution B in the search space and objective space and then adding the distance between that solution and its neighboring solution D in the search and objective spaces.

In general, the Euc_{xy} for each solution x_u is calculated as follows:

$$\text{Euc}_{xy}(x_u) = \sum_{x_v \in \text{NB}(x_u)} \text{Euc}(f_u, f_v) \cdot \text{Euc}(x_u, x_v) \tag{5.1}$$

where $\text{NB}(\text{x}_\text{u})$ contains all the x_u's adjacent solutions on both sides of its corresponding front based on each objective function, while $\text{Euc}(\text{f}_\text{u}, \text{f}_\text{v})$ and $\text{Euc}(\text{x}_\text{u}, \text{x}_\text{v})$ represent the Euclidean distances between x_u and its neighbor solution x_v in the objective space and decision space.

5.2 Inter-front Selection Operations in Multi-modal Multi-objective Optimization Problems

As a result of considering both solutions on the same front (Front_i) and neighboring solutions on previous fronts (i.e. (Front_i) to Front_{i-1}), we propose the second classification of the environmental selection operations for Pareto-dominated-based MMOEAs. In our study, we refer to these types of selection mechanisms as **inter-front selection operations**. Accordingly, the next section presents two novel algorithms that use an alternative method of environment selection based on inter-front selection.

5.2.1 NxEMMO Algorithm

The neighborhood-based evolutionary multimodal multi-objective optimization (NxEMMO) algorithm [103] is developed based on the NSGA-II and includes several modifications. An overview of the NxEMMO algorithm can be found in Algorithm 12.

The population $P(t)$ is initialized at generation t with N random individuals (Lines 1–2). The solutions are evaluated (Line 3), and the parents are then determined using a mating selection operator (Line 5). Offspring Q are generated using the simulated binary crossover (SBX) operator and mutated by the polynomial mutation operator [45] (Lines 6–7). On the basis of the max-min normalization techniques, the solutions are normalized in the search space, after which the modified environmental selection mechanism is applied (Line 10).

The proposed environmental selection mechanism (Algorithm 13) is performed on the combination of P and Q. The algorithm starts by applying non-dominated sorting as in NSGA-II and then sorting the solutions into several fronts, each denoted by Front_i for its first front (Line 1). As with NSGA-II, the solutions from Front_1 to Front_{i-1} are passed on to the new population $(P(t+1))$ (Lines 2–7). It is necessary to truncate the solutions in Front_i if they do not fit the new population.

The major difference between the NxEMMO and NSGA-II algorithms is its truncation approach. NxEMMO has a new crowding distance mechanism which replaces the one from NSGA-II.

There are two cases: 1) The Nearest Neighbor Distance (NND) mechanism is used if the front selected for truncation is itself Front_1. This diversity estimated measurement was originally proposed by Zitzler et al. [112], which is designed to keep the size of a set of solutions to a predefined value. The operation is named Omission (line 10). 2) we perform the Harmonic Average Distance (HAD) for truncation but different from the crowding distance in NSGA-II, we compute HAD between every single solution in Front_i and all other solutions in Front_1 to Front_{i-1}. In other words, we set l in Equation 4.1 as the number of solutions in Front_1 to Front_{i-1}. Addition is referred to as this mechanism (line 8). Through this operator, the solution with the highest HAD value is transferred to the new population ($P(t+1)$). Once the HAD values for the remaining solutions in Front_i have been updated, the next individuals will be selected iteratively until $R(t+1)$ has been filled up [63]. The HAD (Harmonic Average Distance) value is computed by measuring the distance between a solution i and its k-nearest neighbors. The calculation of the HAD value can be performed using the formula provided in Equation 4.1.

The neighborhood size (i.e. k) is calculated as follows:

$$k = \left\lfloor \sqrt{l} \right\rfloor \tag{5.2}$$

Referring to Figure 5.2, solution B and F are two nearest neighbors to solution C, while G and D are two neighbors to solution E. Let's assume $d_{BC} = 1$, $d_{CF} = 3$, $d_{EG} = 2$ and $d_{ED} = 4$. As a result, the harmonic average distance is smaller for C than E, i.e. HAD(C) is smaller than HAD(E). Though, these solutions have an opposite relationship in the case of crowding value.

Referring to Figure 5.2, we observe that solutions B and F are the two closest neighbors to solution C, while solutions G and D are the two nearest neighbors to solution E. Considering the given distance values, such as $d_{BC} = 1$, $d_{CF} = 3$, $d_{EG} = 2$, and $d_{ED} = 4$, it can be concluded that the harmonic average distance (HAD) is smaller for solution C compared to solution E. This indicates that HAD(C) is smaller than HAD(E). However, it is worth noting that these solutions exhibit an opposite relationship when it comes to crowding value.

An example from the MMF1 test [101] in Figure 5.6 (left) illustrates the influence of the addition function on the diversity of solutions in the decision space. Solutions in Front_1 to Front_{i-1} can be found as . A blue $\Diamond$ is used to

Input: Optimization Problem, Search Space S, Population Size N

$t \leftarrow 0$;
$P(t) \leftarrow$ InitPop($P(t)$);
Evaluate($P(t)$);
while *Termination criteria is not fulfilled* **do**
  $P_{mate} \leftarrow$ Select($P(t)$);
  $P' \leftarrow$ Recombine(P_{mate});
  $Q \leftarrow$ Mutate(P');
  Evaluate(Q);
  $t \leftarrow t + 1$;
  $P(t) \leftarrow$ Modified Environmental Selection(P', Q) //Algorithm 13;
end

Output: Final population P

Algorithm 12: NxEMMO Algorithm. Pseudocode based on [103]

Input: Population: $P(t)$, Offspring Population: Q

$Front \leftarrow$ fast-non-dominated-sort($P(t) \cup Q$);
$//Front = (Front_1, Front_2, ...)$
$i = 1$
$U = \emptyset$ **while** $|U| + |Front_i| < N$ **do**
  $U \leftarrow U \cup |Front_i|$;
  $i = i + 1$;
end
if $i \geq 2$ **then**
  $P(t+1) \leftarrow$ Addition($Front_i, U$)
else
  $P(t+1) \leftarrow$ Omission($Front_i$)
end

Output: Population $P(t+1)$

Algorithm 13: Modified Environmental Selection: ModifiedEnvSelection. Pseudocode based on [103]

identify the solutions in Front$_i$. Those solutions in the red circle are the results of the addition function. In addition, the selected solutions are represented by a purple □ based on the crowding distance method. The addition function only selects those solutions that are located in sparse areas, considering all solutions, not just those in Front$_i$ but also all other solutions from Front$_1$ to Front$_{i-1}$. Comparatively, the crowding distance method does not select solutions evenly distributed across the decision space.

Figure 5.6 (right) depicts an example of the omission function based on the

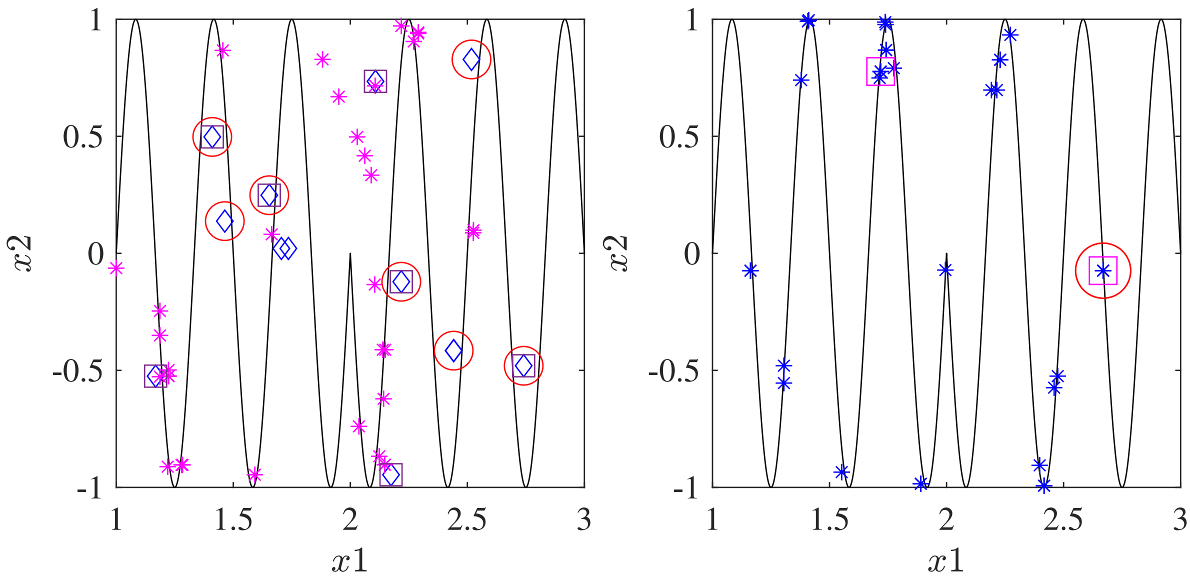


Figure 5.6: Using the test problem MMF1, an example of addition (left) and omission (right) functions [63].

NND mechanism. When only one front exists, this function is activated, which means the truncation occurs in $Front_1$. We seek to omit two solutions in this example. A blue $*$ indicates non-dominated solutions, while a $\square$ shows the result of omission using the NND function. In Figure 5.6, two solutions (duplicates) occupy the same position (marked by the red circle). One of the duplicate solutions and another solution in the crowded area are eliminated using the NND function [63].

Taking the HAD function as opposed to NND gives a better comparison of the results of the omission function. HAD's results are represented by red circles. HAD selected the above duplicates when selecting two solutions for omission. HAD results in the elimination of both duplicates, leaving the empty position in that part of the decision space, whereas NND maintains one of the duplicates at the same position (the red circle) and eliminates the other in a crowded area [63].

5.2.1.1 Configuration of the Proposed Methods

In this research, we conduct a comparative analysis between two of our proposed algorithms, NxEMMO (utilizing inter-front selection operations) and ES-EMMO (also employing inter-front selection operations), and other competing algorithms that are based on similar inter-front operations. The

Table 5.1: A comparison of the IGDx values obtained by different algorithms [63].

	inter-front operations	intra-front operations			
Problems	NxEMMO	ES-EMMO	NSGA-II-CD_{ws}	DN-NSGA-II	Omni-optimizer
MMF1	0.06826 (0.011957)	0.069926 (0.004074)	0.073512* (0.005277)	0.14906* (0.037849)	0.13652* (0.023222)
MMF1z	0.054599 (0.013128)	0.060952* (0.008315)	0.060436* (0.012492)	0.12516* (0.041036)	0.10905* (0.037024)
MMF2	0.086632* (0.060714)	0.099559* (0.099973)	0.068848 (0.041494)	0.15677* (0.096484)	0.12076* (0.09012)
MMF3	0.072191 (0.042195)	0.060297 (0.044795)	0.071141 (0.056237)	0.11243* (0.055217)	0.10998* (0.04382)
MMF4	0.034715 (0.003061)	0.1522* (0.041193)	0.053797* (0.010829)	0.13763* (0.049653)	0.1392* (0.046631)
MMF5	0.56786 (0.013278)	0.57518* (0.026768)	0.59996* (0.011188)	0.6072* (0.040392)	0.5984* (0.039829)
MMF6	0.09557 (0.011502	0.13045* (0.016892)	0.11779* (0.010055)	0.19174* (0.027151)	0.20344* (0.028455)
MMF7	0.036965 (0.003481	0.042214* (0.009148)	0.040405* (0.006513)	0.075119* (0.015421)	0.060416* (0.012387)
MMF8	0.26061* (0.11824)	0.96612* (0.086946)	0.20274 (0.10151)	0.50743* (0.37566)	0.48434* (0.37711)
MMF9	0.007537 (0.000432)	0.019128* (0.010262)	0.024772* (0.038961)	0.035751* (0.019869)	0.030939* (01914)
MMF14	0.007748 (0.000466)	0.010298* (0.001504)	0.036652* (0.050487)	0.03771* (0.021226)	0.032523* (0.022532)
SYM-$PART_{simple}$	0.064747 (0.006078)	3.3772* (1.1078)	2.2426* 2.299	7.0069* (2.2805)	7.0481* (1.9072)
SYM-$PART_{rotated}$	2.2081 (1.4778)	3.2888* (1.8333)	3.004 ((1.0614)	6.1886* (2.4087)	6.7457* (2.5533)
Omni-test	0.035064 (1.4778)	0.434055* (0.343155)	0.06756* (0.025421)	0.848054* (0.310415)	0.99567* (0.285056)

selected algorithms for comparison include Omni-optimizer [100], DN-NSGA-II [101], NSGA-II-CDws, and our proposed ES-EMMO algorithm.

5.2.1.2 Discussion of Results

In Tables 5.1,5.2, and5.3, the best-performing algorithms are indicated in bold font. Additionally, an asterisk (*) is used to denote significant statistical differences compared to the best-performing algorithms. The tables present the algorithms categorized based on their selection methods. Our objective is to compare the performance of the proposed algorithms with other algorithms, as well as to evaluate the performance of the two selection operations against each other [63].

Table 5.2: A comparison of the PSP values obtained by different algorithms [119].

Problems	inter-front operations	intra-front operations			
	NxEMMO	ES-EMMO	NSGA-II-CD$_{ws}$	DN-NSGA-II	Omni-optimizer
MMF1 (2,2)	16.9617 (1.2649)	14.632* (2.8334)	16.2399* (1.7222)	13.5223* (1.1236)	8.4532* (1.7456)
MMF1z (2,2)	21.2536 (4.6165)	18.1833* (4.5993)	21.3127 (3.7551)	16.2743* (3.4372)	8.1724* (1.5981)
MMF2 (2,2)	13.6626 (8.322)	10.7635* (5.6645)	11.2419 (6.687)	13.2636 (8.692)	9.9459 (7.8377)
MMF3 (2,2)	15.2512 (8.8569)	11.3401 (10.1311)	15.3887 (8.5256)	11.4225 (14.095)	12.4462 (9.9079)
MMF4 (2,2)	30.1475 (1.728)	28.6876* (2.5845)	29.5983 (2.3712)	18.4703* (3.5828)	9.1544* (3.1222)
MMF5 (2,2)	1.2524 (0.011468)	1.2386* (0.030162)	1.2288* (0.02723)	1.1609* (0.034616)	1.1292* (0.06761)
MMF6 (2,2)	11.4149 (0.63732)	10.3903* (1.2134)	11.2765 (1.0371)	8.3961* (0.68611)	4.9954* (1.6075)
MMF7 (2,2)	26.7459 (2.5599)	26.5611* (3.6799)	29.0313 (5.1569)	23.93* (3.5717)	13.775* (4.1412)
MMF8 (2,2)	7.0477 (2.6601)	3.5652* (1.5098)	6.1924 (4.2528)	4.804* (2.9008)	1.0041* (0.39043)
MMF9 (2,2)	121.1216* (6.025)	132.6703 (7.5858)	81.3375* (15.4764)	40.368* (46.8601)	10.7495* (34.3256)
Omni-test (2,2)	28.3315* (1.2741)	28.4204* (2.151)	32.689 (2.1142)	14.5811* (6.0094)	3.4256* (8.4479)
SYM-PART (2,2)	14.4054* (1.2994)	15.4253* (1.4269)	16.3943 (1.2325)	0.44329* (0.66802)	0.15392* (0.14778)
Polygon (6,2)	5.654 * (0.16177)	6.3247 * (0.13203)	6.5461 (0.14536)	4.5685 * (0.41006)	4.4309 * (0.31134)
Polygon (6,4)	1.527 (1.5023)	0.53428 * (1.4851)	0.51441 * (1.5042)	0.52719 * (1.2955)	0.42006 * (1.0601)
Polygon (6,6)	0.10114 (0.07837)	0.044371 * (0.074686)	0.042089 * (0.061189)	0.044769 (0.069666)	0.043487 * (0.066217)
Best/All	8/15	1/15	6/15	0/15	0/15
Wins/Losses/Ties	–	1/11/3	4/4/7	0/12/3	0/13/2

In Tables 5.1 and 5.2, we can see that the NxEMMO algorithm outperforms other algorithms in 11 out of 14 test instances. As expected, these results meet our expectation that modifications to the environmental selection and replacing the crowding distance in the NxEMMO algorithm will lead to improved results. When NxEMMO is used, it is possible to detect the solutions in sparse areas in the decision space more effectively than with the crowding distance method used in the algorithm that utilizes inter-front selection operations. Algorithm 13 describes how adding and omitting in subsequent steps leads to enhanced PS approximation. Moreover, as we see in Tables 5.1 and 5.2, ES-EMMO performs the best among the three algorithms on the MMF3 test case that involves local PS. It appears that DN-NSGA-II and Omni-optimizer algorithms

Table 5.3: A comparison of the IGD$^+$ values obtained by different algorithms [119].

Problems	inter-front operations	intra-front operations			
	NxEMMO	ES-EMMO	NSGA-II-CD$_{ws}$	DN-NSGA-II	Omni-optimizer
MMF1 (2,2)	16.9617 (1.2649)	14.632* (2.8334)	16.2399* (1.7222)	13.5223* (1.1236)	8.4532* (1.7456)
MMF1z (2,2)	21.2536 (4.6165)	18.1833* (4.5993)	21.3127 (3.7551)	16.2743* (3.4372)	8.1724* (1.5981)
MMF2 (2,2)	13.6626 (8.322)	10.7635* (5.6645)	11.2419 (6.687)	13.2636 (8.692)	9.9459 (7.8377)
MMF3 (2,2)	15.2512 (8.8569)	11.3401 (10.1311)	15.3887 (8.5256)	11.4225 (14.095)	12.4462 (9.9079)
MMF4 (2,2)	30.1475 (1.728)	28.6876* (2.5845)	29.5983 (2.3712)	18.4703* (3.5828)	9.1544* (3.1222)
MMF5 (2,2)	1.2524 (0.011468)	1.2386* (0.030162)	1.2288* (0.02723)	1.1609* (0.034616)	1.1292* (0.06761)
MMF6 (2,2)	11.4149 (0.63732)	10.3903* (1.2134)	11.2765 (1.0371)	8.3961* (0.68611)	4.9954* (1.6075)
MMF7 (2,2)	26.7459 (2.5599)	26.5611* (3.6799)	29.0313 (5.1569)	23.93* (3.5717)	13.775* (4.1412)
MMF8 (2,2)	7.0477 (2.6601)	3.5652* (1.5098)	6.1924 (4.2528)	4.804* (2.9008)	1.0041* (0.39043)
MMF9 (2,2)	121.1216* (6.025)	132.6703 (7.5858)	81.3375* (15.4764)	40.368* (46.8601)	10.7495* (34.3256)
Omni-test (2,2)	28.3315* (1.2741)	28.4204* (2.151)	32.689 (2.1142)	14.5811* (6.0094)	3.4256* (8.4479)
SYM-PART (2,2)	14.4054* (1.2994)	15.4253* (1.4269)	16.3943 (1.2325)	0.44329* (0.66802)	0.15392* (0.14778)
Polygon (6,2)	5.654 * (0.16177)	6.3247 * (0.13203)	6.5461 (0.14536)	4.5685 * (0.41006)	4.4309 * (0.31134)
Polygon (6,4)	1.527 (1.5023)	0.53428 * (1.4851)	0.51441 * (1.5042)	0.52719 * (1.2955)	0.42006 * (1.0601)
Polygon (6,6)	0.10114 (0.07837)	0.044371 * (0.074686)	0.042089 * (0.061189)	0.044769 (0.069666)	0.043487 * (0.066217)
Best/All	8/15	1/15	6/15	0/15	0/15
Wins/Losses/Ties	–	1/11/3	4/4/7	0/12/3	0/13/2

are more prone to becoming trapped into local PSs.

A closer look at the interquartile results for the IGDx values in Table 5.1 reveals that in 10 out of the 14 test cases, the NxEMMO has a lower score than its competition. This means that the results obtained with the NxEMMO algorithm have better stability and robustness over several runs of the algorithm since they do not show much variation when compared to the other state-of-the-art algorithms. Moreover, when we compare the results for the NxEMMO algorithm and the proposed ES-EMMO algorithm, we can see that the NxEMMO algorithm does better in 10 out of 14 test cases based on statistical significance compared to the ES-EMMO algorithms. In comparison to the results for the MMF3 test cases, ES-EMMO did not differ statistically from NxEMMO, regardless of the better IGDx value [63].

The performance indicator for PSP consists of the ratio of cover rate to IGDx, where cover rate (or overlap rate) indicates the percentage of the defined region between the optimal solution and the PS. The results in Tables 5.1 and 5.2 are nearly identical since the overlap rate of the test results is one, which is an ideal value [63].

Based on the results in terms of intra-front selection operations, ES-EMMO outperforms the NSGA-II-CD_{ws} algorithm in five cases. In addition, in most of the test cases, it also outperformed the Omni-optimizer and DN-NSGA-II algorithms, which use crowding distance metrics on the decision space. These results confirm that using other distance metrics besides the crowding distance metric can help to overcome the abovementioned problem and boost diversification over the PSs [63].

In the Omni-test problem, the ES-EMMO, Omni-optimizer, and DN-NSGA-II algorithms demonstrate subpar performance. This can be attributed to the inadequate distribution of solutions within the search space, as these algorithms were unable to identify all 27 Pareto solutions (PS). Conversely, the NxEMMO algorithm exhibits promising outcomes in terms of preserving a significant number of PSs.

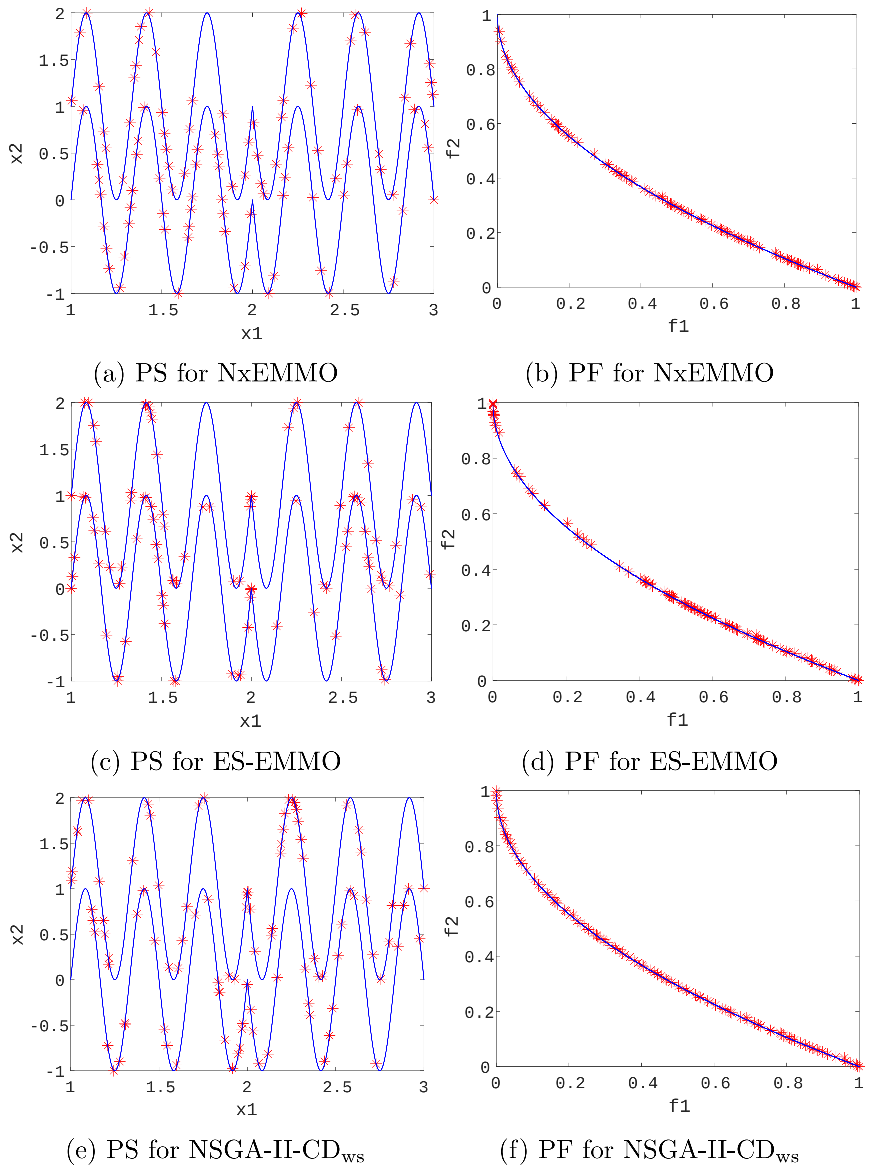
We additionally observe that NSGA-II-CD_{ws} exhibits superiority to all the other algorithms when it comes to IGD+ values (i.e., Table 5.3). We expect this algorithm to have a better approximation of PF in the objective space since it considers the diversity of the solutions in the objective space in its density calculations. According to the analysis, NxEMMO is the second-best algorithm in the objective space, after NSGA-II-CD_{ws}. Both the Omni-optimizer and DN-NSGA-II algorithms perform poorly when compared to other algorithms in the category of intra-front selection. All in all, as expected, considering the effects of neighboring solutions further improves the density estimation of the solutions within the search space. Therefore, inter-front selection operations generally outperform intra-front selection operations.

To provide better visualization of the result population and demonstrate the similarity between the obtained final solutions in both decision and objective spaces, Figures 5.7, 5.8, and 5.9 present the results of the run with the median IGDx performance indicator for the MMF6, MMF9, and SYM-PART$_{simple}$ test cases. A solid gray line corresponds to the actual PF and PS of the test problems, while the red marker represents the solutions obtained using computational algorithms. We compare the results for the NxEMMO algorithm, a representation of an inter-front selection operation, as well as ES-EMMO and NSGA-II-CD$_{ws}$, a representation of an intra-front selection operation. On closer inspection of Figures 5.7, 5.8, and 5.9 in terms of the search space, the obtained results for the algorithm NxEMMO on the MMF6, MMF14, and SYM-PART$_{simple}$ test cases indicate that this algorithm has a better coverage area over the PS than the other two, and the results are distributed more evenly over the PS for those test problems. We can see, for example, that in the SYM-PART$_{simple}$ test case, the NxEMMO algorithm succeeds in finding and preserving all PSs, whereas other state-of-the-art algorithms are able to locate only some PSs. Furthermore, we can see that the NxEMMO algorithm is capable of providing reasonable coverage of the PF.

5.2.2 MMEA-HAD Algorithm

The main content of this section has been taken from the original paper by the author [119]. In this section, we propose our new approach called Multi-modal Multi-objective Evolutionary Algorithm using Harmonic Average Distance (MMEA-HAD) which follows the same general outline similar to typical evolutionary algorithms, such as the NSGA-II algorithm [33].

In MMEA-HAD, we incorporate modifications to both tournament and environmental selection mechanisms, which are important elements for dealing with MMOPs. We performed an experiment on a test problem (MMF1) using the NSGA-II algorithm (with standard tournament selection) containing 100 individuals and, overall, 10000 function evaluations. In 100 generations, we found that, on average, in each generation, 4.145% of the offspring were exact duplicates of their parents. This number of duplicates can have a considerable impact on the environmental selection process in EMMO algorithms, in particular the environmental selection process in our latest algorithm [34]. As illustrated in an example in [34], the duplicate solutions on the same front can be deleted by the HAD mechanism, leaving an empty position in that



(a) PS for NxEMMO (b) PF for NxEMMO

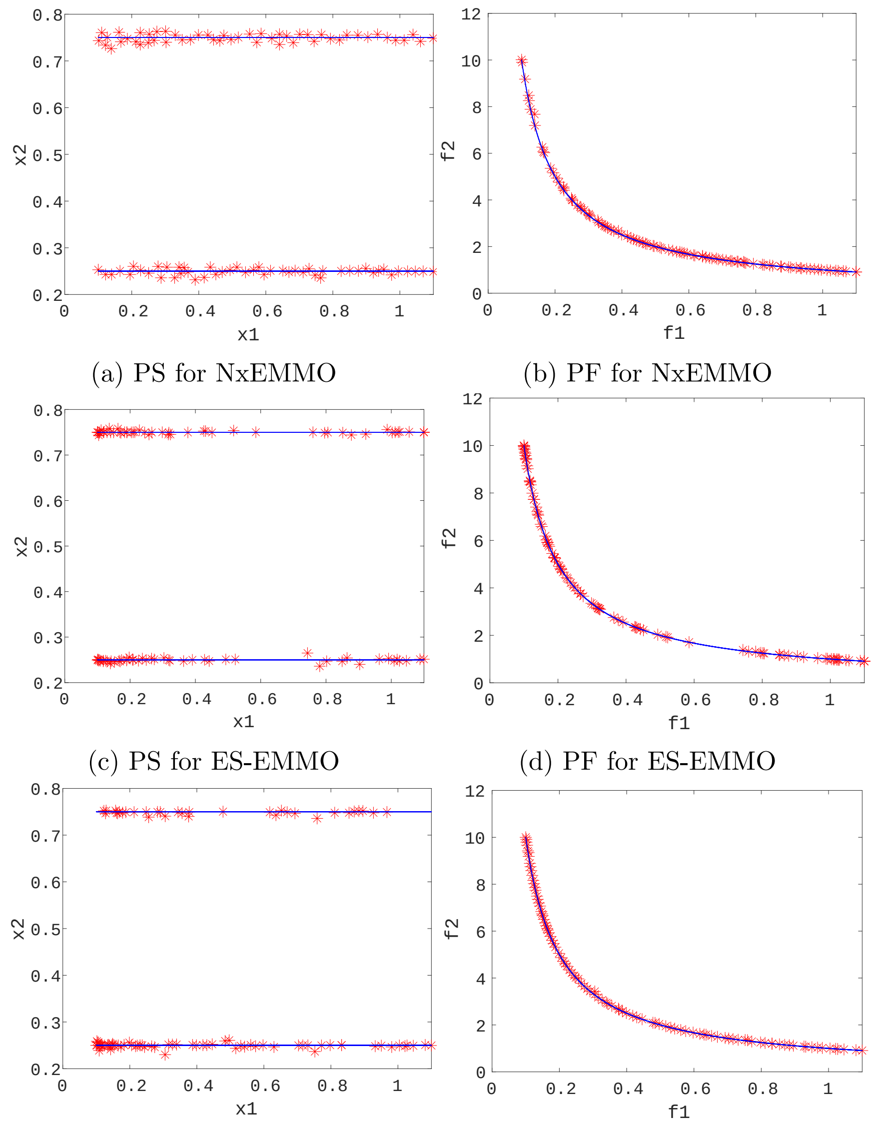
(c) PS for ES-EMMO (d) PF for ES-EMMO

(e) PS for NSGA-II-CD_{ws} (f) PF for NSGA-II-CD_{ws}

Figure 5.7: Obtained solutions in both decision and objective spaces for MMF6 test problem [63]

part of the decision space, while the proposed NND mechanism allows one duplicate to remain at the same location. In this paper, we aim to avoid selecting duplicate solutions by exploiting the full potential of the parental population to generate diverse solutions in the search space. The framework for the proposed The MMEA-HAD algorithm is shown in Algorithm 14.

After initializing the population randomly (Line 1), solutions are evaluated. The following steps are implemented on the population until the termination criteria are met (Lines 4–8). As presented in Algorithm 2, the modified mating



(a) PS for NxEMMO (b) PF for NxEMMO

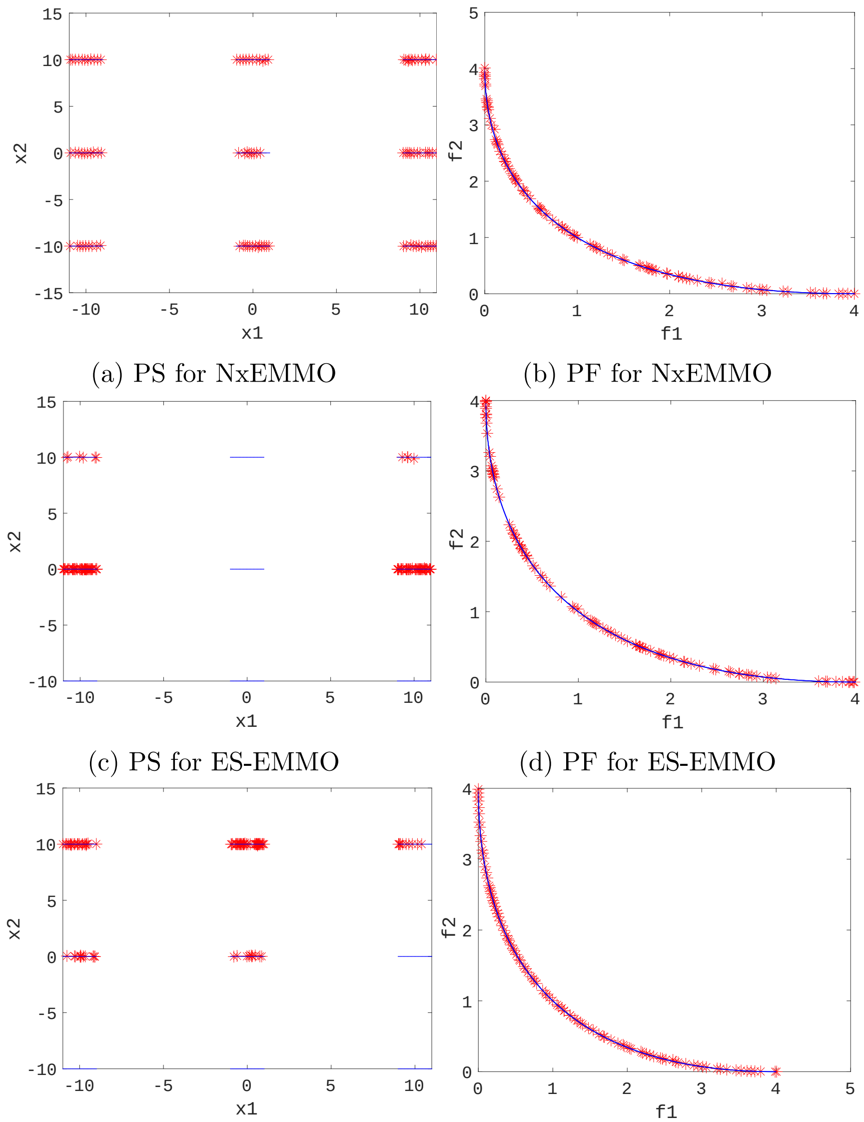
(c) PS for ES-EMMO (d) PF for ES-EMMO

(e) PS for NSGA-II-CD$_{ws}$ (f) PF for NSGA-II-CD$_{ws}$

Figure 5.8: Obtained solutions in both decision and objective spaces for MMF9 test problem [63]

selection criteria are applied to the populations, and the modified reproduction operator is then applied to the parent populations to produce the offspring (Line 5). As soon as the offspring are produced, the results are evaluated (Line 6). Following this, the modified environmental selection mechanism (represented in Algorithm 15) is applied to the combination of the parents and the offspring to make a new population for the next generation (Line 7).

We propose a modified environmental selection mechanism in Algorithm 15. In the case of multiple fronts of solutions, an addition operator is applied



(a) PS for NxEMMO (b) PF for NxEMMO

(c) PS for ES-EMMO (d) PF for ES-EMMO

(e) PS for NSGA-II-CD_{ws} (f) PF for NSGA-II-CD_{ws}

Figure 5.9: Obtained solutions in both decision and objective spaces for SYM-PART$_{simple}$ test problem [63]

to select the solutions in sparser parts of the search space. For each solution at $Front_i$, the HAD value between that solution and every solution in its neighborhood from $Front_1$ to $front_{i-1}$ is calculated. Neighborhood size is defined as the square root of the number of solutions from $Front_1$ to $front_{i-1}$. Afterward, the solution with the largest HAD value is transferred to the new population $R(t+1)$. The iterative process of selecting the next individual and updating the remaining population's harmonic average distance (HAD) values continues until all N possible solutions are chosen (Lines 7-9). As the

```
Input: Population Size N, Search Space S, Max Number of Generations G_max,
Output: Final Population P_final
t = 1 ;
Set up population P(t) with N individuals;
Evaluatation of (P(t));
while t ¡ G_max do
    Q(t) ← Modifed mating Selection and the Reproduction Operatoras (P(t)) ;
     // Algorithm 2
    Evaluatation of (Q(t));
    P(t+1) ← Modified Environmental Selection (P(t), Q(t));        // Algorithm 15
end
return P_final
```

Algorithm 14: General framework for MMEA-HAD algorithm. The pseudocode is based on [119]

solutions approach the Pareto front (PF) in later generations, they may be situated in the first front. In such instances, the omission operator is applied. This involves iteratively removing the solutions with the highest HAD until the population size is reached (Lines 9-11).

The MMEA-HAD is further combined with the tournament selection method proposed in Section 4.1.1 to make an intra-front selection operator that enhances the exploration of solutions.

5.2.2.1 Discussion of Results

Tables 5.4, 5.5, and 5.6 offer a detailed description of IGD+, PSP, and IGDx results. According to the 31 independent runs of the algorithms for every problem, we compute the median and interquartile range (IQR) for each of the performance indicators IGD^+, IGDx, and PSP. An analysis of the statistical difference between the results according to the best algorithm is conducted using a Mann-Whitney U test with a significance level of $\alpha = 0.05$, reflecting the statistical significance of the conclusions. Furthermore, we demonstrate the wins, losses, and ties for each algorithm over the proposed algorithm across all test suites.

In Tables 5.4, 5.5, we compare the results of the algorithms on the basis of IGDx and PSP, which represent their performance in the decision space. Throughout the 15 test problems, the performance of the proposed MMEA-HAD algorithm beats the performance of its previous algorithm (NxEMMO

Input: Population: $P(t)$, Offspring Population: $Q(t)$
Output: Population $P(t+1)$
$Front \leftarrow$ fast-non-dominated-sort($P(t) \cup Q$)
$i = 1$;
while $|P| + |Front_i| \leq N$ **do**
    $P(t+1) \leftarrow P(t) \cup |Front_i|$;
    $i = i + 1$;
end
if $i \geq 2$ **then**
    $P(t+1) \leftarrow$ Addition($Front_i, P(t+1)$)
else
    $P(t+1) \leftarrow$ Omission($Front_i$)
end
Output: Population $P(t+1)$
return $P(t+1)$

Algorithm 15: Modified Environmental Selection: ModifeiEnvironSelect. The pseudocode is based on [34, 119]

algorithm) in seven out of 15 test cases, especially in the higher-dimensional search and objective spaces. TThe utilization of the modified selections of tournament and environment simultaneously in the new algorithm leads to the generation of solutions that are both diverse and convergent along the PS. Consequently, the proposed method achieves a more accurate approximation of the PS in all test cases compared to the baseline algorithm NSGA-II.

One of the reasons that our proposed algorithm performs better than other algorithms could be that it utilizes a tailored tournament selection method, which eliminates duplicate solutions within the population. To thoroughly investigate this aspect, we performed additional research and carried out experiments involving all test functions. The experiments were conducted with 100 individuals and 10,000 function evaluations, utilizing the proposed MMEA-HAD algorithm. Remarkably, across all generations, we consistently observed that the modified mating selection mechanism successfully eliminated the occurrence of duplicate solutions. This finding highlights the efficacy of our algorithm in ensuring a diverse population without any repeated solutions.

Considering both the PSP and IGDx performance indicators, it is evident that the proposed MMEA-HAD algorithm consistently outperformed other competitors in representing the PS in most cases (8 out of 15). Moreover, the algorithm with better average values and the proposed algorithm did not show

Table 5.4: A comparison of the IGDx values obtained via different algorithms. In (m, n), the size of the decision variable n and the number of objectives m are given. The best algorithm is highlighted in blue. The asterisk (*) indicates the statistical significance relative to the best algorithm [119].

Problems	MMEA-HAD	NxEMMO	DNEA	NSGA-II-CD$_{ws}$	NSGA-II
MMF1 (2,2)	0.058847 (0.003946)	0.06826* (0.011957)	0.061305* (0.006423)	0.073512* (0.005277)	0.11478* (0.023295)
MMF1z (2,2)	0.046656 (0.010718)	0.054599* (0.013128)	0.04677 (0.008116)	0.060436* (0.012492)	0.11967* (0.023685)
MMF2 (2,2)	0.070443 (0.039368)	0.086632* (0.060714)	0.073379 (0.037776)	0.068848 (0.041494)	0.085467 (0.064877)
MMF3 (2,2)	0.058886 (0.037601)	0.072191 (0.042195)	0.058824 (0.029222)	0.071141 (0.056237)	0.068035 (0.046307)
MMF4 (2,2)	0.033154 (0.00194)	0.034715* (0.003061)	0.033561 (0.002765)	0.053797* (0.010829)	0.10556* (0.035845)
MMF5 (2,2)	0.56402 (0.004894)	0.56786 (0.013278)	0.57407* (0.006236)	0.59996* (0.011188)	0.61874* (0.015645)
MMF6 (2,2)	0.08721 (0.004791)	0.09557* (0.011502)	0.088428 (0.007831)	0.11779* (0.010055)	0.19387* (0.057345)
MMF7 (2,2)	0.037259* (0.002659)	0.036965* (0.003481)	0.034349 (0.005445)	0.040405* (0.006513)	0.069254* (0.017539)
MMF8 (2,2)	0.13482 (0.048267)	0.26061* (0.11824)	0.15105 (0.08965)	0.20274* (0.10151)	0.7981* (0.33379)
MMF9 (2,2)	0.008256* (0.000419)	0.007537 (0.000432)	0.012294* (0.002397)	0.024772* (0.038961)	0.093028* (0.2199)
Omni-test (2,2)	0.035121* (0.001579)	0.035064* (0.002751)	0.030401 (0.001842)	0.067564* (0.025421)	0.28523* (0.41252)
SYM-PART (2,2)	0.069413* (0.006177)	0.064747* (0.006078)	0.060897 (0.004591)	2.2426* (2.299)	4.8896* (3.2094)
Polygon (6,2)	0.17687 * (0.005087)	0.15811 * (0.003301)	0.15243 (0.003306)	0.21889 * (0.019518)	0.22452 * (0.017737)
Polygon (6,4)	0.63837 (1.4507)	1.7587 * (2.5475)	1.8966 * (2.5725)	1.8092 * (2.4795)	2.0501 * (2.4416)
Polygon (6,6)	4.4861 (2.3019)	6.3022 * (2.3054)	6.2868 * (2.016)	6.2266 (2.1361)	6.2913 (2.1028)
Best/All	8/15	1/15	5/15	1/15	0/15
Wins/Losses/Ties	–	4/7/4	3/6/6	1/11/3	0/12/3

statistically significant differences in three of the test cases. With a closer look at the results, we can see that the algorithm performs better than other competitors in higher-dimensional search and objective spaces. The results demonstrate that there is an advantage to the proposed algorithms in terms of scalability.

Based on our expectations, the proposed algorithm, when emphasizing the distribution of the solutions in the search space, can lead to the loss of the distribution in the objective space over the PF. It is required to find the right balance between the diversity and convergence of the solutions in the search space and the objective space. Despite this, the DNEA algorithm was the best out of the competitors on 11 out of 15 test suits because it uses an archive

Table 5.5: A comparison of the PSP values obtained via different algorithms. In (m, n), the size of the decision variable n and the number of objectives m are given. The best algorithm is highlighted in blue. The asterisk (*) indicates the statistical significance relative to the best algorithm [119].

Problems	MMEA-HAD	NxEMMO	DNEA	NSGA-II-CD_{ws}	NSGA-II
MMF1 (2,2)	16.9617 (1.2649)	14.632* (2.8334)	16.2399* (1.7222)	13.5223* (1.1236)	8.4532* (1.7456)
MMF1z (2,2)	21.2536 (4.6165)	18.1833* (4.5993)	21.3127 (3.7551)	16.2743* (3.4372)	8.1724* (1.5981)
MMF2 (2,2)	13.6626 (8.322)	10.7635* (5.6645)	11.2419 (6.687)	13.2636 (8.692)	9.9459 (7.8377)
MMF3 (2,2)	15.2512 (8.8569)	11.3401 (10.1311)	15.3887 (8.5256)	11.4225 (14.095)	12.4462 (9.9079)
MMF4 (2,2)	30.1475 (1.728)	28.6876* (2.5845)	29.5983 (2.3712)	18.4703* (3.5828)	9.1544* (3.1222)
MMF5 (2,2)	1.2524 (0.011468)	1.2386* (0.030162)	1.2288* (0.02723)	1.1609* (0.034616)	1.1292* (0.06761)
MMF6 (2,2)	11.4149 (0.63732)	10.3903* (1.2134)	11.2765 (1.0371)	8.3961* (0.68611)	4.9954* (1.6075)
MMF7 (2,2)	26.7459 (2.5599)	26.5611* (3.6799)	29.0313 (5.1569)	23.93* (3.5717)	13.775* (4.1412)
MMF8 (2,2)	7.0477 (2.6601)	3.5652* (1.5098)	6.1924 (4.2528)	4.804* (2.9008)	1.0041* (0.39043)
MMF9 (2,2)	121.1216* (6.025)	132.6703 (7.5858)	81.3375* (15.4764)	40.368* (46.8601)	10.7495* (34.3256)
Omni-test (2,2)	28.3315* (1.2741)	28.4204* (2.151)	32.689 (2.1142)	14.5811* (6.0094)	3.4256* (8.4479)
SYM-PART (2,2)	14.4054* (1.2994)	15.4253* (1.4269)	16.3943 (1.2325)	0.44329* (0.66802)	0.15392* (0.14778)
Polygon (6,2)	5.654 * (0.16177)	6.3247 * (0.13203)	6.5461 (0.14536)	4.5685 * (0.41006)	4.4309 * (0.31134)
Polygon (6,4)	1.527 (1.5023)	0.53428 * (1.4851)	0.51441 * (1.5042)	0.52719 * (1.2955)	0.42006 * (1.0601)
Polygon (6,6)	0.10114 (0.07837)	0.044371 * (0.074686)	0.042089 * (0.061189)	0.044769 (0.069666)	0.043487 * (0.066217)
Best/All	8/15	1/15	6/15	0/15	0/15
Wins/Losses/Ties	–	1/11/3	4/4/7	0/12/3	0/13/2

in objective space, which improves the distribution of solutions in objective space.

To illustrate the performance of IGDx over generations, we show an example of the median run of the competitor algorithms for four test problems in Figure 5.10. We observe that NSGA-II loses its approach to PS in the later generations (at the end of optimization), demonstrating that non-tailored EMO Algorithms for MMOPs cannot deal with such problems. From a comparison of the MMEA-HAD algorithm with the DNEA algorithm, it is evident that the DNEA algorithm shows more fluctuation at the beginning, while the proposed algorithm maintains a steady behavior over the entire optimization procedure. As shown, the DNEA algorithm obtained a better result at the

Table 5.6: A comparison of the IGD^+ values obtained via different algorithms. In (m, n), the size of the decision variable n and the number of objectives m are given. The best algorithm is highlighted in blue. The asterisk (*) indicates the statistical significance relative to the best algorithm [119].

Problems	MMEA-HAD	NxEMMO	DNEA	NSGA-II-CD_{ws}	NSGA-II
MMF1 (2,2)	0.004699* (0.000363)	0.004407* (0.000192)	0.003128 (0.000178)	0.003693* (0.000266)	0.0039* (0.000136)
MMF1z (2,2)	0.004255* (0.000481)	0.004409* (0.000496)	0.003067 (0.000149)	0.003767* (0.000206)	0.003841* (0.000304)
MMF2 (2,2)	0.012611 (0.010245)	0.013729 (0.00909)	0.016721 (0.00882)	0.014602 (0.007049)	0.013448 (0.015575)
MMF3 (2,2)	0.010452 (0.010993)	0.011404 (0.005682)	0.011528* (0.004805)	0.009442 (0.005035)	0.009451 (0.007619)
MMF4 (2,2)	0.004711* (0.000476)	0.004546* (0.000452)	0.003054 (0.000237)	0.003513* (0.000244)	0.00347* (0.000179)
MMF5 (2,2)	0.008984* (0.001674)	0.009195* (0.000985)	0.006835 (0.00037)	0.007854* (0.001077)	0.008859* (0.00101)
MMF6 (2,2)	0.004635* (0.000623)	0.004868* (0.000527)	0.003123 (0.000172)	0.003773* (0.000238)	0.003975* (0.00029)
MMF7 (2,2)	0.005672* (0.000535)	0.004802* (0.000363)	0.003352 (0.000185)	0.003844* (0.000252)	0.003858* (0.000161)
MMF8 (2,2)	0.071423* (0.000277)	0.071495* (0.000205)	0.070858 (0.000127)	0.070937* (9e-05)	0.070967* (0.000126)
MMF9 (2,2)	0.006519* (0.000393)	0.005995 (0.000403)	0.006994* (0.000844)	0.006378* (0.000423)	0.006528* (0.000437)
Omni-test (2,2)	1.0018* (0.000384)	1.0017* (0.000348)	1.0009 (0.000216)	1.0009 (0.000139)	1.0009* (0.000309)
SYM-PART (2,2)	0.013816* (0.001603)	0.013407* (0.001858)	0.008298 (0.000218)	0.008747* (0.000575)	0.009031* (0.000653)
Polygon (6,2)	0.12473 * (0.00764)	0.10741 * (0.003583)	0.085192 (0.002205)	0.12592 * (0.009252)	0.12242 * (0.012047)
Polygon (6,4)	0.26578 * (0.027928)	0.224 * (0.019196)	0.1908 (0.009455)	0.2455 * (0.016005)	0.24559 * (0.026836)
Polygon (6,6)	0.411 * (0.040505)	0.36919 * (0.037701)	0.30212 (0.021663)	0.36442 * (0.02947)	0.37658 * (0.03799)
Best/All	1/15	1/15	12/15	1/15	0/15
Wins/Losses/Ties	–	4/2/9	11/2/2	8/3/4	8/2/5

end of optimization for the MMF7 test problem. In summary, the proposed approach has a very smooth convergence and delivers similar or even better results than the other approaches.

5.3 Summary

In this chapter, we have classified the selection operations of Pareto-based MMOEAs into two distinct categories: inter-front and intra-front selection. To make our comparisons, we have considered both the proposed algorithms and other state-of-the-art algorithms that belong to the same categories. Through extensive experiments, we have observed that the proposed inter-front selection method outperforms algorithms utilizing alternative selection approaches.

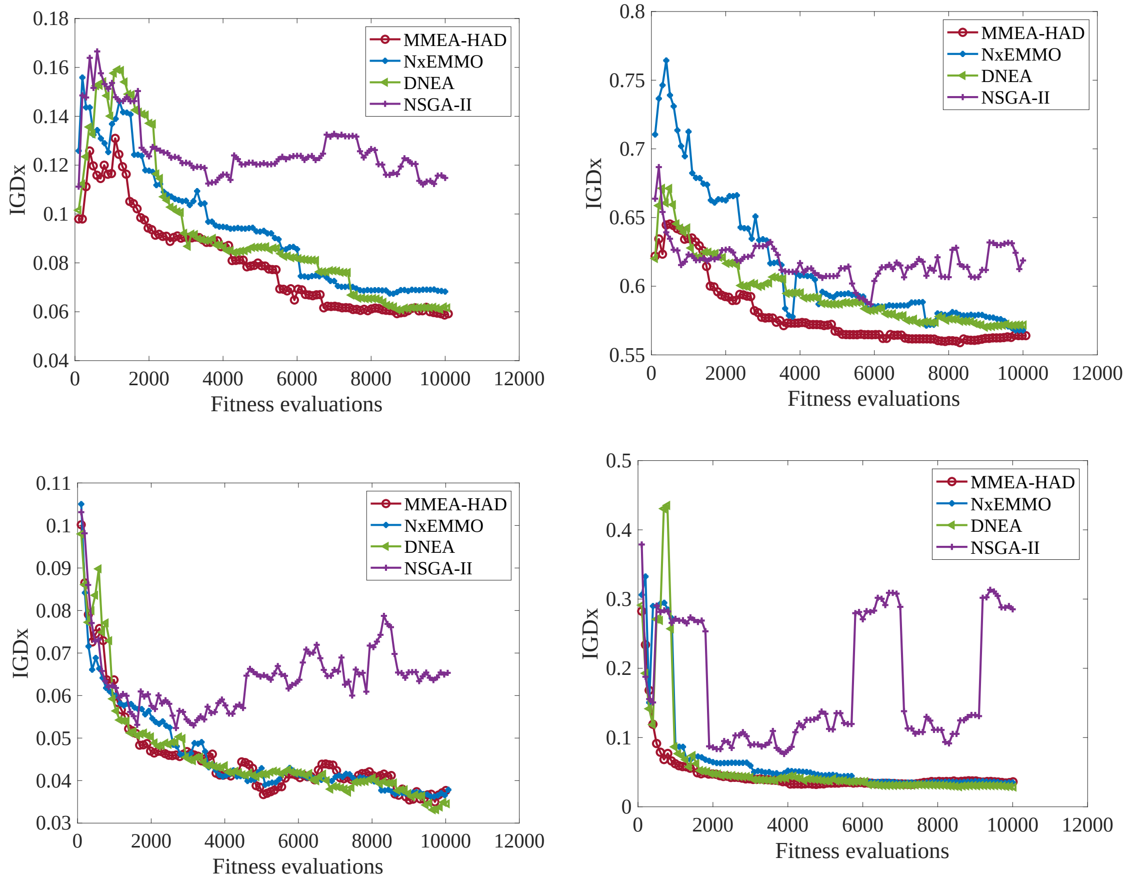


Figure 5.10: Comparison of the IGDx values for the median run over the generations for MMF1 (top), MMF5 (second from top), MMF7 (third from top), Omni (bottom) test problems [119].

To assess the scalability of the proposed algorithms, we have evaluated them across a range of test problems with varying levels of complexity, particularly those with higher decision and objective space dimensions. Based on our findings, the NxEMMO algorithm, in conjunction with the proposed selection mechanism, consistently generates more diverse solutions in the search space compared to other competitive algorithms.

Chapter 6

Conclusions and perspectives

In this chapter, we offer a concise overview of the outcomes derived from this thesis and draw conclusive remarks concerning the research objectives outlined in Chapter 1. Additionally, we outline a number of promising avenues for future research in the field of multi-modal and multi-objective optimization problems.

This thesis focuses on metaheuristic optimization of multi-modal, multi-objective problems. Chapter 2 introduced fundamental concepts such as the formal definition of multi-objective problems, evolutionary algorithms, multi-modality, and Pareto optimality. Additionally, the benchmark problem characteristics and performance indicators used for algorithm comparison in this thesis were presented. Furthermore, the chapter discusses the general experimental settings employed. Chapter 3 reviewed the state-of-the-art literature in multi-objective optimization algorithms, including both single-objective and multi-modal multi-objective optimization problems. The chapter focused on algorithms specifically designed to solve multi-modal, multi-objective optimization problems involving multiple global optimal solutions. Additionally, algorithms used to address the second category of these problems were also covered.

Chapters 4 and 5 presented new contributions to this thesis. The proposed approaches to dealing with MMOPs enhance existing algorithms by combining two types of strategies. The first approach enables deeper exploration of the search space by modifying tournament selection, reproduction, and mutation operators to identify multiple distinct solutions simultaneously. Furthermore, two categories of environmental selection procedures are introduced, tailored to MMOPs involving multiple global Pareto sets and those requiring both local and global Pareto sets. Experimental evaluations compare the proposed methods with prominent related algorithms, demonstrating their superior efficiency and effectiveness in exploring and preserving diverse solutions in the search space across various MMOPs. Chapter 5 introduces a new classification of selection operations in Pareto-based MMOEAs, distinguishing between inter-front and intra-front selection. The performance of the proposed algorithms in these categories is compared with other state-of-the-art algorithms.

6.1 Research objectives

The thesis concludes here with a summary of the research objectives and the results obtained for each objective.

Objective 1: Develop methods and approaches to address Type I MMOPs

The primary objective of this thesis was to propose strategies for solving multi-modal, multi-objective problems with multiple global Pareto sets (PSs). To achieve this, existing methodologies and search strategies were studied, modified, and applied specifically to MMOPs of type I. A key research question addressed in this context pertains to the development of methods to guide the population toward exploring diverse regions of the search space to avoid premature convergence and local optima. A significant portion of this thesis focused on addressing the challenge posed by traditional selection operators in MOEAs, which tend to remove diverse solutions located in crowded regions of the objective space to enhance the distribution of optimal solutions obtained over the approximated Pareto front. To overcome this challenge, novel selection operators were developed to better assess the similarity between solutions and ensure the survival of solutions from distinct regions of the search space that may be close to the objective space. Balancing the diversity of solutions in both the decision and objective spaces is a complex task in MMOPs because improving the distribution of solutions in the search space can negatively impact the distribution of solutions in the objective space, and vice versa. To address this trade-off, a method was developed to measure density in both the search space and objective space, allowing for a balanced consideration of the divergence of solutions in both spaces.

The performance of the proposed methods was comprehensively evaluated by comparing them with recent and well-known algorithms discussed in the chapter. Several multi-modal, multi-objective optimization problems with varying complexities and decision space dimensions were employed to assess the capabilities and practicality of the proposed algorithms. The results obtained from type I MMOPs clearly demonstrated the exceptional performance of the proposed approaches in terms of efficiency and effectiveness, surpassing state-of-the-art algorithms in terms of their ability to explore and preserve diverse solutions within the search space. These findings highlight the excellence of the proposed algorithms and their potential to advance the field of multi-modal, multi-objective optimization.

Objective 2: Develop methods and algorithms to solve Type II MMOPs.

The second objective of this thesis focused on developing approaches to address multi-modal, multi-objective optimization problems that involve the presence of both global and local optima. To achieve this objective, we proposed an algorithm specifically designed to handle such problems. In the proposed algorithm, the population was divided into multiple subpopulations using various clustering techniques, and selection methods were applied to each niche individually. The proposed algorithm integrates a clustering method in the search space to retain local Pareto-optimal solutions. By organizing non-dominated solutions into clusters, the algorithm effectively maintains the local optima while simultaneously searching for global Pareto-optimal solutions throughout the evolutionary process. The incorporation of a density measurement technique facilitates guidance for the optimizer in exploring the search space, resulting in enhanced preservation of local optima.

Objective 3: Classification of the selection mechanisms in Pareto-dominance based MMOEAs

This thesis addresses the absence of a systematic classification of algorithms based on selection operators, which play a crucial role in multi-modal, multi-objective evolutionary algorithms. We identify several limitations of crowding distance methods, particularly their inaccurate density estimations for neighboring solutions within the search space. To rectify these shortcomings, we propose a novel classification for the selection operations of Pareto-based MMOEAs based on how nearby solutions from current or previous fronts are considered when estimating the density of the solution's neighborhood area. To mitigate the deficiencies of existing crowding methods, we introduce two classifications: inter-front and intra-front selection operators, each characterized by distinct selection mechanisms. The evaluation of the proposed algorithms on diverse test problems, including those with higher decision and objective space dimensions, provided insights into their scalability. The results demonstrate that, in most cases, the proposed algorithm—which belongs to the category of inter-front operators and considers neighboring solutions from both the current and previous fronts—achieves a higher level of solution diversity in the search space compared to other competitive algorithms.

Objective 4: Evaluation of the proposed algorithms

The proposed strategies were subjected to comprehensive testing, including comparisons with various state-of-the-art algorithms from the literature as well as among themselves, within the context of this research objective. Performance evaluations involve the utilization of different test functions from the literature, varying in complexity and number of decision variables. The algorithms were assessed based on specific criteria, highlighting their strengths and weaknesses. Performance measurements such as diversity and convergence in both the search space and the objective space were employed to facilitate the comparison of the different methods. Furthermore, the impact of population size on the diversity and convergence speed of the obtained solutions toward the PS and PF was investigated.

6.2 perspectives

The research presented in this thesis represents a significant advancement in the field of multi-modal, multi-objective optimization since its introduction in 2018. While there has been a notable surge in research interest in this domain, particularly in recent years, several challenges persist in the field of multi-modal optimization, offering exciting opportunities for future investigations. Here, we outline several promising areas for further research.

- One crucial future research direction in the field is the development of performance indicators that enable the precise evaluation of algorithm performance, especially for real-world multi-modal multi-objective optimization problems where knowledge about the location of the true Pareto front is unavailable. Introducing reliable performance indicators will provide researchers with more in-depth insights into algorithm performance and enhance the evaluation process, leading to advancements in the field of multi-modal multi-objective optimization. This research direction holds great potential for improving our understanding and application of algorithms in practical scenarios.

- In addition to evaluating the algorithm's performance on large-scale objective MMOPs, another interesting research area to explore is the development of advanced selection mechanisms within the algorithm. Specifically, investigating novel selection strategies that adaptively adjust the selection pressure based on the characteristics of the problem and the current population can lead to improved exploration and exploitation capabilities. These adaptive selection mechanisms have the potential to enhance the algorithm's ability to maintain diversity, overcome premature convergence, and discover a more comprehensive set of Pareto-optimal solutions.

- Another promising direction for future work involves investigating innovative selection strategies that dynamically adapt the selection pressure according to the problem's characteristics and the current population. By incorporating such adaptive selection mechanisms, the algorithm can enhance its exploration and exploitation capabilities, resulting in improved performance. These strategies have the potential to address challenges such as maintaining solution diversity, mitigating premature convergence, and identifying a more extensive set of Pareto-optimal solutions. Exploring these research areas contributes to advancing multi-modal, multi-objective optimization algorithms, making them more effective in tackling complex real-world optimization problems.

- Another potential future direction is the development of comprehensive benchmark functions tailored specifically for evaluating the performance of multi-modal optimization algorithms. These benchmark functions play a crucial role in facilitating fair and standardized comparisons among different approaches. To ensure their effectiveness, these benchmarks should encompass a diverse range of challenging characteristics, including varying modalities, nonlinearity, and high-dimensionality. Such comprehensive benchmark functions will provide a valuable resource for researchers to assess and compare the performance of multi-modal optimization algorithms accurately.

- Another important future direction is the application of multi-modal optimization techniques to address complex real-world problems in various domains. By employing these techniques in practical contexts such as engineering design, or path finding problems, researchers can showcase the practical value and effectiveness of multi-modal optimization approaches. Investigating the applicability and performance of multi-modal optimization algorithms across diverse application domains will provide valuable insights and contribute to the advancement of real-world problem-solving capabilities. This avenue of research will bridge the gap between theoretical advancements and practical applications, ultimately leading to the development of more robust and efficient solutions for real-world challenges.

By focusing on these promising research directions, we can further advance the field of multi-modal, multi-objective optimization and address the remaining challenges, ultimately leading to more effective and efficient solutions for complex optimization problems.

Bibliography

[1] Monalisa Pal and Sanghamitra Bandyopadhyay. Decomposition in decision and objective space for multi-modal multi-objective optimization. *Swarm and Evolutionary Computation*, 62:100842, 2021. (page 1)

[2] Carlos A Coello Coello, Gary B Lamont, David A Van Veldhuizen, et al. *Evolutionary algorithms for solving multi-objective problems*, volume 5. Springer, 2007. (page 2)

[3] Xin Yao. Global optimisation by evolutionary algorithms. In *Proceedings of IEEE International Symposium on parallel algorithms architecture synthesis*, pages 282–291. IEEE, 1997. (page 2)

[4] Eckart Zitzler. *Evolutionary algorithms for multiobjective optimization: Methods and applications*, volume 63. Citeseer, 1999. (page 2)

[5] Andrzej Jaszkiewicz. On the performance of multiple-objective genetic local search on the 0/1 knapsack problem-a comparative experiment. *IEEE Transactions on Evolutionary Computation*, 6(4):402–412, 2002. (page 3)

[6] Jens Weise and Sanaz Mostaghim. A comparison of distance metrics for the multi-objective pathfinding problem. *Natural Computing*, pages 1–14, 2022. (page 3)

[7] Jens Weise and Sanaz Mostaghim. Many-objective pathfinding based on fréchet similarity metric. In *Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2021, Shenzhen, China, March 28–31, 2021, Proceedings*, pages 375–386. Springer, 2021. (page 3)

[8] Jens Weise. *Evolutionary many-objective optimisation for pathfinding problems.* PhD thesis, Dissertation, Magdeburg, Otto-von-Guericke-Universität Magdeburg, 2023, 2023. (page 3)

[9] Michele Sebag, Nicolas Tarrisson, Olivier Teytaud, Julien Lefevre, and Sylvain Baillet. A multi-objective multi-modal optimization approach for mining stable spatio-temporal patterns. In *IJCAI*, pages 859–864, 2005. (page 3)

[10] Fumiya Kudo, Tomohiro Yoshikawa, and Takeshi Furuhashi. A study on analysis of design variables in pareto solutions for conceptual design optimization problem of hybrid rocket engine. In *2011 IEEE Congress of Evolutionary Computation (CEC)*, pages 2558–2562. IEEE, 2011. (page 3)

[11] Mike Preuss, Michael G Epitropakis, Xiaodong Li, and Jonathan E Fieldsend. *Metaheuristics for Finding Multiple Solutions*. Springer, 2021. (page 3, 4, 13)

[12] Christian Grimme, Pascal Kerschke, Pelin Aspar, Heike Trautmann, Mike Preuss, André H. Deutz, Hao Wang, and Michael Emmerich. Peeking beyond peaks: Challenges and research potentials of continuous multimodal multi-objective optimization. *Computers and Operations Research*, 136(September 2020):105489, 2021. (page 3, 37)

[13] Cem C Tutum and Kalyanmoy Deb. A multimodal approach for evolutionary multi-objective optimization (memo): Proof-of-principle results. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 3–18. Springer, 2015. (page 13)

[14] Mike Preuss. *Review of "Multimodal Optimization by Means of Evolutionary Algorithms" by Mike Preuss*, volume 8. 2016. (page 13, 34)

[15] Amiram Moshaiov. The paradox of multimodal optimization: Concepts vs. species in single and multi-objective problems. *2016 IEEE Congress on Evolutionary Computation, CEC 2016*, pages 1743–1748, 2016. (page 14)

[16] Yiping Liu, Gary G. Yen, and Dunwei Gong. A Multimodal Multiobjective Evolutionary Algorithm Using Two-Archive and Recombination Strategies. *IEEE Transactions on Evolutionary Computation*, 23(4):660–674, 2019. (page 14, 39)

[17] Yiping Liu, Hisao Ishibuchi, Gary G. Yen, Yusuke Nojima, and Naoki Masuyama. Handling Imbalance Between Convergence and Diversity in the Decision Space in Evolutionary Multi-Modal Multi-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, pages 1–15, 2019. (page 14)

[18] Ryoji Tanabe and Hisao Ishibuchi. A Review of Evolutionary Multimodal Multi-objective Optimization. *IEEE Transactions on Evolutionary Computation*, 24(1):193–200, 2019. (page 15, 21, 36, 40)

[19] Christian Blum and Andrea Roli. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308, 2003. (page 17)

[20] Kalyanmoy Deb. An introduction to genetic algorithms. *Sadhana*, 24(4-5):293–315, 1999. (page 17)

[21] Robert E Dorsey and Walter J Mayer. Genetic algorithms for estimation problems with multiple optima, nondifferentiability, and other irregular features. *Journal of Business & Economic Statistics*, 13(1):53–66, 1995. (page 17)

[22] Rudolf Kruse, Christian Borgelt, Christian Braune, Sanaz Mostaghim, Matthias Steinbrecher, Frank Klawonn, and Christian Moewes. *Computational intelligence*. Springer, 2011. (page 17)

[23] Fred Glover. Tabu search and adaptive memory programming—advances, applications and challenges. In *Interfaces in computer science and operations research*, pages 1–75. Springer, 1997. (page 17)

[24] A Osycska. Multi-criterion optimization in engineering with fortran examples, 1984. (page 17)

[25] M Dorigo. Stu tzle t. ant colony optimization. *Bradford Company, Scituate*, 2004. (page 17)

[26] Andrew Lewis, Sanaz Mostaghim, and Marcus Randall. *Biologically-Inspired Optimisation Methods: Parallel Algorithms, Systems and Applications*, volume 210. Springer, 2009. (page 17)

[27] Sanaz Mostaghim, Jürgen Branke, and Hartmut Schmeck. Multi-objective particle swarm optimization on computer grids. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 869–875, 2007. (page 17)

[28] Sanaz Mostaghim and Jürgen Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In *Proceedings of the 2003 IEEE Swarm Intelligence Symposium. SIS'03 (Cat. No. 03EX706)*, pages 26–33. IEEE, 2003. (page 17)

[29] Kalyanmoy Deb. Multi-objective optimisation using evolutionary algorithms: an introduction. In *Multi-objective evolutionary optimisation for product design and manufacturing*, pages 3–34. Springer, 2011. (page 17)

[30] Eckart Zitzler and Lothar Thiele. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *IEEE transactions on Evolutionary Computation*, 3(4):257–271, 1999. (page 17)

[31] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. Spea2: Improving the strength pareto evolutionary algorithm. *TIK-report*, 103, 2001. (page 17, 18)

[32] Haiming Lu. *State-of-the-art multiobjective evolutionary algorithms: Pareto ranking, density estimation and dynamic population*. Oklahoma State University, 2002. (page 18)

[33] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, 2002. (page 18, 48, 57, 58, 59, 81, 104, 118)

[34] Mahrokh Javadi and Sanaz Mostaghim. Using neighborhood-based density measures for multimodal multi-objective optimization. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 335–345. Springer, 2021. (page 18, 96, 104, 118, 123)

[35] Salem Fawaz Adra. *Improving convergence, diversity and pertinency in multiobjective optimisation*. PhD thesis, University of Sheffield, Department of Automatic Control and Systems Engineering, 2007. (page 19)

[36] J David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the first international conference on genetic algorithms and their applications, 1985*. Lawrence Erlbaum Associates. Inc., Publishers, 1985. (page 19)

[37] David E Goldberg. Genetic algorithms in search, optimization, and machine learning. addison. *Reading*, 1989. (page 19, 34)

[38] MA Reading and Addison Wesley. Freund, je (1962). mathematical statistics. englewood cliffs, nj.: Prentice-hall. goldberg, d., & richardson, j.(1987). genetic algorithms with sharing for multimodal function optimization. in proceedings of the second international conference on genetic algorithms, 148–154, san mateo, ca. morgan kaufmann. goldberg, de (1989). genetic algorithms in search, optimization, and machine learning. (page 19, 20, 24)

[39] James E Baker et al. Reducing bias and inefficiency in the selection algorithm. In *Proceedings of the second international conference on genetic algorithms*, volume 206, pages 14–21, 1987. (page 19)

[40] Brad L Miller, David E Goldberg, et al. Genetic algorithms, tournament selection, and the effects of noise. *Complex systems*, 9(3):193–212, 1995. (page 19)

[41] GR Harik. Finding multimodal solutions using restricted tournament selection. *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 24–31, 1995. (page 19, 24, 52)

[42] KA DeJong. Analysis of the behavior of a class of genetic adaptive systems. phd thesis, dept. of computer and communication sciences. university of michigan, 1975. 1975. (page 20)

[43] Gilbert Syswerda et al. Uniform crossover in genetic algorithms. In *ICGA*, volume 3, pages 2–9, 1989. (page 20)

[44] Kalyanmoy Deb, Ram Bhushan Agrawal, et al. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995. (page 20)

[45] Kalyanmoy Deb and Hans-georg Beyer. Real-coded genetic algorithms with simulated binary crossover: Studies on multi-modal and multi-objective problems. In *Complex systems*. Citeseer, 1995. (page 20, 110)

[46] Kalyanmoy Deb, Mayank Goyal, et al. A combined genetic adaptive search (geneas) for engineering design. *Computer Science and informatics*, 26:30–45, 1996. (page 20, 56)

[47] Agoston E Eiben, James E Smith, et al. *Introduction to evolutionary computing*, volume 53. Springer, 2003. (page 21)

[48] Agoston E Eiben and Cornelis A Schippers. On evolutionary exploration and exploitation. *Fundamenta Informaticae*, 35(1-4):35–50, 1998. (page 21)

[49] Thomas Back. The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In *Proc. 2nd Conference of Parallel Problem Solving from Nature, 1992*. Elsevier Science Publishers, 1992. (page 21)

[50] Samir W Mahfoud. *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign, 1995. (page 22)

[51] Ronald A Fisher. Darwinian evolution of mutations. *The Eugenics Review*, 14(1):31, 1922. (page 22)

[52] Sewall Wright. Evolution in mendelian populations. *Genetics*, 16(2):97, 1931. (page 22)

[53] Motoo Kimura. *The neutral theory of molecular evolution*. Cambridge University Press, 1983. (page 22)

[54] Ofer M. Shir. *Niching in evolutionary algorithms*, volume 2-4. 2012. (page 23)

[55] Yiming Peng, Hisao Ishibuchi, and Ke Shang. Multi-modal Multi-objective Optimization: Problem Analysis and Case Studies. *2019 IEEE Symposium Series on Computational Intelligence, SSCI 2019*, (April):1865–1872, 2019. (page 23)

[56] Kenneth Alan De Jong. *An analysis of the behavior of a class of genetic adaptive systems.* University of Michigan, 1975. (page 24, 34)

[57] Samir W Mahfoud et al. Crowding and preselection revisited. In *PPSN*, volume 2, pages 27–36. Citeseer, 1992. (page 24)

[58] Mourad Bessaou, Alain Pétrowski, and Patrick Siarry. Island model cooperating with speciation for multimodal optimization. In *International Conference on Parallel Problem Solving from Nature*, pages 437–446. Springer, 2000. (page 24)

[59] Xiaodong Yin and Noel Germay. A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization. In *Artificial neural nets and genetic algorithms*, pages 450–457. Springer, 1993. (page 24)

[60] Jian-Ping Li, Marton E Balazs, Geoffrey T Parks, and P John Clarkson. A species conserving genetic algorithm for multimodal function optimization. *Evolutionary computation*, 10(3):207–234, 2002. (page 24, 34)

[61] Riaan Brits, Andries P Engelbrecht, and Frans van den Bergh. A niching particle swarm optimizer. In *Proceedings of the 4th Asia-Pacific conference on simulated evolution and learning*, volume 2, pages 692–696. Citeseer, 2002. (page 24)

[62] Caitong Yue, Boyang Qu, Kunjie Yu, Jing Liang, and Xiaodong Li. A novel scalable test problem suite for multimodal multiobjective optimization. *Swarm and Evolutionary Computation*, 48(April):62–71, 2019. (page 26, 27)

[63] Mahrokh Javadi and Sanaz Mostaghim. Analysis of inter and intra-front operations in multi-modal multi-objective optimization problems. *Natural Computing*, pages 1–16, 2022. (page 26, 27, 29, 36, 37, 47, 48, 103, 104, 105, 106, 107, 108, 109, 111, 113, 114, 117, 119, 120, 121, 152, 155, 156)

[64] Hisao Ishibuchi, Yu Setoguchi, Hiroyuki Masuda, and Yusuke Nojima. Performance of decomposition-based many-objective algorithms strongly

depends on pareto front shapes. *IEEE Transactions on Evolutionary Computation*, 21(2):169–190, 2016. (page 27)

[65] Mahrokh Javadi, Cristian Ramirez-Atencia, and Sanaz Mostaghim. A novel grid-based crowding distance for multimodal multi-objective optimization. In *2020 IEEE Congress on Evolutionary Computation (CEC)*, pages 1–8. IEEE, 2020. (page 28, 76, 79, 81, 87, 88, 150, 155)

[66] Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. *Genetic and Evolutionary Computation Conference, GECCO'11*, pages 649–656, 2011. (page 28, 49)

[67] Yiping Liu, Hisao Ishibuchi, Yusuke Nojima B, and Naoki Masuyama. *A double-niched evolutionary algorithm and its behavior on polygon-based problems.*, volume 1. Springer International Publishing, 2018. (page 29, 41)

[68] Yiming Peng and Hisao Ishibuchi. A Decomposition-based Large-scale Multi-modal Multi-objective Optimization Algorithm. (2020), 2020. (page 29)

[69] Ishibuchi. Modified Distance Calculation in Generational Distance and Inverted Generational Distance. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9019:110–125, 2015. (page 29, 31)

[70] Aimin Zhou, Qingfu Zhang, and Yaochu Jin. Approximating the set of pareto-optimal solutions in both the decision and objective spaces by an estimation of distribution algorithm. *IEEE Transactions on Evolutionary Computation*, 13(5):1167–1189, 2009. (page 29, 30)

[71] Caitong Yue, Boyang Qu, and Jing Liang. A Multiobjective Particle Swarm Optimizer Using Ring Topology for Solving Multimodal Multiobjective Problems. *IEEE Transactions on Evolutionary Computation*, 22(5):805–817, 2018. (page 29, 30, 36, 41, 42, 48)

[72] Lixin Tang and Xianpeng Wang. A hybrid multiobjective evolutionary algorithm for multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, 17(1):20–45, 2012. (page 30)

[73] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Modified distance calculation in generational distance and inverted generational distance. In *International conference on evolutionary multi-criterion optimization*, pages 110–125. Springer, 2015. (page 31)

[74] Peter AN Bosman and Dirk Thierens. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE transactions on evolutionary computation*, 7(2):174–188, 2003. (page 31)

[75] Andries P Engelbrecht. *Computational intelligence: an introduction*. John Wiley & Sons, 2007. (page 34)

[76] Pourya Haghbayan, Hossein Nezamabadi-Pour, and Shima Kamyab. A niche gsa method with nearest neighbor scheme for multimodal optimization. *Swarm and evolutionary computation*, 35:78–92, 2017. (page 34)

[77] G Harick. Finding multi-modal solutions in problems of bounded difficulty. Technical report, Technical report, Illinois Genetic Algorithms Laboratory, report, 1994. (page 34)

[78] Esmat Rashedi, Hossein Nezamabadi-Pour, and Saeid Saryazdi. Gsa: a gravitational search algorithm. *Information sciences*, 179(13):2232–2248, 2009. (page 34)

[79] Daniel Parrott and Xiaodong Li. Locating and tracking multiple dynamic optima by a particle swarm model using speciation. *IEEE Transactions on Evolutionary Computation*, 10(4):440–458, 2006. (page 34)

[80] A. Petrowski. Clearing procedure as a niching method for genetic algorithms. *Proceedings of the IEEE Conference on Evolutionary Computation*, pages 798–803, 1996. (page 34)

[81] EL Yu and Ponnuthurai N Suganthan. Ensemble of niching algorithms. *Information Sciences*, 180(15):2815–2833, 2010. (page 34, 35)

[82] Ole J Mengshoel and David E Goldberg. Probabilistic crowding: Deterministic crowding with probabilistic replacement. 1999. (page 34)

[83] Shima Kamyab and Mahdi Eftekhari. Using a self-adaptive neighborhood scheme with crowding replacement memory in genetic algorithm for

multimodal optimization. *Swarm and Evolutionary Computation*, 12:1–17, 2013. (page 34)

[84] Xiaodong Li. Niching without niching parameters: particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, 14(1):150–169, 2009. (page 35)

[85] Bo-Yang Qu, Ponnuthurai Nagaratnam Suganthan, and Swagatam Das. A distance-based locally informed particle swarm model for multimodal optimization. *IEEE Transactions on Evolutionary Computation*, 17(3):387–402, 2012. (page 35)

[86] Mohammad Bagher Dowlatshahi and Hossein Nezamabadi-Pour. Ggsa: a grouping gravitational search algorithm for data clustering. *Engineering Applications of Artificial Intelligence*, 36:114–121, 2014. (page 35)

[87] Mohadeseh Soleimanpour-Moghadam, Hossein Nezamabadi-Pour, and Malihe M Farsangi. A quantum inspired gravitational search algorithm for numerical function optimization. *Information Sciences*, 267:83–100, 2014. (page 35)

[88] Hossein Nezamabadi-Pour. A quantum-inspired gravitational search algorithm for binary encoded optimization problems. *Engineering Applications of Artificial Intelligence*, 40:62–75, 2015. (page 35)

[89] Anupam Yadav and Joong Hoon Kim. A niching co-swarm gravitational search algorithm for multi-modal optimization. In *Proceedings of fourth international conference on soft computing for problem solving*, pages 599–607. Springer, 2015. (page 35)

[90] Ka-Chun Wong. Evolutionary multimodal optimization: A short survey. *arXiv preprint arXiv:1508.00457*, 2015. (page 35)

[91] Swagatam Das, Sayan Maity, Bo-Yang Qu, and Ponnuthurai Nagaratnam Suganthan. Real-parameter evolutionary multimodal optimization—a survey of the state-of-the-art. *Swarm and Evolutionary Computation*, 1(2):71–88, 2011. (page 35)

[92] Chenxu Hu and Hisao Ishibuchi. Incorporation of a decision space diversity maintenance mechanism into moea/d for multi-modal multi-objective optimization. *GECCO 2018 Companion - Proceedings of the*

2018 Genetic and Evolutionary Computation Conference Companion, pages 1898–1901, 2018. (page 36)

[93] Qingfu Zhang and Hui Li. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on Evolutionary Computation*, 11(6):712–731, 2007. (page 36)

[94] Ryoji Tanabe and Hisao Ishibuchi. A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11101 LNCS:249–261, 2018. (page 36, 37)

[95] Oliver Kramer and Holger Danielsiek. Dbscan-based multi-objective niching to approximate equivalent pareto-subsets. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 503–510, 2010. (page 36)

[96] Oliver Kramer and Patrick Koch. Rake selection: A novel evolutionary multi-objective optimization algorithm. In *Annual Conference on Artificial Intelligence*, pages 177–184. Springer, 2009. (page 36)

[97] Michèle Sebag, Nicolas Tarrisson, Olivier Teytaud, Julien Lefevre, and Sylvain Baillet. A multi-objective multi-modal optimization approach for mining stable spatio-temporal patterns. *IJCAI International Joint Conference on Artificial Intelligence*, 2(1):859–864, 2005. (page 36)

[98] Tomoyuki Hiroyasu, Seiichi Nakayama, and Mitsunori Miki. Comparison study of SPEA2+, SPEA2, and NSGA-II in diesel engine emissions and fuel economy problem. *2005 IEEE Congress on Evolutionary Computation, IEEE CEC 2005. Proceedings*, 1:236–242, 2005. (page 36, 42)

[99] Mifa Kim, Tomoyuki Hiroyasu, Mitsunori Miki, and Shinya Watanabe. SPEA2+: Improving the performance of the strength pareto evolutionary algorithm 2. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 3242:742–751, 2004. (page 36, 42)

[100] Kalyanmoy Deb and Santosh Tiwari. Omni-optimizer: A generic evolutionary algorithm for single and multi-objective optimization. *European Journal of Operational Research*, 185(3):1062–1087, 2008. (page 36, 40, 49, 59, 70, 105, 114)

[101] J J Liang and C T Yue. Multimodal Multi-objective Optimization : A Preliminary Study. pages 2454–2461, 2016. (page 36, 41, 49, 105, 111, 114)

[102] Mahrokh Javadi, Heiner Zille, and Sanaz Mostaghim. Modified crowding distance and mutation for multimodal multi-objective optimization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pages 211–212, 2019. (page 36, 55, 58, 62, 63, 64, 65, 66, 75, 84, 151, 154)

[103] Mahrokh Javadi and Sanaz Mostaghim. A Neighborhood-based Density Measure for Multimodal Multi-Objective Optimization. In *Ishibuchi H. et al. (eds) Evolutionary Multi-Criterion Optimization. EMO 2021*, pages 335–345. Lecture Notes in Computer Science, vol 12654. Springer, Cham., 2021. (page 36, 110, 112)

[104] Yali Wang, Michael Emmerich, André Deutz, and Thomas Bäck. Diversity-indicator based multi-objective evolutionary algorithm: Di-moea. In *International Conference on Evolutionary Multi-Criterion Optimization*, pages 346–358. Springer, 2019. (page 37)

[105] Qingfu Zhang and Hui Li. Moea/d: A multiobjective evolutionary algorithm based on decomposition. *IEEE Transactions on evolutionary computation*, 11(6):712–731, 2007. (page 37)

[106] Ryoji Tanabe and Hisao Ishibuchi. A decomposition-based evolutionary algorithm for multi-modal multi-objective optimization. In *Parallel Problem Solving from Nature–PPSN XV: 15th International Conference, Coimbra, Portugal, September 8–12, 2018, Proceedings, Part I 15*, pages 249–261. Springer, 2018. (page 38)

[107] Ryoji Tanabe and Hisao Ishibuchi. A Framework to Handle Multi-modal Multi-objective Optimization in Decomposition-based Evolutionary Al-

gorithms. *IEEE Transactions on Evolutionary Computation*, pages 1–1, 2019. (page 39)

[108] Mario Inostroza-ponta and Manuel Villalobos-cid. A multimodal multi-objective optimisation approach to deal with the phylogenetic inference problem. 2020. (page 40)

[109] Xiaodong Li. Niching without niching parameters: Particle swarm optimization using a ring topology. *IEEE Transactions on Evolutionary Computation*, 14(1):150–169, 2010. (page 42)

[110] Jing Liang, Qianqian Guo, Caitong Yue, Boyang Qu, and Kunjie Yu. A self-organizing multi-objective particle swarm optimization algorithm for multimodal multi-objective problems. In *Advances in Swarm Intelligence: 9th International Conference, ICSI 2018, Shanghai, China, June 17-22, 2018, Proceedings, Part I 9*, pages 550–560. Springer, 2018. (page 42)

[111] Ofer M Shir, Mike Preuss, Boris Naujoks, and Michael Emmerich. Enhancing decision space diversity in evolutionary multiobjective algorithms. In *Evolutionary Multi-Criterion Optimization: 5th International Conference, EMO 2009, Nantes, France, April 7-10, 2009. Proceedings 5*, pages 95–109. Springer, 2009. (page 42)

[112] Eckart Zitzler, Marco Laumanns, and Lothar Thiele. SPEA2: Improving the Strength Pareto Evolutionary Algorithm. *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pages 95–100, 2001. (page 42, 110)

[113] Dũng H Phan and Junichi Suzuki. R2-ibea: R2 indicator based evolutionary algorithm for multiobjective optimization. In *2013 IEEE congress on evolutionary computation*, pages 1836–1845. IEEE, 2013. (page 44)

[114] C. T. Yue, J. J. Liang, P. N. Suganthan, B. Y. Qu, K. J. Yu, and S. Liu. MMOGA for Solving Multimodal Multiobjective Optimization Problems with Local Pareto Sets. pages 1–8, 2020. (page 44, 45)

[115] Qiuzhen Lin, Wu Lin, Zexuan Zhu, Maoguo Gong, Jianqiang Li, and Carlos Coello. Multimodal Multi-objective Evolutionary Optimization

with Dual Clustering in Decision and Objective Spaces. *IEEE Transactions on Evolutionary Computation*, (c):1–1, 2020. (page 45, 91, 97, 98)

[116] Yiping Liu, Hisao Ishibuchi, Yusuke Nojima, Naoki Masuyama, and Yuyan Han. Searching for Local Pareto Optimal Solutions: A Case Study on Polygon-Based Problems. *2019 IEEE Congress on Evolutionary Computation, CEC 2019 - Proceedings*, (2017):896–903, 2019. (page 46)

[117] Wenhua Li, Xingyi Yao, Tao Zhang, Rui Wang, and Ling Wang. Hierarchy ranking method for multimodal multi-objective optimization with local pareto fronts. *IEEE Transactions on Evolutionary Computation*, 2022. (page 46)

[118] Henry B Mann and Donald R Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, pages 50–60, 1947. (page 47)

[119] Mahrokh Javadi and Sanaz Mostaghim. A Multi-objective Multimodal Evolutionary Algorithm Using a Novel Tournament and Environmental Selections. 2021. (page 47, 52, 53, 54, 104, 115, 116, 118, 122, 123, 124, 125, 126, 127, 152, 153, 156)

[120] J J Liang, B Y Qu, D W Gong, and C T Yue. Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization Problem Definitions and Evaluation Criteria for the CEC 2019 Special Session on Multimodal Multiobjective Optimization Introduction to t. *Cec2019*, (March), 2019. (page 49)

[121] Günter Rudolph, Boris Naujoks, and Mike Preuss. Capabilities of EMOA to detect and preserve equivalent pareto subsets. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4403 LNCS:36–50, 2007. (page 49)

[122] Hisao Ishibuchi, Yasuhiro Hitotsuyanagi, Noritaka Tsukamoto, and Yusuke Nojima. Many-objective test problems to visually examine the behavior of multiobjective evolution in a decision space. In *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference,*

Kraków, Poland, September 11-15, 2010, Proceedings, Part II 11, pages 91–100. Springer, 2010. (page 49)

[123] Mahrokh Javadi and Kourosh Eshghi. A new combinatorial clustering protocol for decreasing energy consumption in wireless sensor networks based on Ant Colony Optimization (ACO). 2011. (page 52)

[124] Mahrokh Javadi, Cristian Ramirez-Atencia, and Sanaz Mostaghim. Combining Manhattan and Crowding distances in Decision Space for Multimodal Multi-objective Optimization problems. *Springer ECCOMAS book series on Computational Methods in Applied Sciences*, (Mdcd), 2020. (page 52, 69, 70, 71, 72, 73, 74, 76, 77, 155)

[125] Mahrokh Javadi, Heiner Zille, and Sanaz Mostaghim. The Efects of Crowding Distance and Mutation in Multimodal and Multi-objective Optimization Problems. (1):1–15. (page 52)

[126] Mahrokh Javadi, Heiner Zille, and Sanaz Mostaghim. The Effects of Crowding Distance and Mutationin Multimodal and Multi-objective Optimization Problems. *Springer ECCOMAS book series on Computational Methods in Applied Sciences*, 2020. (page 55, 57, 58, 60, 61, 62, 63, 67, 68, 69, 74, 105, 150, 154, 155)

[127] Bo-Yang Qu, Ponnuthurai Nagaratnam Suganthan, and Jane-Jing Liang. Differential evolution with neighborhood mutation for multimodal optimization. *IEEE transactions on evolutionary computation*, 16(5):601–614, 2012. (page 55, 56, 62)

[128] Mohammad Hamdan. The distribution index in polynomial mutation for evolutionary multiobjective optimisation algorithms: An experimental study. In *International Conference on Electronics Computer Technology*, Kanyakumari,India, 2012. IEEE. (page 56)

[129] Antony W Iorio and Xiaodong Li. A cooperative coevolutionary multiobjective algorithm using non-dominated sorting. In *Genetic and Evolutionary Computation Conference*, pages 537–548. Springer, 2004. (page 56, 57, 58)

[130] Yi Hu, Jie Wang, Jing Liang, Kunjie Yu, Hui Song, Qianqian Guo, Caitong Yue, and Yanli Wang. A self-organizing multimodal multi-

objective pigeon-inspired optimization algorithm. 62(July):1–17, 2019. (page 68)

[131] Shengxiang Yang, Miqing Li, Xiaohui Liu, and Jinhua Zheng. A grid-based evolutionary algorithm for many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 17(5):721–736, 2013. (page 77)

[132] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231, 1996. (page 89, 92)

[133] Yiming Peng and Hisao Ishibuchi. A Diversity-enhanced Subset Selection Framework for Multi-modal Multi-objective Optimization. *IEEE Transactions on Evolutionary Computation*, PP(XX):1–1, 2021. (page 92)

[134] Yusliza Yusoff, Mohd Salihin Ngadiman, and Azlan Mohd Zain. Overview of nsga-ii for optimizing machining process parameters. *Procedia Engineering*, 15:3978–3983, 2011. (page 104)

List of Tables

List of Figures

Appendix A

Publications of the candidate

1. Mahrokh Javadi, Heiner Zille, and Sanaz Mostaghim. Modified crowding distance andmutation for multimodal multi-objective optimization. In Proceedings of the Genetic and Evolutionary Computation (GECCO) Conference Companion, pages 211–212, ACM, Prague, Czech Republic, July 2019.

2. Marde Helbig, Heiner Zille, Mahrokh Javadi and Sanaz Mostaghim, Performance of Dynamic Algorithms on the Dynamic Distance Minimization Problem, ACM Genetic and Evolutionary Computation Conference (GECCO) Companion, Pages 205-206, ACM, Prague, Czech Republic, July 2019.

3. Mahrokh Javadi, Heiner Zille, and Sanaz Mostaghim. The effects of crowding distance and mutation in multimodal and multi-objective optimization problems. In Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, pages 115–130. Springer, 2021.

4. Mahrokh Javadi, Cristian Ramirez-Atencia, and Sanaz Mostaghim. Combining Manhattan and crowding distances in decision space for multimodal multi-objective optimization problems. In Advances in Evolutionary and Deterministic Methods for Design, Optimization and Control in Engineering and Sciences, pages 131–145. Springer, 2021.

5. Mahrokh Javadi, Cristian Ramirez-Atencia, and Sanaz Mostaghim. A novel grid-based crowding distance for multimodal multi-objective opti-

mization. In2020 IEEE Congress on Evolutionary Computation (CEC), pages 1–8. IEEE, 2020.

6. Mahrokh Javadi and Sanaz Mostaghim. Using neighborhood-based density measures for multimodal multi-objective optimization. In Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2021, Shenzhen, China, March 28–31, 2021,Proceedings 11, pages 335–345. Springer International Publishing, 2021.

7. Mahrokh Javadi and Sanaz Mostaghim. A multi-objective multimodal evolutionary algorithm using a novel tournament and environmental selections. In 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2021.

8. Mahrokh Javadi and Sanaz Mostaghim. Analysis of inter and intra-front operations multimodal multi-objective optimization problems. In special issue of Natural Computing on Deterministic and Stochastic Methods for Multi-objective Optimization, 2022.

9. Mahrokh Javadi and Sanaz Mostaghim. The Impact of Selection Strategies for Preserving Diversity in Multimodal Multi-objective Optimization. Submitted, In Evolutionary Multi-Criterion Optimization: 11th International Conference, EMO 2023, Nederland, Leiden, March 20–24.