Online Knowledge-based Evolutionary Multi-objective Optimisation

Author:
Zhang, Bin

Publication Date:
2015

DOI:
https://doi.org/10.26190/unsworks/18575

Online Knowledge-based Evolutionary Multi-objective Optimisation

Bin Zhang

M.Sc. (Computer Science) Ocean University of China, China

B.Sc. (Mathematics) Ocean University of China, China

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy at the
School of Engineering and Information Technology
University of New South Wales, Canberra
Australian Defence Force Academy

Abstract

Knowledge-based optimization is a recent direction in evolutionary optimization research which aims at understanding the optimization process, discovering relationships between decision variables and performance parameters, and using discovered knowledge to improve the optimization process, using machine learning techniques.

This thesis makes two major contributions in the existing body of knowledge in the area of evolutionary multi-objective optimization. First, in addition to the well-researched objective space, it highlights the need for focusing on decision space performance analysis for benchmarking multi-objective evolutionary algorithms in general, and more specifically the knowledge-based class of these algorithms. In this respect, the thesis proposes a new method to generate multi-objective optimization test problems with clustered Pareto sets in hyper-rectangular defined areas of the decision space, which mimics knowledge representation in propositional logic. Further, a new metric is introduced for performance measurement in terms of their coverage of the optimal decision sub-space. The proposed test problems and metrics are used to benchmark multi-objective evolutionary algorithms in both objective and decision spaces.

Second, this thesis introduces a novel evolutionary optimization framework that incorporates a knowledge-based representation to search for Pareto optimal patterns in decision space replacing the conventional point-based representation. Compared to the extant approaches, which process the post-optimization Pareto sets for knowledge discovery using statistical or machine learning methods, the framework facilitates online discovery of knowledge during the optimization process in the form of interpretable rules. The core contributing idea is that the multi-objective evolutionary process is applied on a population of bounding hypervolumes, or rules, instead of evolving individual point-based solutions. The framework is generic in the sense that existing algorithms can be adapted to evaluate the quality of rules based on sampled solutions from the bounded space. Two algorithmic instantiations of the framework are presented in this thesis for both the multi and many objective optimizations respectively. The results and analysis of the experimentation with standard and proposed test benchmarks demonstrate the capabilities of the proposed optimization algorithm in comparison to the state-of-the-art multi-objective evolutionary algorithms.

keywords

Evolutionary Multi-objective Optimization, Evolutionary Many-objective Optimization, Knowledge-based Multi-objective Evolutionary Optimization, Performance Metrics, Decision Space, Objective Space, Decision Space Analysis.

Acknowledgement

I would have never been able to finish my dissertation without the continuous support from my supervisors, the University of New South Wales, my family and friends.

Foremost, I would like to express my sincere gratitude to Dr. Kamran Shafi and Prof. Hussein Abbass for the excellent supervision over my whole PhD candidature. The research experience becomes unforgettable and enjoyable because of their patience and encouragement at times when the sailing was not smooth, and the expert advice for overcoming research challenges. I am deeply grateful to the consideration and guidance from my supervisors.

Second, I wish to extend my gratitude to the University of New South Wales, Canberra for the services delivered. My gratitude goes to Mr Craig Edwards for support during my stay on campus, and Mrs Elvira Berra for managing my scholarship. My acknowledgement also goes to the Ocean University of China, the China Scholarship Council and the Education Office, Embassy of the People's Republic of China in Australia for the sponsorship, which made all of this possible.

Next, I would like to thank my friends for the discussions that contributed to my research. A thank you to Dr. Jiangjun Tang, Dr. Miao Xie, Dr. Bing Wang, Dr. Kun Wang, Mr. Murad Hossain, and many others for the company and support in my life during the PhD degree.

Personally, I sincerely thank my parents, my wife and daughter for the love and courage in all stages of my life.

Bin Zhang

Canberra 2015

Certificate of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by colleagues, with whom I have worked at UNSW or elsewhere, during my candidature, is fully acknowledged.

I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

Bin Zhang

Contents

List of Figures

List of Tables

List of Acronyms

EMO	Evolutionary Multi-objective Optimization
GA	Genetic Algorithm
GD	Generational Distance
HPS	Hyperrectangular Pareto Sets
IGD	Inverse Generational Distance
KB-EMO	Knowledge-Based Evolutionary Multi-objective Optimization
KB-MOEA	Knowledge-Based Multi-objective Optimization Evolutionary Algorithm
MOP	Multi-objective Optimization
MOEA	Multi-objective Optimization Evolutionary Algorithm
PSV	Pareto Set Volume
$RB - MOEA_{NS}$	Rule Based Multi-objective Optimization Evolutionary Algorithm using Non-Dominated sorting
$RB - MOEA_{REF}$	Rule-Based Many-objective Optimization Evolutionary Algorithm using Reference Points

List of Publications

Conference Proceedings

1. Bin Zhang, Kamran Shafi, and Hussein Abbass. Online Knowledge-based Evolutionary Multi-Objective Optimization, In the proceedings of the 2014 IEEE Congress on Evolutionary Computation (IEEE CEC 2014), pp. 2222-2229, 6-11 July 2014, Beijing, China.

2. Bin Zhang, Kamran Shafi, and Hussein Abbass. Robo-Teacher: A Computational Simulation Based Educational System to Improve Cyber Security, In the proceedings of the 1st International Conference on Robot Intelligence Technology and Applications (RiTA 2012), Robot Intelligence Technology and Applications 2012, pp. 179-186, 16-18 December 2012, Gwangju, Korea.

3. Bin Zhang, Kamran Shafi, and Hussein Abbass. Density Based Multi-Objective Optimization for Smart Distribution Grid Design, In the proceedings of the Ninth International Conference on Simulated Evolution And Learning (SEAL 2012), LNCS 7673, pp. 73-82, 16-19 December 2012, Ha Noi, Vietnam.

Journal Articles

1. Bin Zhang, Kamran Shafi, and Hussein Abbass. On Benchmark Problems and Metrics for Decision Space performance Analysis in Multi-objective Optimization. Under Review.

Chapter 1

Introduction

1.1 Context

Optimization refers to the searching process of one or more solutions with best performance from a given set according to some objectives (criteria) [4]. It usually requires maximizing or minimizing certain objective function(s) under a set of constraints and has been a significant approach for problem solving. Optimization is ubiquitous with various challenging applications and critical for decision making in practice, i.e. the determination of the structure and materials for the minimization of manufacturing cost of a car model.

Usually, there are many factors that make a decision problem and hence the optimization of the problem challenging. These include the number of decision variables which define the size of the search space, the number and complexity of the objective functions, types of constraints that limit the feasibility of the solutions, noise in objective functions and the computational cost to estimate solution quality and so on.

Multi-objective optimization problems (MOPs) are a well-known class of optimization problems which require dealing with multiple objectives simultaneously to achieve the best outcomes. Usually, the objectives in MOPs are conflicting with each other and no single solution exists to optimize all objectives unilaterally. On the contrary, there is a number of, or even infinite, solutions in the search space trading off these objectives. Such problems are prevalent and can be found in a diverse range of application domains including science, engineering, economics, business, finance and other fields. The resolution of an MOP requires the identification of the most efficient trade-off frontier of solutions, where none of the objective functions can be improved in value without degrading some of the others. However, these problems are complex and difficult to solve and many of them fall under the NP-hard category because of their properties and structure [135].

Researchers study MOPs from different viewpoints and, thus, there exists d-ifferent methodologies and goals when setting and solving them. If the guarantee of optimality is most important, there exist exact algorithms for linear and small size problems, which can return optimal solutions, especially for bi-objective optimization problems, such as branch and bound algorithms [148][127], A* algorithms [131][99], dynamic programming algorithms [125], to name a few. If the goal is to approximate or compute all or a representative set of optimal solutions when analysing and solving an MOP, evolutionary multi-objective optimization methods are more popular.

Multi-objective Optimization Evolutionary Algorithms (MOEAs), as a representative of population-based metaheuristics, have emerged as an alternative to solve and analyse this class of problems. They have gained significant attention over the last two decades[1] and have been successfully applied to a wide variety of MOPs. These algorithms, generally, tackle optimization problems as search problems by representing the solution space in an appropriate form and by iteratively searching for a set of optimal solutions in this space by sampling a set (or population) of solutions, evaluating the quality of sampled solutions based on the given objectives and determining the direction of search using some heuristics based on the sampled solutions and their quality.

The Evolutionary Multi-objective Optimization (EMO) enjoys a number of ad-

[1]while references to evolutionary multi-objective optimization exist from much earlier [146], MOEAs were popularized during early to mid-90s by several researchers including the infamous NSGA [128].

vantages [57]. First, they have shown applicability and adaptability on a variety of optimization problems and they are able to deal with large problems efficiently, despite being simple and flexible for implementation. Given the difficulties and complexities, evolutionary multi-objective algorithms are the only viable techniques when dealing with MOPs in many cases. They are easy to be implemented and the conceptual simplicity enable them to be applicable broadly, especially for problems that have no known approaches. They outperform the classic methods enumerated above when complex conditions are imposed, such as nonlinear constraints, noisy observations or dynamic objective functions, that do not conform well to the prerequisites of classic optimization techniques. Second, when decision making is emphasized, the objective of solving an MOP is to support a decision maker in finding a potential solution according to his/her subjective preferences for implementation in practice while satisfying some performance metrics and other considerations. The population-based evolutionary computation methods are able to provide simultaneous information over the trade-off frontier for efficient and flexible decision making.

1.2 Motivation

Although as famous general-purpose optimizers, MOEAs are successfully applied to MOPs. They are considered as black box techniques, which means that it is hard to tell how the optimal solutions are retrieved [4]. Presenting only a set of approximate, discrete and static optimal solutions is often insufficient for decision makers in a multi-objective context. The decision makers often deal with a dilemma where no (or only a few) solutions are viable for implementation even though the so-called pseudo optimized results have been returned. In such situations, what interests the decision makers more is not only an optimized set of solutions but also an understanding of the problem and an identification of patterns in the design space that lead to better objective performance. This becomes even critical when there is a need to generalize optimization results for problems with similar design structures, or when the problem definition changes dynamically. Knowledge extraction or discovery from a multi-objective optimization process, thus, has important implications including better understanding of the optimization process as well as of the relationship between decision variables.

There is great effort in the literature to deal with the extraction of knowledge from MOPs to support decision making. They try to identify important and hidden information or properties either from the problem definition or common to the optimal or high-performing solutions.

One of the extant approaches relies on pre-optimization formulation and monotonicity analysis of the objective functions [114]. While these techniques can provide insights into the relationship between decision variables and optimal solutions, their use is limited due to the monotonic assumptions required for constraints and objective functions. On the other hand, recent research [12] has focused on devising techniques to automatically extract knowledge from post-optimization processing of the Pareto optimal solutions. The objective of these approaches is to determine important design rules by approximating the final Pareto set with respect to the decision variables using statistical and machine learning techniques.

There are two problems with the post-optimization approaches: First the knowledge discovery process does not start until the optimization process is completed and second no information is available about how the search has progressed during the optimization process. The first problem implicates higher computational time, a longer wait to obtain the extracted knowledge and hence waste of resources; while the second problem relates to a poor understanding of the optimization process – a key motivation for carrying out the knowledge discovery process. The second problem is also restrictive for dynamic optimization problems where the search or fitness landscapes may change over time and there is never a final Pareto set which might be post-processed for knowledge extraction and discovery. The parsing of Pareto set and application of regression techniques to approximate the Pareto front can also become a bottleneck for high-dimensional MOP as most simple regression techniques do not scale to high dimensional data spaces. Finally, the post-optimization extracted knowledge is regarded irrelevant for improving the optimization problem at hand.

Based on the discussion above, knowledge-based evolutionary multi-objective optimization tries to use a knowledge-based representation to search for patterns of Pareto optimal design variables. It integrates knowledge extraction and solution optimization into a single process. At the end of the evolution, it does not only return a set of competitive solutions, but also a set of knowledge capturing interesting patterns in the decision space.

1.3 Research Hypothesis and Questions

The research problem in this thesis is to deal with online knowledge-based evolutionary multi-objective optimization. Overall, the research questions can be summarized as following:

- How to design knowledge based evolution? We have to choose appropriate knowledge representation, figure out the relationship between knowledge and solutions and coordinate knowledge optimization as well as the evolution of solutions.
- How to integrate the current techniques into the knowledge based evolution? Developments in multi-objective optimization and many-objective optimization literature can assist the design of knowledge-based algorithms regarding common issues, such as solution fitness assignment, objective space diversity and so on.
- How to benchmark the performance in the decision space? The decision space is seldom analysed compared to objective space. Decision space performance analysis requires new test problems and performance metrics to reflect the distribution or diversity of the Pareto set for testing evolutionary algorithms.

1.4 Thesis Contributions

The thesis contributes to the current multi-objective optimization field as following:

- A Knowledge-Based Evolutionary Multi-objective Optimization Framework (KB-EMO Framework) is proposed. The framework applies a multi-objective evolutionary process on a population of bounding hypervolumes, or rules, instead of evolving individual point-based solutions. The evolutionary process coordinates both rule optimization and solution optimization to extract knowledge online, which is patterns in the design space that lead to Pareto optimal solutions in the objective space.

- Knowledge-Based Multi-objective Optimization Evolutionary Algorithms (KB-MOEA) are proposed for multi and many-objective optimization problems, respectively. As instantiations of the KB-EMO framework, an algorithm termed as $RB - MOEA_{NS}$ is implemented using rule-based knowledge and non-dominated sorting based rule-quality evaluation mechanism for multi-objective optimization. The KB-EMO framework is further extended to many-objective optimization with a resulting algorithm termed as $RB - MOEA_{REF}$ using a hybrid approach. The optimization capabilities has been validated with experimentation using standard and proposed test benchmarks.

- A set of test problems, which we refer to as Hyperrectangular Pareto Sets (HPS), are proposed. HPS problems define clustered Pareto sets in hyper-rectangular areas in the decision space. HPS test suite can be used to benchmark the performance of multi-objective evolutionary algorithms in the decision space.

- A new metric, which we refer to as Pareto Set Volume (PSV), is proposed. It focuses on measuring the MOEA decision space performance in terms of the percentage of covered volume by the solution set with respect to the total volume occupied by the Pareto sets. The PSV metric, together with HPS test suites, provide an evaluation framework to investigate performance in the decision space, compensating the current objective-space centric performance assessment.

1.4.1 KB-EMO Framework

The points listed above will be explained with details in this and following subsections.

The KB-EMO framework facilitates the online identification of optimal patterns in the decision space in the form of interpretable rules during the optimization process. The knowledge-based evolutionary multi-objective optimization means that a set of rules are available during the evolutionary process capturing the current state of the optimization problem. The rules, using a hypervolume representation, provide a powerful and intuitive way of capturing knowledge. This is especially useful for robust optimization and dynamic optimization problems where decision makers might need a set of solutions and hence the knowledge about them at any point in time. However, it is worth mentioning that the framework is representation independent in that different rule representations can be used, and is not only limited to hyperrectangles which allow expressing rules in a simple if-then form.

The rule-based representation of the optimal design space also provides decision makers with a greater flexibility in exercising their preferences. Structured information of Pareto optima approximation provide insights over the problem and avoid selection pressures from only a set of approximate, discrete and static optimal solutions, which the classic algorithms present. For decision makers, especially in the case when the number of solutions is large and the number of dimensions is high, choosing one solution for implementation is difficult and extra analysis and corresponding tools are required. On the contrary, the knowledge-based evolution extracts rules alongside of the optimization process to support an understanding of the decision space and areas of good solutions with a greater depth.

The KB-EMO framework is generic in the sense that development for existing MOEAs can be adapted to support rule-based evolution. Instantiation for multi-objective optimization with hyper-rectangular rule representation and non-dominated sorting based rule evaluation is implemented. The framework is also able to deal with many-objective optimization problems with a hybrid design that combine the knowledge-based method with the state-of-the-art algorithms, such as NSGA3.

It is important to note that our framework is different from the conventional evolutionary rule learning systems [56] that use MOEA to evolve classification rules [129][155][123]. The main purpose of the above class of systems is to learn rules for data classification during a training phase where a system receives input data with class labels. The evolutionary algorithms are used to generate and refine these rules during training which are then used to classify future cases. There is no input data with correct labels in our context.

1.4.2 Performance Evaluation Benchmark and Metric

In the development and validation of MOEAs, the evaluation methodology has played an important role, involving test problem and performance metric design. Compared to the current benchmarks and metrics emphasizing objective space analysis, this thesis complements the research in multi-objective test problems with performance in the decision space in mind.

Specifically, we are interested in designing test problems where Pareto sets are sparsely clustered in the decision space to simulate optimal design *patterns*, while still maintaining the conflicting relationship in the objective space over multiple objectives. In comparison to the research in many-objective test problem generation, we restrict ourselves to only bi-objective problems but instead focus on scaling the decision space, and hence the Pareto optimal design patterns, in multiple dimensions. In addition to providing a better visualization ability, such a formulation allows to investigate the impact of increasing dimensions in the decision space in isolation of the increasing dimensionality in the objective space. We believe that such test problems could improve the development efforts for knowledge-based, specifically rule-based, multi-objective optimization algorithms.

The thesis also complements related research by proposing a new metric to measure MOEA performance in the decision space, especially in those problems where the Pareto sets are distributed over the search space in patterns. The proposed metric aims at measuring the performance based on the coverage of Pareto sets in the decision space. The metric is generic in the sense that it can be adopted for both existing MOEAs and rule-based MOEAs that work on a solution representation at a higher level of abstraction.

The proposed test problem as well as classic benchmarks can be used to evaluate the KB-MOEAs and current representative algorithms. Existing metrics and our proposed metric are used to compare their performance in both the design and objective space for a comprehensive investigation. Hence, when organizing the structure of the thesis, we first elaborate on the design of the evaluation tools first, and followed by the design of online knowledge-based evolutionary multi-objective optimization.

1.5 Thesis Outlines

The rest of the thesis is organized as following.

In Chapter 2, we provide a summary of background concepts and related work to build up the context of this research. First, the literature on MOPs, MOEAs, many-objective optimization, benchmarks and metrics is reviewed. It covers brief introduction of classic and state-of-the-art research in each area to provide the basic materials of knowledge-based multi-objective optimization. Second, the related work that highlights decision space performance analysis of multi-objective optimization, and knowledge discovery from multi-objective optimization is covered. The related research is categorized and analysed to identify the research gaps that this thesis will build on.

In Chapter 3, based on current development over the performance evaluation benchmarks and metrics, this thesis makes two contributions by providing a set of test problems and a metric highlighting performance analysis in the decision space. First, the chapter proposes a new method to generate multi-objective optimization test problems with clustered Pareto sets in hyper-rectangular defined areas of the decision space. Second, a new metric is introduced to measure the performance of algorithms in terms of their coverage of the optimal decision sub-space. The proposed test problems and metric are used to benchmark multi-objective evolutionary algorithms in both objective and decision spaces.

In Chapter 4, the online Knowledge-Based Evolutionary Multi-objective Optimization (KB-EMO) is set up. First, it presents the KB-EMO framework and then instantiates the framework using hyperrectangular representation and non-dominated sorting based rule evaluation. Comprehensive experimentation and analysis are performed based on the aforementioned test benchmarks. They are presented to demonstrate the working and convergence properties of Knowledge-Based Multi-Objective Evolutionary Algorithms (KB-MOEAs).

In Chapter 5, the KB-MOEA is extended to a many-objective environment. By combining the latest development of many-objective evolutionary algorithms, such as NSGA3, a hybrid approach is designed which combines both the point-based solution evolution and hypervolume-based rule evolution in a single run. The performance of the hybrid approach is evaluated with a group of scalable test problems with up to 15 objectives.

Finally, the thesis concludes in Chapter 6. First, the contributions of this thesis are summarized. Then the limitations are explained and potential future research directions are presented.

Chapter 2

Background and Related Work

This chapter builds up the context of the thesis and provides a brief background of the concepts related to the work presented in this thesis. It also covers a review of the existing work in the literature related to main contributions of this thesis with an aim to position the presented work appropriately in the literature. Broadly, the research work in this thesis falls under the area of EMO with a specific focus on developing the KB-MOEAs for multi and many-objective optimization and their performance evaluation. Figure 2.1 presents an overview of the areas related to this thesis and highlights the connection between the main contributions of the thesis and related concepts.

The chapter is structured as following: Section 2.1 covers background on MOPs, their formulation, the concept of Pareto optimality and considerations for solving MOPs. Sections 2.2 and 2.3 cover background on EMO and provide descriptions of some famous multi and many objective optimization evolutionary algorithms. Sections 2.4 and 2.5 cover background on performance evaluation of MOEAs and provide descriptions of the widely accepted benchmarks and performance metrics respectively. Section 2.6 reviews the existing methods for knowledge-based EMO. Finally, the chapter concludes in Section 2.7.

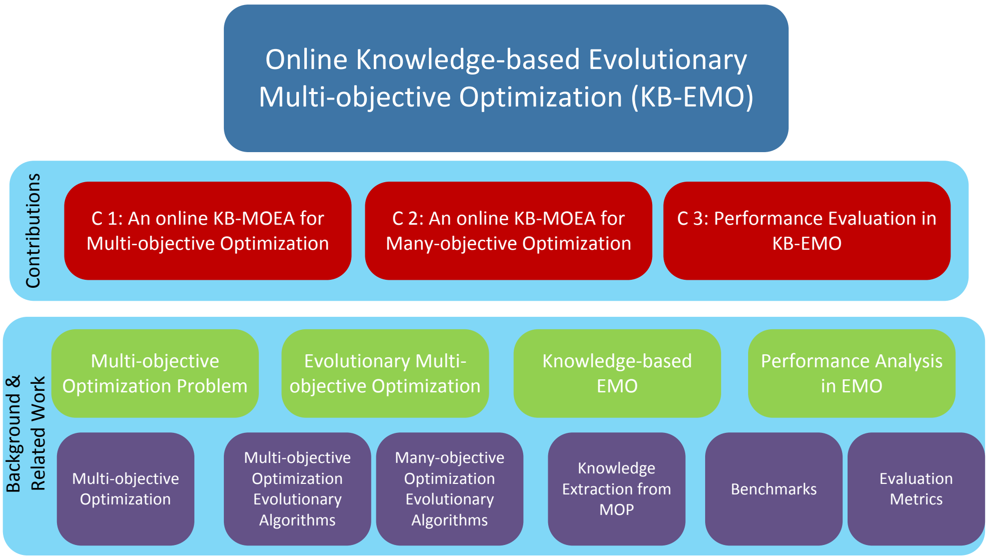


Figure 2.1: Background and Related Work of Concepts Covered in This Chapter in the Context of Thesis Contributions

2.1 Multi-objective Optimization Problems

In many real life cases, people are usually concerned about large and complex optimization problems involving several objectives (criteria), instead of a single one. A typical definition of MOP is as following (considering minimization of objectives without the loss of generality):

Definition:Multi-objective Optimization Problem (MOP)

$$\begin{aligned} &\text{Minimize} \quad && F(X) = \begin{pmatrix} f_1(X) \\ f_2(X) \\ \dots \\ f_m(X) \end{pmatrix} \\ &\text{subject to} \quad && X \in \Omega \\ &\text{where} \quad && \Omega = \{X | X \in \mathbb{R}^n, H(X) = 0 \text{ and } G(X) \leq 0\} \end{aligned} \tag{2.1}$$

Here, solution X is a n-dimensional vector $X = (x_1, x_2, \cdots, x_n)$ and Ω is the feasible set of solutions restricted by equality or inequality constraints and variable bounds. $(f_1, f_2, \ldots, f_m)$ is the corresponding objective vector of X of size m.

In real cases, the objectives $f_1, f_2, \cdots, f_m$ are usually conflict with each other and lead to tradeoffs among themselves. Hence, the optimal solution for MOPs is not a single solution, but a set of solutions instead. The optimality for MOP is based on a partial order relation, known as Pareto dominance and the related concepts are defined below:

Definition: Pareto Dominance

For two objective vectors $U = (u_1, u_2, \ldots, u_m)$ and $V = (v_1, v_2, \ldots, v_m)$ of MOP 2.1, U is said to dominate V $(U \prec V)$, if and only if all the m objective values of U are not greater than the objective values of V and at least one of them, say f_i, is strictly smaller.

$$\forall i \in \{1, 2, \ldots, m\} : u_i \leq v_i \wedge \exists i \in \{1, 2, \ldots, m\} : u_i < v_i$$

Definition: Pareto Optimality

Solution X^* is said to be Pareto optimal for MOP 2.1 if there is no solution that dominates X^*.

$$\{X | F(X) \prec F(X^*), X \in \Omega, X^* \in \Omega\} = \emptyset$$

Definition: Pareto Set (PS)

The Pareto set $\mathcal{P}^*$ of MOP 2.1 is the set of all the Pareto optimal solutions in decision space.

$$\mathcal{P}^* = \{X^* | X^* \in \Omega \wedge \nexists X \in \Omega, F(X) \prec F(X^*)\}$$

Definition: Pareto Front (PF)

The Pareto front $\mathcal{PF}$ of MOP 2.1 is the set of objective vectors of solutions in Pareto set.

$$\mathcal{PF} = \{F(X^*)|X^* \in \mathcal{P}^*\}$$

The concept of Pareto optimality is proposed early in [54][115][130] and a Pareto optimal solution means it is not possible to improve a given objective without deteriorating at least another. The Pareto front represents the best compromise between the different conflicting objectives.

2.1.1 MOP Complexity and Resolution Approaches

The resolution of MOP requires the identification of the Pareto optimality. However, this is not easy since most MOPs are NP-hard problems [135]. Generally, the complexity and difficulty when solving an MOP is:

- There are usually many or even infinite Pareto optimal solutions in the Pareto front. The number of Pareto optimal solutions increases according to the size of the problem, especially the number of objectives under consideration. It is clear that all Pareto solutions in an m-objective environment are still optimal of the same problem with an additional objective $(m + 1)$ [135]. The size of Pareto front can go exponential with respect to the number of objectives.

- The landscapes of objective functions of MOPs hinder the exploration of optimality with factors such as parameter dependency, modality, many-to-one mappings, non-differentiability and so on. The optimizers may either find required information like gradient unavailable or be trapped around the local optima in feasible areas. Sometimes for many problems in practical applications, one has to resort to simulation or physical models to evaluate the objective values, where the objectives cannot be formulated with analytical representation, referred to as black box optimization [83].

- The geometry of Pareto front challenges the exploitation of optimizers with complicated structures regarding linearity, convexity and continuity.

- The feasible area in decision space is commonly constrained. The distribution of optimal solutions in decision space is scattered or follow certain patterns and make it hard to be fully identified.

There exist exact algorithms when analysing and solving MOPs, which can return optimal solutions and guarantee their optimality, especially for bi-objective optimization problems, such as branch and bound algorithms [148][127], A* algorithms [131][99], dynamic programming algorithms [125] and so on. However, given the difficulties and complexities as above, metaheuristic methods are more popular when dealing with MOPs, with the aim to obtain high quality solutions in practical cases to approximate the Pareto optimality nicely instead guarantee of the global optima.

Generally, in objective space, the goal for solving an MOP is to achieve good convergence to the Pareto front and diversity along the entire front. The convergence is to minimize the distance from the obtained set of solutions to optimal solutions, while the diversity indicates uniform distribution of these solutions along the entire front so that no valuable information is lost. In decision space, the obtained set of solutions is required to represent the whole Pareto set sufficiently for the sake of diversity, flexibility, robustness and applicability to support design making and implementation. In these regards, MOEAs are more prominent for the tackling of MOP.

2.2 Multi-objective Optimization Evolutionary Algorithms

As a famous population-based metaheuristics, MOEAs are able to approximate the Pareto front in a single run. Compared to other methods, MOEAs enjoy a lot of attraction and effort from the community during the last few decades, and this is still one of the hottest research topics in the field of evolutionary computation [136]. We have witnessed quite a number of successful designs, to name a few in chronological order and their updates, Vector Evaluated Genetic Algorithms (VEGA) [119], Multi-objective Genetic Algorithm (MOGA) [59], Niched Pareto Genetic Algorithm (NPGA) [67], Non-dominated Sorting Genetic Algorithm (NSGA) [128], Strength Pareto Evolutionary Algorithm (SPEA) [172], SPEA2 [167], Pareto Archived Evolution Strategy (PAES) [84], Pareto Envelope-based Selection Algorithm (PESA) [35], PESA2 [32], Pareto-frontier Differential Evolution (PDE) [1], Fast Non-dominated Sorting Genetic Algorithm (NSGA2) [41], Multi-objective Evolutionary Algorithm based on Decomposition (MOEA/D) [162], IBEA [170], HypE [8], Direction-based Multi-objective Evolutionary Algorithm (DMEA) [21] and DMEA2 [109].

Since the huge volume of related literature[1], it's not easy to categorize the algorithms appropriately. From the perspective of evolutionary framework, there are Genetic Algorithm based approaches [41], Differential Evolution based approach [1][117], Particle Swarm based approach [30], and so on. From the perspective of fitness assignment, there are Pareto dominance based approaches, aggregation and decomposition based approaches and indicator based approaches, and so on. From the perspective of objective vector diversity, there are fitness sharing based approaches, distance based approaches, cell density based approaches, and so on. From the perspective of elitism implementation, there are combined population based approaches, external archive based approaches and so on. People can also classify MOEAs from the perspective of constraint handling and others. A brief categorization is shown in Table 2.1. The classification of approaches is extremely difficult determined by the broad variety of methodologies for MOEA design.

This section is not going to enumerate and review all the MOEA designs here. Instead, it reviews especially the recent developments to MOEAs and groups them mainly from the perspective of fitness assignment, and focus on the approaches that have been more popular in the research community. For comprehensive knowledge

[1]See the literature repository at http://delta.cs.cinvestav.mx/~ccoello/EMOO/ for reference, created and maintained by Professor Carlos A. Coello Coello.

about all the specific issues, we have several reviews and surveys in the literature [27][28][86][166][39].

2.2.1 Dominance Based Approach

Directly based on the definition of Pareto optimality, a non-dominated sorting method is proposed by David E. Goldberg for fitness assignment of the solutions in evolution [64]. The general idea is as following: first, identify the non-dominated solutions and assign the best rank value to them. Usually lower ranking corresponds to better solution quality. Next, exclude the solutions with best rank, identify the new non-dominated solutions from the rest and assign the next rank value to them. This process will be repeated until the all the solutions are ranked. By doing this, it actually maps the multi-objective space into a single dimension in order to allow a direct comparison among solutions and hence to establish preferences for selection operation. Influenced by his ideas, many actual implementations emerged.

Srinivas and Deb proposed Non-dominated Sorting Genetic Algorithm (NSGA) [128]. NSGA first ranks the individuals on the basis of non-dominated sorting. Then a fitness value is assigned to each non-dominated ranking levels. For solutions of each ranking, a fitness sharing method is used to maintain the diversity of the population. NSGA2 [41][44] also utilizes Pareto dominance to promise the convergence of the evolutionary search. In order to obtain a wide spread over the whole Pareto front, NSGA2 uses the crowding distance in objective space to maintain a diverse population. It is defined as the distance of two neighbouring solutions on either side of a solution for each objective. This density formulation helps preserve diversity since we only pick solutions with higher estimation. As a representative algorithm, NSGA2 has been successfully applied to a wide range of multi-objective problems. It is a benchmark algorithm for the performance analysis and comparison in the literature, also adopted in this thesis.

Pareto dominance based fitness assignment is popular in the design of MOEAs besides NSGA and NSGA2. They come with different diversity maintaining mechanisms and elitism mechanisms to form quite a number of variants. Some selected algorithms are introduced below.

Strength Pareto Evolutionary Algorithm (SPEA) was introduced by Zitzler and Thiele [171][172]. SPEA designs an external archive to store non-dominated solutions found previously. In each generation, non-dominated individuals are copied to the external archive. For each archived individual, a strength value proportional to the number of solutions which that individual dominates is calculated. The fitness is based on individual's strength value and takes both the distance to the Pareto front and distribution of individuals. SPEA2 [167] improves the design by: (1) finer-grained fitness design that is based on both the number of individuals that dominate the individual under evaluation and the number of individuals dominated by the individual under evaluation; (2) a nearest neighbour density estimation to improve the search efficiency, and (3) archive truncation that preserves the boundary solutions.

Pareto Archived Evolution Strategy (PAES) was introduced by Knowles and Corne [84]. PAES also utilizes a historical archive that stores the non-dominated solutions previously found for the evaluation of offspring individuals. This archive keeps the elitist and maintains diversity using a crowding procedure based on a grid division of the objective space. In improved versions, PESA [35], prefers the solutions with lower density in the cell of the grid partition and PESA2 [32] would assign higher probability to sparsely occupied cells, instead of solutions when selecting.

Table 2.1: Selected Multi-objective Evolutionary Algorithms

Algorithms	Author & Year	Evolutionary Framework	Fitness Assignment	Diversity	Elitism	External Archive
VEGA	Schaffer, 1985	GA Based	Altering Objective values	N/A	No	No
MOGA	Fonseca et al., 1993	GA Based	Dominance Based	Fitness Sharing	No	No
NPGA	Horn et al., 1994	GA Based	Pareto Dominance	Fitness Sharing	No	No
NSGA	Srinivas and Deb, 1994	GA Based	Pareto Dominance	Fitness Sharing	No	No
NSGA2	Deb et al., 2000	GA Based	Pareto Dominance	Crowding Distance	Yes	No
SPEA	Zitzler and Thiele, 1999	GA Based	Dominance Based	Density Based	Yes	Yes
SPEA2	Zitzler, 2001	GA Based	Dominance Based	Density Based	Yes	Yes
PAES	Knowles and Corne, 2000	GA Based	Dominance Based	Density Based	Yes	Yes
PESA	Corne et al., 2000	GA Based	Dominance Based	Density Based	Yes	Yes
PESA2	Corne et al., 2001	GA Based	Dominance Based	Density Based	Yes	Yes
PDE	Hussein et al., 2001	Differential Evolution	Pareto Based	Crowding Distance	Yes	No
SMS-MOEA	Beume et al., 2006	GA Based	Hypervolume Based	N/A	Yes	No
CCEA	Tan, 2006	Operative Coevolution	Pareto Dominance Based	Niche Count	Yes	Yes
MOEA/D	Zhang and Li, 2007	GA Based	Aggregation	Reference Points	Yes	No
IBEA	Zitzler and Künzli, 2004	Genetic Algorithm	Indicator Based	N/A	Yes	N/A
DMEA	Bui et al., 2011	Differential Evolution	Pareto Dominance	Reference Rays	Yes	Yes
DMEA2	Nguyen et al., 2014	Differential Evolution	Pareto Based	Reference Rays	Yes	Yes

2.2.2 Aggregation and decomposition based approach

Using aggregation functions to evaluate solutions is another typical approach [80]. The basic idea of aggregation based decomposition is based on the observation that a Pareto optimal solution to the original MOP is an optimal solution of an aggregate function of the entire original f_i's. By optimizing a set of such different aggregate functions simultaneously, we can expect the set of solutions to approximate the Pareto front of original MOP. If the aggregate functions are organized properly in a way, such as by using a set of evenly spread reference points, the solutions returned are supposed to have a similar distribution along the Pareto front. Hence, this approach first decomposes an MOP into a number of subproblems in aggregate function form. Aggregate function can be constructed via weighted sum approach [104], Tchebycheff approach [104] and the boundary intersection method [36][100]. Then optimizes them collaboratively.

The best representative of the decomposition based algorithm is MOEA/D [162]. It is based on conventional aggregation approaches and it explicitly decomposes an MOP into scalar optimization subproblems and solves these subproblems simultaneously by evolving a population of solutions. The objective of each subproblem, is an aggregation of the individual objectives. At each generation, the population is composed of the best solutions found so far (i.e. since the start of the run of the algorithm) for each subproblem. Each subproblem (i.e., scalar aggregation function) is optimized in MOEA/D by performing evolutionary operations on its own solution or using information only from its neighbouring subproblems. The neighbourhood relations are defined based on the distances between their aggregation coefficient vectors, based on the assumption that the optimal solutions to two neighbouring subproblems should be very similar.

Several improvements on MOEA/Ds have been made recently. Li and Zhang [95] suggested using two different neighbourhood structures for balancing exploitation and exploration. Zhang et al. [163] proposed a scheme for dynamically allocating computational efforts to different subproblems in an MOEA/D in order to reduce the overall cost and improve the algorithm performance. This implementation of MOEA/D is efficient and effective and has won the Congress on Evolutionary Computation (CEC) 2009 MOEA competition. Nebro and Durillo [106] developed a thread-based parallel version of MOEA/D, which can be executed on multicore computers. Ishibuchi et al. [77] proposed using different aggregation functions at different search stages. MOEA/D algorithms have also been successfully applied to a number of applications, such as the flowshop scheduling problem [23] and sensor network routing [87].

2.2.3 Indicator Based Approach

The quality of an approximated Pareto front can be measured by performance indicators (reviewed in Section 2.5), such as the generational distance and hypervolume. The design of MOEAs can be directly based on an indicator to guide the search, particularly to perform environmental selection.

Zitzler and Knzli first suggested a general indicator-based evolutionary algorithm (IBEA) [170]. This framework incorporates any indicator to compare a pair of candidate solutions. An indicator-based model for handling uncertainty is proposed by Basseur and Zitzler [15]. General approach to incorporate objective reduction techniques into hypervolume-based algorithms is discussed by Brockhoff and Zitzler [20] and different objective reduction strategies are studied for improving the performance of hypervolume-based MOEAs. Bader and Zitzler suggested a fast hypervolume-based MOEA for many-objective optimization [8]. To reduce the computational overhead in hypervolume computation, a fast method based on Monte Carlo simulations is proposed to estimate the hypervolume value of an approximation set. Therefore, the proposed hypervolume-based MOEA is reported to be applied to problems with many objectives. Very recently, they [7] further investigated the robustness of hypervolume-based multi-objective search methods. Three existing approaches for handling robustness in the area of evolutionary computing, modifying the objective functions, additional objectives, and additional robustness constraints, are integrated into a multi-objective hypervolume-based search. An extension of the hypervolume indicator is also proposed for robust multi-objective optimization.

2.2.4 Other Approaches

Besides the algorithms described above, many other approaches are utilized for appropriate MOEA design from different perspectives. For example, while the size of Pareto front can be very large or infinite, the decision maker may be only interested in part of solutions they prefer. Hence, such information can be useful to guide the evolutionary search. Many attempts fall into this category even before the introduction of user preference to MOEA design [139]. For example, see [2][3]. Deb et al. suggested an interactive MOEA based on reference directions that decision makers provide to guide the search towards the interested region [43]. The author also proposed an interactive MOEA with an approximate value function progressively and periodically generated every few generations [47]. Several non-dominated solutions are presented to the decision maker for ranking from the worst to the best to take preference information into consideration. Thiele et al. represent preferences interactively in the form of reference points for the calculation of an achievement scalarization function [141]. The selection is expected to lead the search to focus on the most interesting parts with the support of achievement scalarization functions, which project a given reference point onto the Pareto optimality.

Another important point of concern is that genetic search is not well suited for fine-tuning structures which are too close to optimal solution [64]. Many algorithms started to introduce local search models into the population-based global search methods, which resulted in the development of memetic computing by simple hybridization and incorporation of separate rule based meme population for coevolution [113]. In a simple memetic algorithm, a population of solutions is initialized. After selection, reproduction operations, a local improvement procedure such as hill climbing is highlighted. The update to the population can use either Lamarckian inheritance or Baldwinian inheritance [24]. The process repeats and terminates when a criterion is satisfied. Multiple local search operators can also be used within this evolutionary system [22][108]. Coevolving memetic algorithms further include a population of memes and letting the gene and meme populations progress cooperatively or competitively [124]. The representation of meme for this coevolution adopts the form of rules which undergo their own initialization, variation, fitness assignment, selection and replacement operations. The rules serve the purpose of sampling solutions from neighbourhood after pairing with certain solutions. The memetic computation framework combined with the fitness assignment approaches explained in this section can be applied to deal with MOPs. There are multi-objective memetic algorithms using either dominance relations or aggregation function proposed in the literature [107] shown as competitive problem solvers for domains such as scheduling [14], decision making [121], engineering design [65] and game development [102].

The design of MOEAs is an open area and new ideas are also developed. Recently, Bui et al. proposed a novel direction-based multi-objective evolutionary algorithm (DMEA) [21], showing competitive results over the classic algorithms. In DMEA, two types of direction are of interests: convergence direction and spread direction. The direction information is obtained from the current population and archived non-dominated solutions are used to perturb the current parental population to produce offspring and then to create both the next-generation archive and parental pools. The diversity is maintained using niching criterion based on emit rays from ideal point into the hyper-quadrant that contains the current front estimate. In the improved version DMEA2 [109], an adaptation of the balance between convergence and spread direction, ray-based density niching and corresponding updated selection mechanism are highlighted.

2.2.5 Section Summary

The current MOEAs attempt to maintain the balance between convergence and diversity for optimization to approximate the entire Pareto front in objective space. Convergence requires to minimize the distance between populated solutions and the real Pareto front while the diversity encourages the population to occupy and represent the whole front [96]. When emulating natural selection for the purpose of optimization, an MOEA has to implement a fitness measurement to guide the evolution to converge, and also design diversity maintaining method to get a wide and even distribution. We briefly classified current algorithms mainly based on the fitness assignment and reviewed the mechanisms for both convergence and diversity. Due to the considerable volume of research on evolutionary multi-objective optimization, we only pick the representative algorithms for each category.

2.3 MOEA for Many-objective Optimization

Recently, many-objective optimization (typically four or more objectives involved), has attracted increasing attention in evolutionary multi-objective optimization community. The research is first motivated by real world applications optimizing a high number of objectives, e.g., control system design [66], industrial scheduling [134], software engineering [105]. Second, there is no effective optimizer to solve these problems in practice. The popular Pareto-dominance based MOEAs, such as NSGA-II [44] and SPEA2 [167], have encountered great difficulties in many-objective optimization, although they have shown excellent performance on problems with two or three objectives. Detailed information over the recent development regarding many-objective optimization is provided in survey [94].

Generally, the current algorithms may face the following difficulties [42][78]:

- Explosion of non-dominated solutions: When the number of objectives grows, the number of non-dominated solutions to one randomly generated solution will grow exponentially [61]. This will occupy almost all the population slots very soon. When an algorithm utilize non-dominated relations for selection, this will lead to the severe loss of selection pressure towards the Pareto front when handling many objectives [160]. Therefore, the overall performance of corresponding algorithm will deteriorate significantly.

- Inefficient diversity measure: Taking crowding distance in NSGA2 as an example, first, the identification of neighbour solutions for the calculation of hyper-cuboid distance is computationally expensive in a comparatively larger objective space. Second, the crowding distance in many-objective environments will cause an unacceptable distribution of solutions at the end [78]. New mechanism for the estimation of diversity is required in evolutionary many-objective computation.

- Inefficient recombination operation: In a many-objective problem, the distribution of Pareto optimal solutions are likely to be widely distant from each other and sparsely located in the objective space. In this case, extra effort should special recombination operators (mating restriction or other schemes) may be necessary for handling many-objective problems efficiently.

- Representation of Pareto Front: In order to fully represent the high dimensional Pareto front, exponentially more solutions are required. This will cause pressure on the limited population size in MOEAs. In addition, it will be difficult for a decision-maker to understand and implement a preferred solution from a much larger population for the representation of the resulting Pareto-optimal surface.

- Expensive computation for performance evaluation: Since more decision and performance variables are introduced in many-objective optimization problems, comparatively more computational effort is required for the evaluation of performance metrics. For instance,the computing of hypervolume requires exponentially more computations with the increase in objective numbers [60][150]..

The difficulties listed above are new challenges for current MOEAs working well with 2 or 3 objectives. They overall require certain modifications to existing methodologies for the improvement to their performance in many objective environments.

2.3.1 Current Methodologies

Very intuitively, the very simple starting point for tackling many objectives is by the reduction of the number of objectives. If we can choose only a few important objectives, almost all difficulties in evolutionary many-objective optimization are e-liminated and we successfully change a many-objective problem to a multi-objective problem applicable by traditional methods. Deb and Saxena [46][45] proposed an objective reduction method, which is based on principle component analysis. Their idea is to remove unnecessary objectives while maintaining the shape of the Pare-to front in the reduced objective space. On the other hand, Brockhoff and Zitzler [19][20] proposed a different idea where objective reduction is based on Pareto dom-inance. That is, an objective is removed when it does not change (or only slightly change) the Pareto dominance relation among solutions.

To overcome the drawback of Pareto-dominance based MOEAs, some efforts have been made in the literature [78]. Many improvements to the current designs have been proposed. The developing techniques can be roughly classified into the following two groups: Adoption of new preference relations and adoption of new diversity promotion mechanisms.

Since the Pareto-dominance relation scales poorly in many-objective optimiza-tion, it is natural to use other preference relations, including modified Pareto-dominance and different ranking schemes, so as to produce fine selection pressure towards Pareto front. Up to now, many alternative preference relations have been proposed.

Modification of Pareto dominance in EMO algorithms has often been discussed in the evolutionary multi-objective optimization community. Sato et al. [118] demonstrated that the use of a modified dominance clearly improved the perfor-mance of NSGA2 for many-objective problems. Instead of the standard Pareto dominance, a wider angle of the dominated region of a solution is suggested to refuse large deterioration in one objective for the tradeoff of a small improvement in another objective. Similar ideas, such as the concept of α-dominance [70], preference incorporation [18][17], are also discussed in the literature.

Besides the modification of standard Pareto dominance, different ranking methods are also recommended. Drechsler et al. [52] proposed the use of a relation called favour to differentiate between non-dominated solutions for the handling of many-objective problems. They defined the relation favour based on the number of objectives for which one solution is better than the other. The relation favour was modified by taking into account not only the number of objectives for which one solution is better than the other but also the difference in objective values between the two solutions [134]. Corne and Knowles [33] reported that the best results were obtained using a simple average ranking method. Kukkonen and Lampinen [91] examined the average and minimum ranking methods. More complicated ranking methods based on ϵ-dominance and fuzzy Pareto dominance are also analysed [88]. The introduction of different ranks to non-dominated solutions for the purpose of increasing selection pressure toward the Pareto front seems to decrease the diversity of solutions. In some cases, the population converges to a few solutions (or a single solution) as reported in [33]. In short, the performance of approaches above has not been validated systematically.

Adoption of new diversity promotion mechanisms is also considered. In many-objective optimization, Pareto-dominance could not provide sufficient selection pressure to make progress in a given population, so the diversity criterion begins to play a key role in such cases. However, the existing diversity preservation criteria, such as crowding distance [44], are not suitable for many-objective problems [88][90]. Thus, a new mechanism promoting the diversity is needed. Adra and Flemming proposed two mechanisms for managing diversity and investigate their impacts on overall convergence in many objective optimization [5]. The recently proposed NSGA3 [42] replaced the crowding distance operator in NSGA2 with an association operator aided by a set of well-distributed reference points. Hence, NSGA3 is more dedicated for many-objective optimization only and it utilizes a set of reference points to achieve diversity. Solutions that are closer to reference points are more preferred. Hence, if the reference points are evenly spread, good distribution of solutions in objective space can be induced.

It is worth noting that, unlike Pareto-dominance based MOEAs, other two class of approaches in Section 2.2, decomposition based and indicator-based approaches, have been found very promising in many-objective optimization. The former decomposes a problem with multiple objectives into a set of single-objective subproblems and the latter employs a single performance indicator, such as hypervolume, to optimize a desired property of evolutionary population. One issue is regarding the exact calculation of hypervolume is especially computationally expensive in high dimensional objective space, the fast algorithm [8] that uses Monte Carlo simulation to approximate the exact hypervolume values has been developed.

2.3.2 Section Summary

During the rapid growth of evolutionary multi-objective optimization, algorithms for problems with 2 and 3 objectives seem to be well-established regarding the objective space and now the research trend moves to many objective environments with four and more objectives. It has been one of the major research topics for the community in recent years. However, this is still an open area and no dominating algorithms exist. The current methodologies still suffer limitations regarding many issues. For instance, in reference points related algorithms, such as NSGA3 and MOEA/D, the reference points or weight vectors are likely to be distributed very sparsely in the objective space and hence lead to inefficiency when representing the whole Pareto front.

2.4 Test Problems for MOEAs

Test problems play an important role in performance evaluation under controlled experimental conditions and hence development of MOEAs. Historically the two topics in multi-objective evolutionary optimization, algorithm development and test problem design, have been developed hand-in-hand and have helped promote each other [74].

Generally, test problems are synthetically constructed and have tunable features, such as number of decision variables, to generate a number of specific problem instances with varying degrees of complexities. Test problem design is a well-researched area in EMO literature. In this section, we only go through some prominent test suites.

Since 1990s, the construction of test problems has drawn special interest from the research community and many instances are proposed since then. To name a few, we have Binh and Korn function [16], Farina Function [55], Fonseca and Fleming Function [58], Ishibuchi and Murata Function [76], JOS Function [81], LDZ Function [93], TKLY Function [137], Schaffer Function [119], Kursawe Function [92], and so on. The utility of these functions, however, depends on a number of considerations, such as dimensionality scalability in objective space and decision space, modality in function landscapes and other features. Huband et al. [69] provides a good discussion of different test functions introduced in EMO. A more recent review can be found in [74].

2.4.1 Classic Benchmarks

The ZDT [169] test suite was introduced in 2000 and is a widely adopted benchmark in the field of EMO. Some commonly used ZDT instances are shown in Table 2.2. It provides test problems using a modularized structure. Theoretically, the Pareto set is allocated at $x_1 \in [0, 1]$ and the other $x_i = 0$, forming a simple pattern in the decision space. ZDT benchmark is usually criticized for the bi-objective design and the extreme location of optima.

DTLZ [50] is another benchmark test suite with richer features than ZDT, such as scalability in the number of objectives, scalability in the number of decision variables to control the location of Pareto front and support generation of various geometries of Pareto front. This flexibility makes it a suitable candidate to be used in multiple and many objective optimization problems [42]. Some selected instances are shown in Table 2.2. The set of Pareto solutions of m-objective DTLZ problems is $0 \leq x_i \leq 1, i = 1, 2, \ldots, m-1$ and $x_i = 1/2, i = m, m+1, \ldots, n$ except DTLZ6,

Table 2.2: ZDT and DTLZ Benchmarks: # of Variables(n) and # of Objectives(m)

Problem	Objective Functions	Decision Space	n/m
ZDT1	$f_1 = x_1, f_2 = g \cdot h$ $g = 1 + 9 \cdot \Sigma_{i=2}^{n} x_i/(n-1)$, $h = 1 - \sqrt{f_1/g}$	$x_i \in [0,1]$	30 / 2
ZDT2	as ZDT1, except $h = 1 - (f_1/g)^2$	$x_i \in [0,1]$	30 / 2
ZDT4	as ZDT1, except $g = 1 + 10(n-1) + \Sigma_{i=2}^{n}(x_i^2 - 10\cos(4\pi x_i))$	$x_1 \in [0,1]$, $x_{2,\ldots,n} \in [-5,5]$	10 / 2
ZDT6	$f_1 = 1 - exp(-4x_1)\sin^6(6\pi x_1)$ $g = 1 + 9 \cdot (\Sigma_{i=2}^{n} x_i/(n-1))^{0.25}$ $h = 1 - (f_1/g)^2$	$x_i \in [0,1]$	10 / 2
DTLZ1	$f_1 = 0.5(1+g)\Pi_{i=1}^{m-1} x_i$ $f_{i=2:m-1} = 0.5(1+g)(\Pi_{j=1}^{m-i} x_j)(1 - x_{m-i+1})$ $f_m = 0.5(1+g)(1-x_1)$ $g_1 = 100[k + \Sigma_{i=1}^{k}((x_{i+m-1} - 0.5)^2 - \cos(20\pi(x_{i+m-1} - 0.5)))]$	$x_i \in [0,1]$	7 / 3
DTLZ2	$f_1 = (1+g)\Pi_{i=1}^{m-1}\cos(\frac{\pi}{2}x_i)$ $f_{i=2:m-1} = (1+g)(\Pi_{j=1}^{m-i}\cos(\frac{\pi}{2}x_j))\sin(\frac{\pi}{2}x_{m-i+1})$ $f_m = (1+g)\sin(\frac{\pi}{2}x_1)$ $g_2 = \Sigma_{i=1}^{k}((x_{i+m-1} - 0.5)^2$	$x_i \in [0,1]$	12 / 3
DTLZ3	As DTLZ2, except g is replaced by g_1	$x_i \in [0,1]$	12 / 3
DTLZ4	As DTLZ2, except x_i is replaced by x_i^{100}	$x_i \in [0,1]$	12 / 3
DTLZ5	As DTLZ2, except $x_{2,\ldots,m-1}$ is replaced by $(1 + 2gx_i)/(2(1+g))$	$x_i \in [0,1]$	12 / 3
DTLZ6	As DTLZ5, except g is replaced by $g = \Sigma_{i=1}^{k} x_{i+m-1}^{0.1}$	$x_i \in [0,1]$	12 / 3

where $x_i = 0, i = m, m+1, \ldots, n$.

Huband et al. [69] provides a rigorous review of the test problems for multi-objective optimization. They consider fitness landscape and Pareto front geometry properties and provide a set of recommendations and features that a good test suite should follow. A summary of these recommendations is given in Table 2.3. A group of test problems, named WFG, are then proposed. The Pareto set for WFG problems forms a simple pattern similar to ZDT and DTLZ and the first few variables are specified over the whole range and others at certain values.

Overall, the existing benchmarks mainly concentrate on function landscape design and the front geometries in the objective space. It's worth noting that the variable dependencies and separability has been considered in WFG problems [69]. A group of modified ZDT problems also looked at enhancing the linkage between

Table 2.3: Features for EMO Test Problems [69]

Feature (F)	Comment
F1: Pareto Optimal Geometry	Linear, Convex, Concave, Mixed, Degenerate, Disconnected, or some combination
F2: Parameter Dependencies	separable or non-separable
F3: Bias	Substantially more solutions exist in some regions of fitness space than they do in others
F4: Many-to-one mappings	Pareto one-to-one/many-to-one, flat regions, isolated optima
F5: Modality	uni-modal or multi-modal (possibly deceptive multi-modality)

decision variables [48]. However, the distribution of Pareto set in decision space is not considered under these problems. The simple rules emerged in the benchmarks above are defined on the whole range of some variables. They are useful but not sufficient for comprehensive performance analysis for algorithms in design space.

2.4.2 Test Problems with Complicated Pareto Sets

Although the classic benchmarks are used widely in the literature, one argument over them is the shape simplicity of Pareto sets [111, 95]. Some research effort has been made on the construction of test problems which consider shapes of Pareto sets in their design. [111] proposed a mechanism to generate complicated Pareto sets in parameter space through a series of mappings for mainly bi-objective optimization problems. Instances of their test problems are shown in Table 2.4. A set of more complicated test instances are recently provided with quadratic or trigonometric relationships for the distribution of Pareto sets [95, 164], as shown in Table 2.4.

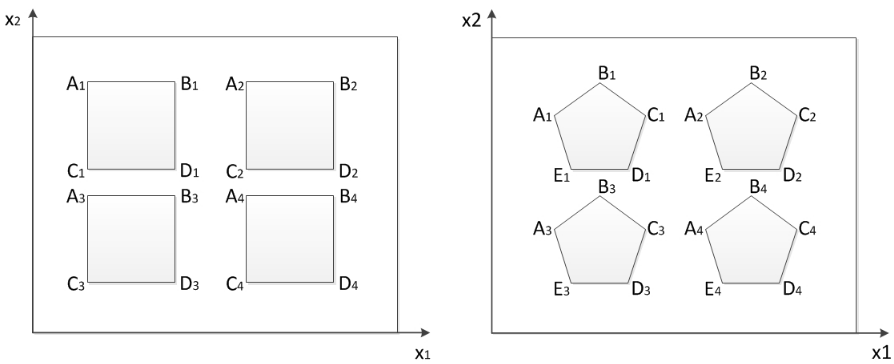
While the above problems aim at introducing difficult Pareto front and set geometries for MOEA evaluation, they are not specifically designed for the performance evaluation of MOEAs in the decision space.

Table 2.4: OKA and LZ Test Problems (examples shown, not all the instances)

Problem	Objective Functions	Pareto Optimality
OKA1	$f_1 = x'_1$ $f_2 = \sqrt{2\pi} - \sqrt{\lvert x'_1\rvert} + 2\lvert x'_2 - 3\cos(x'_1) - 3\rvert^{\frac{1}{3}}$ $x'_1 = \cos(\frac{\pi}{12})x_1 - \sin(\frac{\pi}{12})x_2$ $x'_2 = \sin(\frac{\pi}{12})x_1 - \cos(\frac{\pi}{12})x_2$	$\mathcal{PS}:$ $x'_2 = 3\cos(x'_1) + 3$ $\mathcal{PF}:$ $f_2 = \sqrt{2\pi} - \sqrt{f_1}$
LZ1	$f_1 = x_1 + \frac{2}{\lvert J_1\rvert}\Sigma_{j\in J_1}(x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$ $f_2 = 1 - \sqrt{x_1} + \frac{2}{\lvert J_2\rvert}\Sigma_{j\in J_2}(x_j - x_1^{0.5(1.0+\frac{3(j-2)}{n-2})})^2$ $J_1 = j\lvert j$ is odd and $2 \leq j \leq n$ $J_2 = j\lvert j$ is even and $2 \leq j \leq n$	$\mathcal{PS}:$ $x_j = x_1^{0.5(1.0+\frac{3(j-2)}{n-2})}$ $j = 2, \ldots, n$

2.4.3 Test Problems with Pattern-based Pareto sets

As noted before, introduction of test problems which explicitly focus on decision space performance evaluation of MOEAs is an important direction and can have important implications for better design of MOEAs, specifically the more recent knowledge-based MOEAs. In this regard, [73] has recently proposed a polygon-based test problem for decision space diversity investigation and compared the related performance of several mainstream MOEAs on these problems [72, 79].



(a) Square Based 4 Objectives (b) Pentagon Based 5 Objectives

Figure 2.2: Ishibuchi et al.'s Multi-Objective Test Problems for Decision Space Diversity Investigation to Minimize the Distances to Vertices A_is, B_is, C_is, … Simultaneously

The Polygon based test problems for the investigation of decision space diversity is defined as shown in Figure 2.2. The number of objectives is in fact equal to the number of chosen polygon vertices. Each objective function corresponds to minimization of the distance to one of the vertices at the same position of corresponding polygons in a 2-dimensional decision space. The Pareto set clusters in optimal areas bounded by the polygons (including edges) as shown in grey in the figure. The problem design allows better visualization, scalability of function numbers supporting many objective optimization, scalability of optimal area numbers increasing the diversity of decision space and also creating distribution patterns in decision space. However, there are also a number of limitations of this test problem. First, the test function is defined in a two-dimensional decision space and it is not clear if it can scale to higher dimensions. Second, no mechanisms are provided for defining fitness landscapes and front geometries. In addition the function definition lacks in explaining controls for function modality, parameter dependencies, front convexity and other features.

2.5 Performance Metrics

Since the optimal solutions to an MOP, specifically a continuous optimization problem, can be unlimited, performance metrics are introduced to measure the quality of obtained solutions and benchmarking either different variants of the same MOEA (or an MOEA run with different parameter settings) or compare performance of different MOEAs on same problems. Zitzler et al. [169] suggested three aspects of MOEA search for consideration in objective space: 1) distance from the obtained set to the global Pareto optima; 2) distribution of obtained solutions in objective space; and 3) the extent of the obtained solutions along the whole Pareto front. Effectively, the first consideration focuses on evaluating the convergence while the rest two emphasize diversity measurement of obtained solution sets [158]. The following broad classification can be used to introduce current metrics.

2.5.1 Convergence Focused Metrics

The metrics under this category are used to measure the extent to which the true PF is approximated or deviation from the true PF. There are two approaches in this category which can be grouped either based on the number of Pareto optimal solutions in the obtained set, or based on the distance from the obtained set to the global optima.

- *Ratio of Non-dominated Solutions [138]:* This measurement records the percentage of non-dominated solutions in the obtained set and is defined as $|S_{ND}|/n_P$, where $|S_{ND}|$ is the number of non-dominated solutions in the population of size n_P.

- *Error Ratio [147]:* Non-dominated solutions do not necessarily imply Pareto optimality. This measurement is interested in the ratio of optimal objective vectors in the obtained set and is evaluated as $\sum_{i=1}^{n_P} e(X_i)/n_P$, where $e(X_i)$ is 0 if X_i is Pareto optimal, 1 otherwise.

- *Nondominance Ratio [63]:* This metric takes two or more different obtained sets into consideration, say $A_1, A_2, \ldots, A_{n_A}$ and evaluate the contribution of these different sets by comparing the number of non-dominated solutions in these sets against number of non-dominated solutions across all the sets. Let B denote the set of non-dominated solutions in $\cup_{i=1}^{n_A} A_i$, the measurement of set A_i is then given by: $|A_i \cap B|/|B|$.

- *Generational Distance (GD) [147]:* GD is a statistics that tries to capture the distance of the obtained set from the true PF. Let P^* be a set of optimal solutions uniformly distributed on the PF of a MOP and P an approximate of P^*. Then the GD value is calculated as

  $$\mathrm{GD}\left(P, P^*\right) = \frac{\sum_{s \in P} d\left(s, P^*\right)}{|P|}$$

  where $d\left(s, P^*\right)$ is the minimum Euclidean distance from solution $s \in P$ to all the optimal solutions in P^*.

- *Maximum Pareto Front Error [147]:* Different from the GD that uses the average, this measurement is interested in measuring the worst case performance, defined as $\max_{s \in P} d\left(s, P^*\right)$.

2.5.2 Diversity Focused Metrics

Metrics falling into this category are dedicated to evaluate the spread of the obtained set. They can be used to rank obtained sets sharing similar convergence measurements.

- *Spacing [120, 112]:* It estimates how evenly the distribution of obtained solutions is, defined as

$$\sqrt{\frac{1}{n_P - 1}\sum_{i=1}^{n_P}(d_i - \bar{d})^2}$$

where $d_i = \min_j(\sum_{k=1}^{m} |f_k(s_i) - f_k(s_j)|)$ and s_j is the solution in the obtained set but different from s_i, $\bar{d}$ is the average of d_i.

- *Maximum Spread [169, 159]:* This metric evaluates the coverage of the obtained set over PF, given by:

$$\sqrt{\frac{1}{m}\sum_{i=1}^{m}\left[\frac{\min(\max(f_i), \max(f_i^*)) - \max(\min(f_i), \min(f_i^*))}{\max(f_i^*) - \min(f_i^*)}\right]^2}$$

2.5.3 Metrics Evaluating Both Convergence and Diversity

Metrics in this group are able to reflect both the convergence and diversity of objective vectors. They comparatively enjoy more application in the literature. However, their accuracy is highly affected by the setting of reference points used in the calculation [75, 6].

- *Inverted Generational Distance (IGD) [31, 75]:* It can measure both the convergence and diversity of an approximation set to the optima. Let P^* be a set of optimal solutions uniformly distributed on the PF of a MOP and P an approximation of P^*. Then the IGD value is calculated as

$$\mathrm{IGD}\left(P^*, P\right) = \frac{\sum_{s^* \in P^*} d\left(s^*, P\right)}{|P^*|} \tag{2.2}$$

where $d(s^*, P)$ is the minimum Euclidean distance from the optimal solution s^* in P^* to the solutions in P.

- *Hypervolume [168, 6, 149]:* It measures the volume of the objective subspace that is dominated by the current obtained solution set A under evaluation, defined as (with a reference set R) $\lambda(H(A, R))$ where $H(A, R)$ denotes the area dominated by solutions in A but dominates the reference points in R and λ stands for the Lebesgue measure as $\int 1_{H(A,R)}(z)dz$.

2.5.4 Solow-Polasky Diversity Measure for Pareto Set

The Solow-Polasky Diversity (SPD) [126] is dedicated to evaluate the diversity of decision vectors, instead of objective vectors and hence is a decision space measure [145, 143]. For a set of output solutions P from an MOEA, this measure is calculated as:

$$SPD = eM^{-1}e^T \tag{2.3}$$

where M is a square matrix with each element defined as

$$m_{ij} = exp(-\theta \times d(s_i, s_j))$$

. Coefficient vector e is $(1, 1, \ldots, 1)$ and $\theta = 1$ is suggested.

The properties of Solow-Polasky diversity measure is summarized as follows [145, 143]:

- *Monotonicity in Varieties:* $SPD(P) < SPD(P \cup s)$ if solution $s \notin P$. Introducing new solution to P will increase its diversity.

- *Twinning:* $SPD(P) = SPD(P \cup s)$ if solution $s \in P$. The duplication of solutions won't change the diversity of set under evaluation.

- *Monotonicity in Distance:* $SPD(P) \leq SPD(Q)$ if $d(X_i, X_j) \leq d(Y_i, Y_j)$ for all pairs of solutions $X_i, X_j \in P$ and $Y_i, Y_j \in Q$. This enables the performance comparison between different sets using this measurement.

Although the metric have some good features, its limitations include a very high computational complexity requiring pair-wise distance calculations between all solutions as well as an expensive matrix inversion, obscure definition and complicated computations which are difficult to interpret.

2.5.5 Section Summary

From the brief review over test problems and performance metrics existed in the literature, we can clearly see they emphasize on the objective space, rather than the decision space. The current designs mainly focus on creating challenges and quantifying performance regarding the convergence or diversity or both to show how good MOEA performs in identifying and approximating the Pareto front, but seldom considers and evaluates Pareto set. Only some of them, such as the Ishibuchi problems and Solow-Polasky Diversity, touch the distribution of Pareto sets. However, they have many limitations the application in evolutionary multi-objective optimization research is highly constrained. The current lack of appropriate evaluation tools motivates us to design new benchmarks for MOEA in order to evaluate the performance relating decision space from new perspectives.

2.6 Knowledge Extraction from Multi-objective Optimization

The MOEA community has witnessed considerable volume of research dedicated to enhancing evolutionary algorithms via machine leaning techniques or adapting evolutionary algorithms for machine learning purposes in the literature.

When using machine learning techniques for better design of evolutionary algorithms, one important point is that the evolution in process has stored ample information about the MOP, search space, problem features, and population diversity. The information, patterns and knowledge extracted from these pieces of information can be utilized to make the evolutionary search process more effective and efficient. This can be done by embedding learning capabilities into the design of evolutionary algorithms or through interactive ways to provide knowledge or preference to guide the evolutionary search. For instance, a knowledge model is integrated into an evolutionary algorithm for flexible job-shop scheduling problems [156] and domain knowledge is provided interactively for the analysis of capability planning problems [152][157]. The authors extended the knowledge assisted evolutionary approaches to deal with complex group decision making [153] and resilient project scheduling [154]. Knowledge-based interactions are used for a multi-agent genetic algorithm design for global numerical optimization [165]. If the knowledge is used to enhance the local search, the problem can also be solved using coevolving memetic algorithms which includes an independent rule-based meme population that progresses adaptively with the solution population for efficient sampling from the neighbourhood of a solution [124].

At the same time, many attempts have been made to apply variants of evolutionary algorithms as types of effective and efficient techniques to extract knowledge, patterns and useful information. As an example, multi-objective evolutionary algorithms have been utilized to detect interesting local community structures in signed social networks [97][98]. Evolutionary algorithms are also reported for knowledge extraction purposes in [37][82]. Multi-objective memetic algorithms are also employed for association rule mining [53], clustering [68], tree induction [122], and high order learning [101].

Although there is a big overlap of these two fast growing fields of machine learning and evolutionary computation, in this section, we focus on the knowledge extraction from multi-objective optimization problems. The knowledge here is general, not constrained by the form of rules mentioned in Section 5.1. The discovery of knowledge from a MOP context is then briefly reviewed. A simple taxonomy of the related work is based on the time and source of knowledge extraction, the form of knowledge, the methodology and purpose of knowledge extraction. Some selected publications are listed in Table 2.5.

First, the knowledge discovery process can occur either before or after optimization process for non-dominated solutions. Before the optimization starts, the

Table 2.5: Selected Publications on Knowledge Extraction from Multi-objective Optimization

Publication	Type	Optimization + Knowledge Extraction Methodology	Knowledge Forms
Michelena and Agogino, 1988 [103]	a priori	Hypergraph Partitioning	N/A
Sarkar et al., 2008 [116]	a priori	Singular Value Decomposition (SVD)	Semantic patterns
Obayashi and Sasaki, 2003 [110]	posteriori	MOGA + Self Organizing Map	Clusters
Sugimura et al., 2007 [133]	posteriori	MOGA + Decision Tree Analysis or Rough Set Theory	Rules
Chiba et al., 2007 [25]	posteriori	Adaptive Range MOGA + Self Organizing Map	Clustering
Ulrich et al., 2008 [144]	posteriori	SPEA2 + Clustering	Dendrogram
Sugimura et al., 2009 [132]	posteriori	Kriging Surrogate Model Based + Association Rule Mining	Quantitative rules
Doncieux and Hamdaoui, 2011 [51]	posteriori	NSGA2 + Regression	Analytical Function
Kudo and Yoshikawa, 2012 [89]	posteriori	NSGA2 + Multidimensional Scaling	Isomap
Bandaru and Deb, 2013 [12]	posteriori	NSGA2 + Clustering	Polynomials
Tatsukawa et al., 2013 [140]	posteriori	NSGA2 + Multi-objective Genetic Programming	Analytical Relationships
Ulrich, et al., 2013 [142]	online	Biobjective Partitioning Optimizer - PAN	Clusters
Zhang, et al.,2014 [161]	online	Knowledge Based MOEA	Rules

source of knowledge is naturally and directly from the definition and formulation of specific multi-objective problem, such as monotonicity analysis [114]. It is a pre-optimization technique for investigation of important properties among decision variables and the optimal solutions when there are monotonic objective functions or constraint functions. Such methods [116][151] try to capture and utilize the mathematical characteristics of functions to model the original problem to get some insights for the understanding of MOP. However, they are often restricted by strong conditions and not popularly used. On the contrary, the mainstream of knowledge discovery from multi-objective optimization employs an extra analysis of the Pareto optimality after optimization, although the dominated solutions can also be of high value [26]. Deb et al. first created the concept of *innovization* (innovation through optimization) aimed to unveil innovative design principles by means of multiple conflicting objectives [38][49]. Then they proposed a post-optimization analysis framework for automated discovery of vital knowledge from Pareto optimal solutions [9] and successfully demonstrated the applicability of this framework with applications such as truss structure design [10][11]. They finally differentiate the knowledge to higher levels and lower levels [12]. In their methodology, the knowledge denotes hidden problem structure characteristics, such as the correlations between variables and objectives, sensitivity to variables or constraints and so on, and is formulated as polynomials. Then they exploited and optimized these polynomial relationships between variables and objective functions with another evolutionary process after optimization to original MOP under the belief that the optimal solutions satisfying strong relationships will cluster together and the dominating cluster reveals the design principle. However, polynomial relationships are not general patterns and the application seems to be restricted with one or two well-designed examples. The methodology has not been validated with systematic experimentation, although the performance on a simplified and modified ZDT1 benchmark was reported by Gaur and Deb [62].

Beside the mathematical form of knowledge, many other forms are also of interest in practice. Association rules are targeted for extraction from the Pareto front by combining optimization and data mining techniques [132]. Clusters in both decision space and the non-dominated front are partitioned for a truss bridge design problem [142].

Not only does the form of knowledge varies, but also many techniques are involved for different purposes regarding this topic. For instance, self-organizing maps (SOMs) [85] are often utilized when grouping data. It is used to visualize tradeoffs of Pareto solutions for data mining in order to find clusters with high correlations [110], to identify the variable of greatest impact, such as in aircraft wing design [25][51]. Paper [144] clustered optimized solutions hierarchically through the construction of dendrograms. Isometrics feature mapping is used to extract design principles in a hybrid rocket design problem [89]. The rough set theory is employed for the knowledge discovery purposes in [133]. A new multi-objective genetic programming is also proposed for multi-objective design exploration [140].

To be summarized, all the approaches above tend to integrate data mining and machine learning techniques, such as clustering, into the optimization process to obtain interesting knowledge. They have focused on knowledge extraction from optimized solutions, or MOPs themselves in advance. The implementations are highly dependent on the specific problems they are facing.

2.7 Summary

In this chapter, we first briefly reviewed the viable designs of MOEAs, the current test instances for benchmarking MOEA performance and the metrics in use. We can see that the current theory to supervise the construction of algorithms, test problems and metrics has primarily focused on the features of objective space. The current research on multi-objective optimization mainly deal with location and distribution of the optimized objective vectors, but seldom consider factors such as the distribution patterns, implementation viability of the optimized parameters and so on. There are only a few attempts in the literature with decision space features regarding solution diversity in mind. Overall, decision space, compared to objective space, doesn't attract much attention. Different from traditional optimization techniques, evolutionary computation is often criticized as a black-box optimization technique working on a black-box problem without any insight of the MOP provided. The investigation over decision space, not only the objective space, can make the causative side of the problem clearer hence facilitate the understanding of resulting optimal front. The parameter space, instead of the objective space, is where the decision makers and designers can manipulate. Hence, we suggest a transition from objective space to decision space.

Second, there is great effort in the literature dedicated to extract interesting information from both the objective space and the less explored decision space to support final decision making. They vary in knowledge representation, which can be formulated as self-organizing maps, or association rules and so on for better visualization and classification purposes. The knowledge extraction process can be performed either before evolutionary optimization process starts through monotonicity analysis on objective functions, or after optimization process to extract knowledge automatically using machine learning techniques from the non-dominated set. Many real engineering design problems are also discussed. However, the current approaches are usually criticized by introducing extra process either pre-optimization or post-optimization to obtain knowledge and the consequent high computational complexity. These methods are also restrictive on certain applications, not general knowledge extraction methodology for multi-objective environment.

Overall, there are many aspects to consider for the design online knowledge-based evolutionary multi-objective optimization approach. Apparently, based on the review of background and related research, we need new algorithm designs, benchmarks and performance metrics to serve this purpose. In the next chapters, we will elaborate the designs over both evaluation tools and the algorithm and analyse the performance comprehensively.

Chapter 3

Evaluation Criteria for Multi-objective Optimization

This chapter establishes the performance evaluation criteria for multi-objective evolutionary algorithms with decision space analysis in mind. First, we attempt to design test problems which increase the distribution complexity of optimal solutions in decision space to verify the capability of algorithms to capture the Pareto set. Specifically, the distribution that Pareto sets are sparsely clustered is of interests. The distribution clusters of optimal solutions in the decision space allow the generation of simulate optimal design *patterns*, and hence improve the development for knowledge-based (rule-represented) multi-objective optimization algorithms. Second, in order to measure the corresponding performance of MOEAs, a novel metric is proposed aiming at measuring the performance based on the coverage of Pareto sets in the decision space. Three leading MOEAs are then evaluated on the proposed test problem and a comparison of their performance in both the design and objective space is presented based on existing and our proposed metrics.

This chapter is organized as follows: Section 3.1 introduces the current benchmarks and metrics simply to motivate the design of new evaluation tools. Section 3.2 presents our proposed multi-objective test problem with pattern-based Pareto sets. Section 3.3 describe the proposed convex-hull based metric for measuring the decision space performance. The experimental setup and results are discussed in Sections 3.4 before the concluding of the chapter in Section 3.5.

3.1 Introduction

In the development of MOEAs, evaluation methodology has played an important role, involving test problem and performance metric design. Since the optimal solutions for most real world MOPs are not known a-priori, test problems are designed with known optimal solutions to guide development and evaluation of MOEAs. Numerous test problems have been proposed in the literature over the years and some of them, including ZDT [169], DTLZ [50], and WFG [69], have become the de-facto benchmarks in MOEA research. The main motivation behind the design of most benchmarks is to represent the various challenges faced in real optimization problems at some level of abstraction. Hence they provide parameterized frameworks to generate specific test problem instances with varying degrees of complexities in terms of number of decision and objective variables, shape and configuration of optimal trade-off surface (Pareto front), modality of search space, and so on. Subsequently, the MOEAs are tested for their performance in these problems based on several factors such as convergence to Pareto optimal front, speed of convergence, diversity of solutions in terms of coverage of Pareto front, etc. A number of metrics have been proposed in the literature to this effect including generational distance metric (GD), inverse generational distance metric (IGD), etc.

The current test problems and metrics help benchmark the objective space performance and substantially facilitate the design of MOEAs. However, in comparison to MOEAs performance evaluation in objective space, research in performance evaluation of MOEAs in the decision space has not attracted as much attention in the literature. While evaluating MOEAs with respect to their performance in objective space is necessary, performance analysis in decision space can have important implications for the EMO research field. Some motivations for such an analysis include understanding the diversity of solutions in the decision space, designing robust optimizers, design of problems which are closer in characteristics to the real-world optimization problem and knowledge-based optimization.

Overall, in order to do a comprehensive performance evaluation covering not only the objective vectors, but the decision variables for the proposed algorithm, test suites and performance metrics that reflect both the decision space and objective space are required.

3.2 Hyper-rectangular Pareto Set Test Function

This section introduces our proposed test function, which we refer as Hyper-rectangular Pareto Sets (HPS). Our goal in this work is to design test problems where Pareto sets are sparsely clustered in the decision space. Specifically, we are interested in maintaining hyper-rectangular clusters. Such patterns are most suitable to benchmark rule-based MOEAs as they allow specifying the precise number of rules to be learnt by such algorithms. However, same design principles can be adopted to generate test problems with other shaped clusters such as hyper-spheres. The proposed HPS is bi-objective but scalable in the decision space and allows investigation of solution diversity by underlying MOEAs.

In designing our test problem we observed the following general principles suggested in the literature [169, 50, 69]:

- Test function should be composed of several functional units which are easy to understand and manipulate. In general, at least two parts are required in a test function. The first one determines the characteristics of Pareto optimum and the second determines its location.
- Test problems should support customization without much expert knowledge. A variety of example functions should be provided and the number of control parameters should be minimized.

Keeping in view the principles above, HPS construction involves two main components or functions *Gamma* (Γ) and *Delta* (Δ), as shown in Equation 3.1, where the decision vector includes n variables $X = (x_1, \cdots, x_k, x_{k+1}, \cdots, x_n)$ and

Algorithm 1: The Construction of HPS Test Problems

Intput : The number of dimensions (decision variables) in the test problem, n.
The number of dimensions to be partitioned (or define Pareto set clusters), k.

1 Divide k-dimensional space into a given number of partitions *s.t.* $\prod_{i=1}^{k} d_i$, where d_i is the number of divisions for each of the k dimensions;
2 Assign a rank value r (front level) for each cell in the partitioned area based on either a pre-define scheme or uniform random assignment;
3 Define mapping between the partitioned decision space and the objective space using cell ranks and inversely related Γ functions defined over the k variables;
4 Define mapping between the remaining decision variables and the objective space using the Δ function;
5 Define the overall mapping and final objective functions as the product of both Γ and Δ functions;

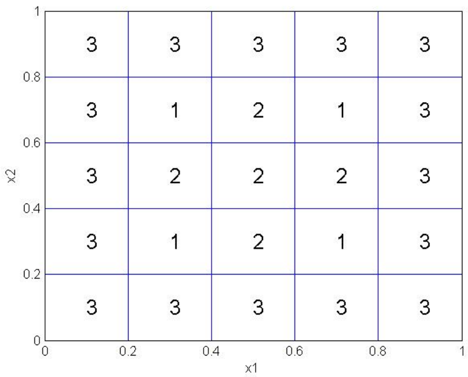
$x_i \in [0, 1], \ \forall i \in \{1, 2, \ldots, n\}$.

$$\begin{aligned} f_1(X) &= \Gamma_1(x_1, x_2, \cdots, x_k) \times \Delta(x_{k+1}, \cdots, x_n) \\ f_2(X) &= \Gamma_2(x_1, x_2, \cdots, x_k) \times \Delta(x_{k+1}, \cdots, x_n) \end{aligned} \tag{3.1}$$

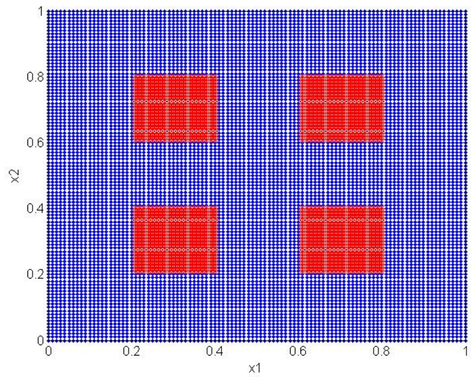
Subject to $X \in \Omega$

The overall procedure to generate an HPS test function instance involves discretising part of an $n-dimensional$ decision space and creating a mapping between this discretised space and the objective space using inversely related bi-objective Γ functions defined over the decision variables used in discretisation and using the Δ function defined over the remaining variables, to create a continuous mapping between the decision variables and the objective space. The detailed procedure is given in Algorithm 1.

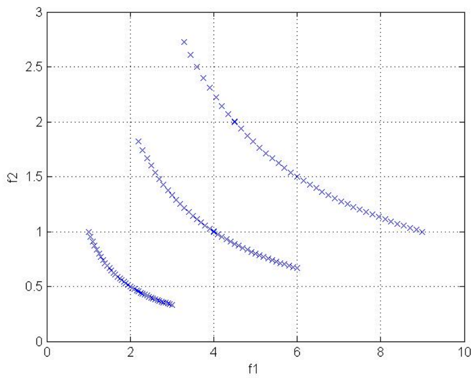
In effect, the Γ component serves to control the geometry of Pareto fronts in objective space and their mapping to the parts of the decision space or the Pareto set clusters defined by hyper-rectangles. The function is parameterized to allow defining local and global optimal fronts using a rank assignment procedure discussed later in Section 3.2.1. The Δ component, on the other hand, is used to configure and control



(a) Partition and Ranking of Cells



(b) Optimal Patterns in Red



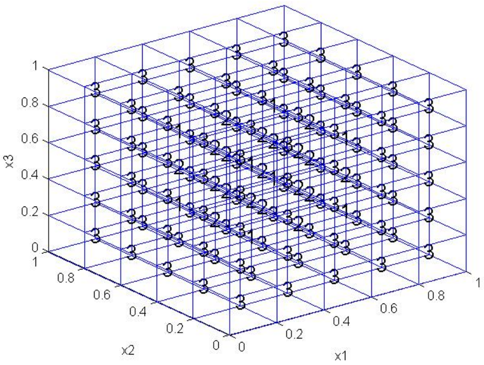
(c) Objective Space of Γ Functions

Figure 3.1: Examples of Partitions, Cell Rank Settings, and Objective Space for Γ Functions When $k = 2$

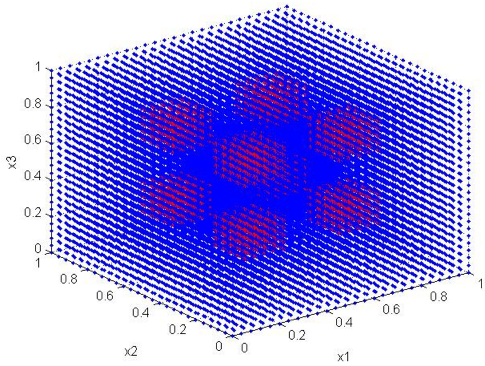
complexity of objective functions with challenges such as multi-modality, variable dependency and so forth. It is discussed in Section 3.2.2.

3.2.1 The Discretisation and Pareto Set Definition Component

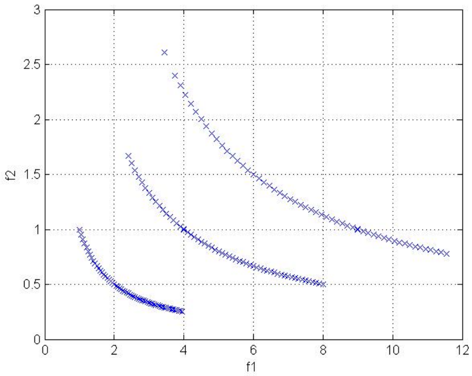
This component corresponds to the first three steps listed in Algorithm 1 and involves discretisation of the decision space and mapping the resulting rectangular or hyper-rectangular cells into ranked Pareto fronts in a bi-objective space using the Γ functions. The discretisation is carried out as following: suppose the first k of n dimensions are to be discretised. The next step is to determine the number and length of divisions for each of the k dimensions.



(a) Partition and Ranking of Cells



(b) Optimal Patterns in Red



(c) Objective Space of Γ Functions

Figure 3.2: Examples of Partitions, Cell Rank Settings, and Objective Space for Γ Functions When $k = 3$

Let the number of divisions for each of the k dimensions is denoted by the vector $d_1, d_2, \ldots, d_k$ and let the lengths of d_i divisions for the i^{th} dimension is given by $l_{i1}, l_{i2}, \ldots, l_{id_i}$ s.t $\sum_{j=1}^{d_i} l_{ij} = 1$. In the simplest division scheme each of the k dimensions can be divided into d divisions of equal length. For the test instances generated for experiments reported in this chapter, we used $d_i = d = 5$ and $l_{ij} = 0.2$ that divides each dimension into five equal partitions.

The next step is to assign a rank to each hyper-rectangle or cell in the partitioned space, using either a predefined scheme or uniform randomly, where each rank would correspond to a front level defined by non-dominated sorting in a bi-objective space. Finally, each of the ranked cells in the discretised decision space is mapped to the objective space using the Γ functions given in Equation 3.2, where

$\varsigma_{\overrightarrow{\lambda}}$, $\overrightarrow{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_k)$ represents the rank assigned to the cell bounded by the lower ($x^l_{\lambda_i}$) and upper bounds ($x^u_{\lambda_i}$) with $i \in \{1, 2, \ldots, k\}$ and λ_i with $\lambda_i \in \{1, 2, \ldots, d_i\}$ representing the corresponding division in each of the k dimensions.

$$\begin{aligned} \Gamma_1 &= \varsigma_{\overrightarrow{\lambda}} \times \left(\sum_{i=1}^{k} \frac{x_i - x^l_{\lambda_i}}{x^u_{\lambda_i} - x^l_{\lambda_i}} + 1 \right) \\ \Gamma_2 &= \varsigma_{\overrightarrow{\lambda}} \times \frac{1}{\displaystyle\sum_{i=1}^{k} \frac{x_i - x^l_{\lambda_i}}{x^u_{\lambda_i} - x^l_{\lambda_i}} + 1} \end{aligned} \tag{3.2}$$

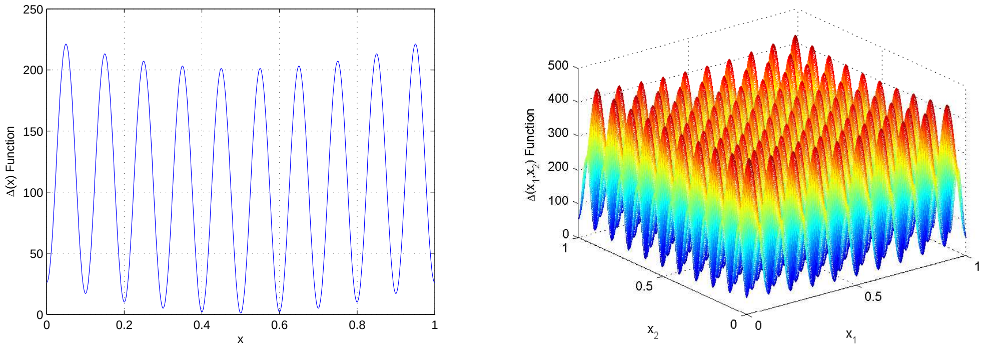
The relationships in Equation 3.2 ensure that the minimum rank value (e.g. 1) is used to define the global front (for minimization tasks). For the test instances generated for experiments reported in this chapter, we only used integer values ($\{1, 2, 3\}$) as ranks for better interpretability and visualization of the test functions, but any other values including real values can be used to create as many fronts as desired.

An example of dicretization, pre-defined assignment of ranks and corresponding fronts in objective space for a two ($k = 2$) and three ($k = 3$) dimensional decision space partitioning is shown in Figures 3.1 and 3.2 respectively. Note that cells with rank 1 are mapped to the global Pareto fronts (considering minimization of objectives) in these cases.

3.2.2 Δ Component and Overall Decision to Objective S-pace Mapping

The procedure used in the discretisation step allows a simple mechanism to define H-PS and their mapping to a bi-objective space. However, a number of other features (see Table 2.3) are generally needed to define a challenging problem that reflects characteristics of the real-world optimization tasks. The Δ component allows integrating such features in the HPS problems as well as re-mapping the decision and objective space association to a continuous domain.

In ZDT [169] & DTLZ [50] problems, a few different g functions are proposed



(a) Δ Function when $n-k=1$ (b) Δ Function when $n-k=2$

Figure 3.3: Examples of Landscapes of Δ Function

to promise separability and modality for various function geometries. For the WFG problems [69] a few different operations are recommended, such as shift and reduction, over decision variables to achieve similar objectives. In this chapter, we adopt the g function from DTLZ1 problem given in Equation 3.3 as the Δ component in generating the HPS instances in this chapter for demonstration. But other functions can also be easily adopted to define HPS test problems.

$$\Delta(x_{k+1},\dots,x_n) = 1+100\times\left[n-k+\sum_{i=k+1}^{n}\Big((x_i-0.5)^2-\cos\big(20\pi(x_i-0.5)\big)\Big)\right] \quad (3.3)$$

Notice that this function has a global optimum at $\forall_i,\ x_i = 0.5$. The landscape for a one and two variables Δ function is illustrated in Figure 3.3. For HPS_1–HPS_5 the number of decision variables in Δ function $(n-k)$ is set to 5, as recommended in DTLZ1. HPS_6 is built with a ten variables Δ function.

The overall mapping from decision to objective space in HPS test problems is finally defined by combining both Γ and Δ functions using the relations shown in Equation 3.2 and Equation 3.3. The Pareto front of this bi-objective optimization is shwon in Equation 3.4.

$$f_1 \times f_2 = 1 \text{ and } f_1 \in [1, k+1] \quad (3.4)$$

3.2.3 Test Problem Instantiation

Table 3.1: Instances of HPS Test Problem with Number of Dimensions for Partition k, Number of Overall Variables n, Number of Optimal Patterns in Decision Space, Shape of the Optimal Pattern, and the Volume of Pareto Sets

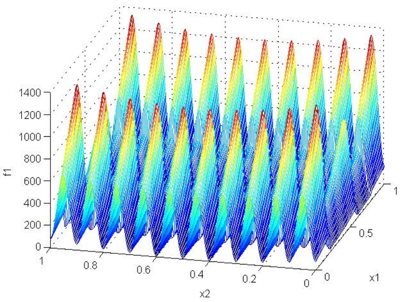
Problems	k	n	# of Patterns	Shape of Patterns	Volume
HPS_1	1	6	2	Line Segment	0.4
HPS_2	2	7	4	Square	0.16
HPS_3	3	8	8	Cube	0.064
HPS_4	4	9	16	4-D Hyper-Cube	0.0256
HPS_5	5	10	32	5-D Hyper-Cube	0.01024
HPS_6	2	12	4	Square	0.16

A number of test instances using the proposed procedure and different parameter settings can be created. Table 3.1 shows six such test instances generated by varying the k and n parameters.

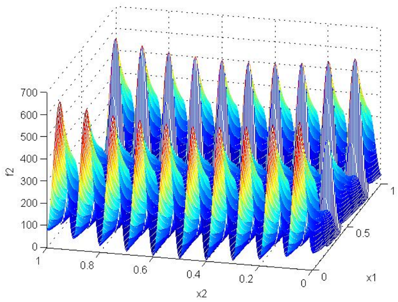
In the simplest case k maybe equal to n, in this case the Δ component does not apply in the construction of objective functions and the resulting test instance will only have discrete distribution patterns. Equation 3.5 shows the objective function representation of the simplest test problem (HPS_0) and Figure 3.4 shows the corresponding visualizations for function landscapes, decision space and objective space.

To increase the level of complexity, n needs to be greater than k. For the Δ component adopted in this chapter, $n - k \geq 5$ provides a mapping with better complexity (notice that this corresponds to the DTLZ1's g function which uses 5 variables). HPS_6 comes with $n - k = 10$ to enhance the search difficulty. Note that the indexing used for our test instances does not necessarily indicate increasing complexity, e.g. HPS_6 may not be the most difficult problem in this set, because the complexity of HPS problems is determined by both k and n parameters.

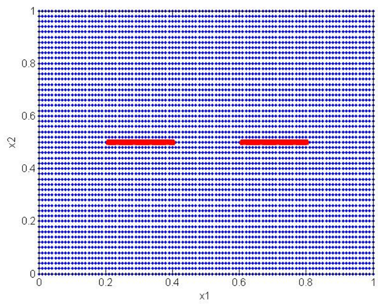
$$
\begin{aligned}
f_1 &= \varsigma_{\lambda_1} \times \left(\frac{x_1 - x^l_{\lambda_1}}{x^u_{\lambda_1} - x^l_{\lambda_1}} + 1 \right) \times (1 + 100 \times (1 + (x_2 - 0.5)^2 - \cos(20\pi(x_2 - 0.5)))) \\
f_2 &= \varsigma_{\lambda_1} \times \frac{1}{\frac{x_1 - x^l_{\lambda_1}}{x^u_{\lambda_1} - x^l_{\lambda_1}} + 1} \times (1 + 100 \times (1 + (x_2 - 0.5)^2 - \cos(20\pi(x_2 - 0.5))))
\end{aligned} \tag{3.5}
$$



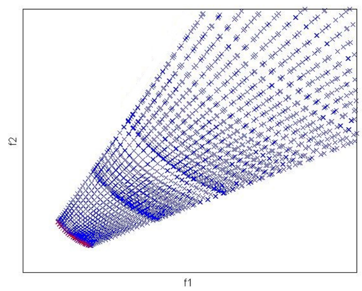
(a) Landscape of f_1 of HPS_0



(b) Landscape of f_2 of HPS_0



(c) Decision Space of HPS_0 with the Optimal Areas in Red



(d) Objective Space of HPS_0 with the Pareto Front in Red

Figure 3.4: the Landscapes of Test Functions, the Decision Space and Objective Space of HPS_0

3.3 A Metric to Measure Pareto Set Coverage

The use of metrics to evaluate MOEA performance has been discussed in detail in Section 2.5. As noted, given the scarcity of research in decision space performance analysis of MOEAs, only a few metrics have been proposed for measuring the decision space performance [126, 79, 143, 145]. The metric proposed in this chapter, which we refer as Pareto Set Volume (PSV) metric, focuses on measuring the MOEA decision space performance in terms of *the percentage of covered volume by the solution set with respect to the total volume occupied by the Pareto sets.* Here the covered volume for a given MOEA corresponds to the volume or area bounded by the convex hull (or convex envelope) of all optimal solutions in decision space over the overall areas of optimal region of the MOP.

Convex hull is a concept from computational geometry. For a set X of points in Euclidean space, the convex hull or convex envelope is mathematically the smallest convex set that contains X. For a finite point set X on a plane, its convex hull can be visualized as the convex polygon shown in Figure 3.5.

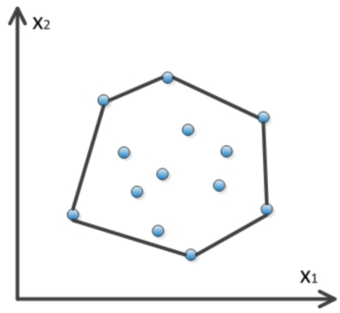


Figure 3.5: Convex Hull in Two Dimensional Space

The convex hull can be explained in two ways. The first is to define it as the intersection of all convex sets containing X. Alternatively, the convex hull can be regarded as the set of all convex combinations of points in X. A convex combination of points $x_1, x_2, \ldots, x_n$ is a point of the form $\Sigma_{i=1}^{n} \alpha_i \times x_i$ with all $\alpha_i \geq 0$ and $\Sigma_{i=1}^{n} \alpha_i = 1$. Based on this, the formal expression of convex hull is as following: Given a finite point set $X = \{x_1, x_2, \ldots, x_n\}$, its convex hull is the set

$$\left\{ \sum_{i=1}^{n} \alpha_i x_i \;\middle|\; (\forall \alpha_i \geq 0) \wedge \sum_{i=1}^{n} \alpha_i = 1 \right\}$$

In computational geometry, finding convex hull of a finite set of points in Euclidean spaces is a well-researched problem. Numerous algorithms have been proposed for computing the convex hull of a finite set of points, with various computational complexities including Package Wrapping Algorithm, Graham Scan Algorithm, Quickhull Algorithm, Divide and Conquer Algorithm, Monotone Chain Algorithm, the Ultimate Planar Convex Hull Algorithm and so on. In this work we used the Quickhull Algorithm [13] to compute the convex hull in 2D, 3D, and higher dimensions[1].

Algorithm 2: The Computation of PSV Metric

Intput : A MOP with n_{opti} optimal regions in decision space;
A set of obtained solutions with size n_s;

Calculate the total area of predefined optimal regions $S = \sum_{i=1}^{n_{opti}} s_i$;
for $i = 1 : n_s$ **do**
    for $j = 1 : n_{opti}$ **do**
        if *solution* $i \in$ *optimal region* j **then**
            solution $i \rightarrow$ group j;
            $cnt_j + +$;
        end
    end
end
for $i = 1 : n_{opti}$ **do**
    if $cnt_i > 0$ **then**
        Calculate the area of the convex hull of solutions in group i, denoted as c_i;
    else
        $c_i = 0$;
    end
end
PSV$= \sum_{i=1}^{n_{opti}} c_i/S$;

It may be argued that a volume metric will be more suited to a rule-based MOEA and hence may give such algorithms an advantage over the general class of point based MOEAs, however, the convex hull method to compute the covered volume ensures against such biases.

The procedure to compute PSV is shown in Algorithm 2 and is pictorially demonstrated using the example in Figure 3.6. Given four Pareto sets represented by the rectangles s_1, s_2, s_3, s_4 in the decision space and a solution set represented by the blue asterisks, PSV is computed as $\sum_{i=1}^{4} C_i/S_i$. Where C_i and S_i represent the areas for the i^{th} convex hull c_i and rectangle s_i respectively. Any solutions that do

[1] For more information, see http://www.qhull.org/.

not fall within the bounds of any of the optimal areas, such as those represented by points A and B, are excluded from the computation of convex hull.

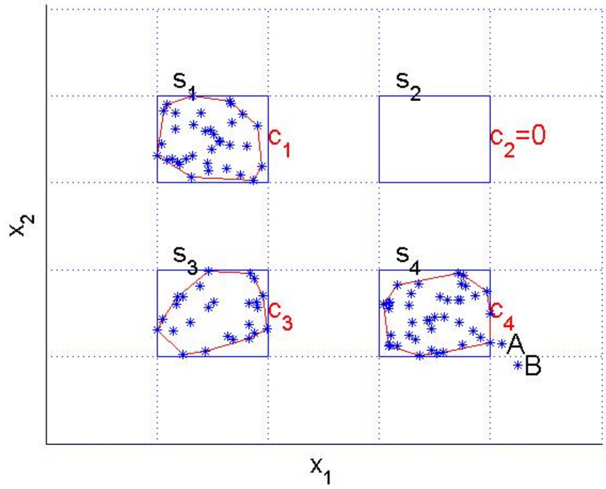


Figure 3.6: The Calculation of PSV Metric in 2D Decision Space with Optimal Areas s_1, s_2, s_3, s_4 respectively in blue and Convex Hull Volumes as c_1, c_2, c_3, c_4 in red

3.4 Experiments

3.4.1 Experimental Setup

This section presents the results of the experiments conducted to test the performance of three leading MOEAs, including NSGA2, NSGA3 and MOEA/D, introduced in Chapter 2 Section 2.2, with different instances, listed in Table 3.1, of our proposed HPS test function.

Three metrics, including the traditional Inverted Generational Distance (IGD) described in Chapter 2 Section 2.5.3, Solow-Polasky Diversity (SPD) described in Chapter 2 Section 2.5.4 and our proposed PSV metric introduced in Section 3.3, are used to compare the performance of these algorithms in terms of their ability to maintain solution diversity and learn optimal distribution patterns in the decision space as well as in the objective space.

For all three algorithms same genetic operators, i.e. Simulated Binary Crossover

(SBX) with default distribution index 20 and polynomial mutation with distribution index 10, are adopted. The crossover and mutation rates are set to 1 and $1/n$ respectively, where n is the number of variables in the decision space. All experiments are repeated with five different population size settings, including 100, 500, 1000, 5000 and 10000, for sensitivity analysis. The maximum generation number is set to 1000 for all experiments and the results are averaged over 10 independent runs.

The MOEA/D version with Tchebycheff approach [162] is used here. Both NSGA3 and MOEA/D use the same set of vectors as their reference points and weight vectors respectively. Since the test problems have only 2 objectives, for N points the vectors are generated are as follows:

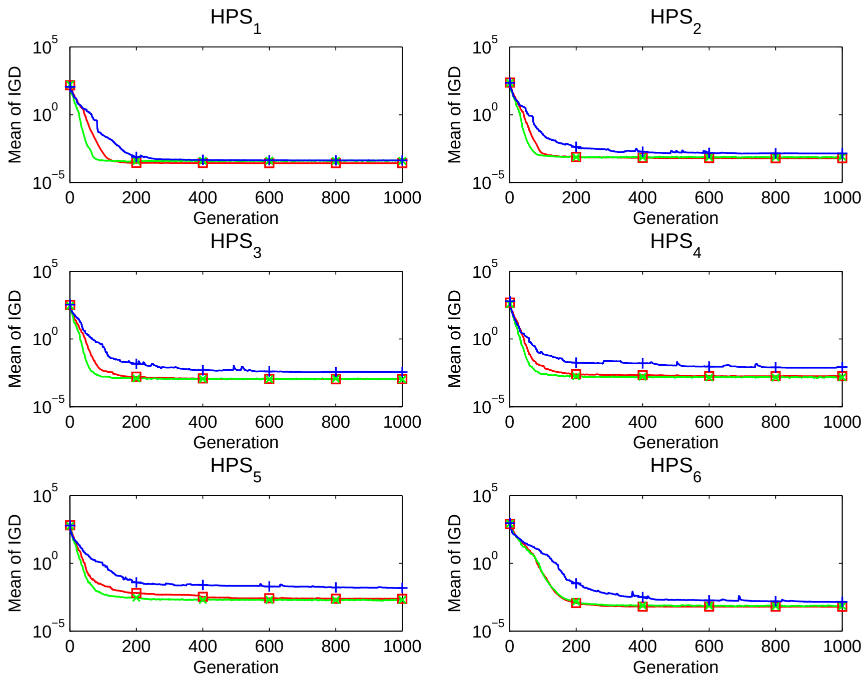
$$\left\{ \left(\frac{i}{N-1}, \frac{N-1-i}{N-1} \right) \middle| i = 0, \ldots, N-1 \right\}$$

3.4.2 Results and Discussion

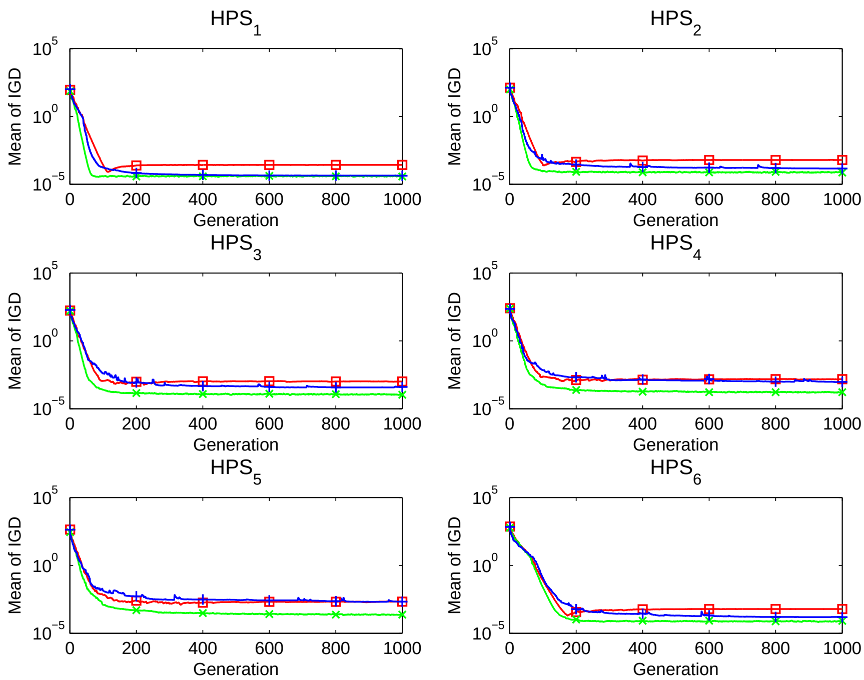
The experimental results are organized in Tables 3.2– 3.4 and Figures 3.7– 3.12. The tables summarize the average results for the three algorithms on six different problem instances HPS_1–HPS_6 (organized in multirows) and five different population sizes (organized in columns) at the final generation based on the three metrics listed above respectively. Figures 3.8, 3.10, and 3.12 complement the tabular results for a easier interpretation. Figures 3.7, 3.9, and 3.11 present the performance of each algorithm over time, or the number of generations, based on the three metrics. The results for each of the six problem instances are organized in subfigures and results are presented for only two population sizes. For all figures, the legend is consistent, NSGA3 performance is represented by a red line, NSGA2 in green and MOEA/D in blue.

3.4.2.1 Objective Space Performance Comparison using IGD

IGD evaluates both convergence and diversity in objective space. The lower IGD values are considered better with a zero value representing ideal solutions. From

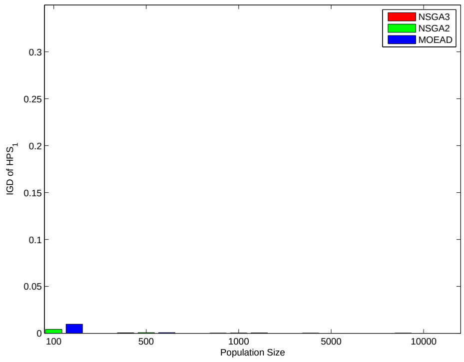


(a) Average IGD Over Time When Population Size is 1000

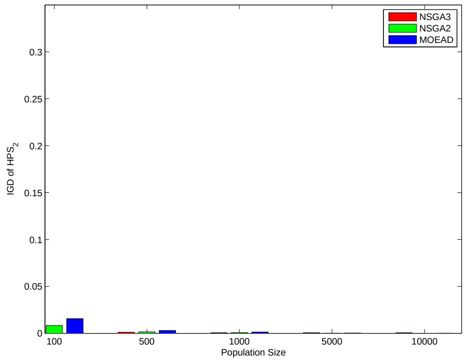


(b) Average IGD Over Time When Population Size is 10000

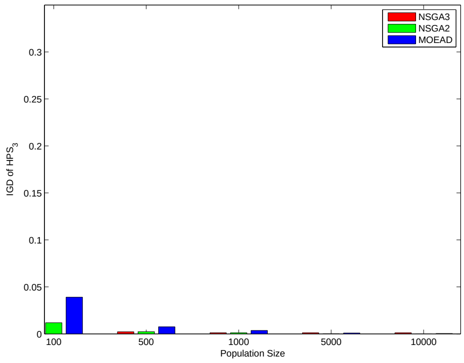
Figure 3.7: Average IGD Over Time Using Logarithmic Y-Axis



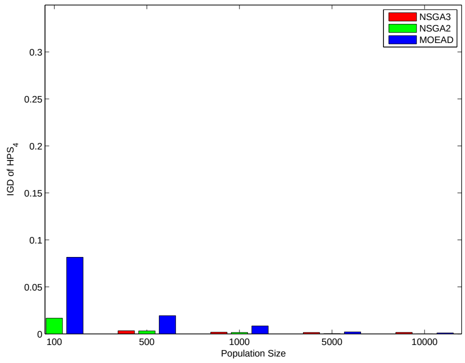
(a) HPS_1



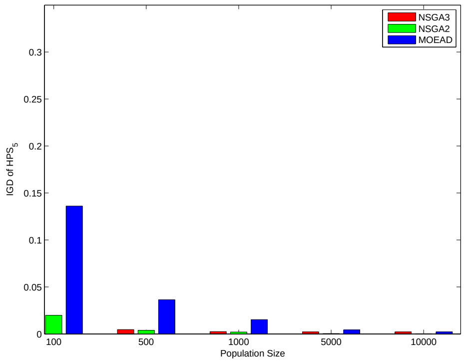
(b) HPS_2



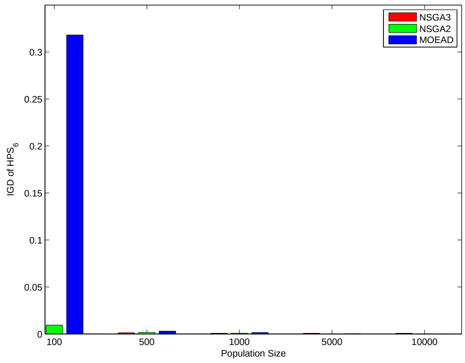
(c) HPS_3



(d) HPS_4



(e) HPS_5



(f) HPS_6

Figure 3.8: Average IGD values achieved by the three algorithms in each of the six problems at the last generation for different population sizes.

Table 3.2: Average IGD at Last Generation

Population Size		100	500	1000	5000	10000
	NSGA3	3.065e-03	5.328e-04	2.640e-04	2.631e-04	2.637e-04
HPS_1	NSGA2	4.191e-03	6.834e-04	3.939e-04	7.817e-05	3.764e-05
	MOEA/D	9.757e-03	7.121e-04	4.209e-04	7.227e-05	4.218e-05
	NSGA3	6.706e-03	1.173e-03	6.124e-04	6.145e-04	6.093e-04
HPS_2	NSGA2	8.396e-03	1.491e-03	7.846e-04	1.396e-04	7.278e-05
	MOEA/D	1.568e-02	3.013e-03	1.376e-03	2.932e-04	1.420e-04
	NSGA3	1.060e-02	2.107e-03	1.099e-03	1.030e-03	1.009e-03
HPS_3	NSGA2	1.178e-02	2.253e-03	1.043e-03	2.215e-04	1.078e-04
	MOEA/D	3.901e-02	7.398e-03	3.587e-03	8.341e-04	4.054e-04
	NSGA3	1.630e-02	3.236e-03	1.807e-03	1.473e-03	1.501e-03
HPS_4	NSGA2	1.665e-02	3.130e-03	1.489e-03	3.095e-04	1.653e-04
	MOEA/D	8.153e-02	1.933e-02	8.371e-03	1.996e-03	9.350e-04
	NSGA3	2.266e-02	4.601e-03	2.426e-03	2.144e-03	2.123e-03
HPS_5	NSGA2	1.980e-02	3.891e-03	2.030e-03	4.038e-04	2.315e-04
	MOEA/D	1.361e-01	3.639e-02	1.517e-02	4.249e-03	2.127e-03
	NSGA3	7.270e-03	1.202e-03	6.293e-04	6.148e-04	6.041e-04
HPS_6	NSGA2	9.369e-03	1.519e-03	6.990e-04	1.485e-04	8.128e-05
	MOEA/D	3.181e-01	2.964e-03	1.420e-03	3.070e-04	1.551e-04

Table 3.2 and Figure 3.8 we can see that generally all algorithms show improved performance in terms of IGD values with increasing population sizes. When comparing the performance of the three algorithms in terms of this metric, it can be observed that MOEA/D almost always performs relatively poor in all problems when the population size is small. Whereas there is no clear winner between NSGA2 and NSGA3 as in some problems one algorithm performs better than other and vice versa. All algorithms perform almost equally with higher population sizes. For lower population sizes (100 in the figures) we can see that the algorithm performance degrades, as they achieve higher IGD values, when we move from simpler to more difficult problems HPS_1 to HPS_6. This indicates that IGD values are probably a good representative of the problem difficulty in this context. However note that, since the IGD calculations are dependent upon the geometry of the true optimal and the obtained sets [75], this metric may not be considered true representative of the problem difficulty.

Figure 3.7 shows the IGD curves over time or number of generations. It can be seen that all algorithms converge nicely as the number of generations increase. This happens around 200 to 300 generations when the population size is 100 and around 50 to 100 generations when the population size is 10000. It is interesting to note that although all algorithms achieve lower IGD values in HPS_6 but the convergence is the slowest of all problems. This is most likely because of the number of more local optima in this problem than any other.

A small fluctuation in IGD values over time can also be observed. This is probably because of the artefact of IGD computation which is not strictly monotonic with the quality of solution sets, as noted by [75].

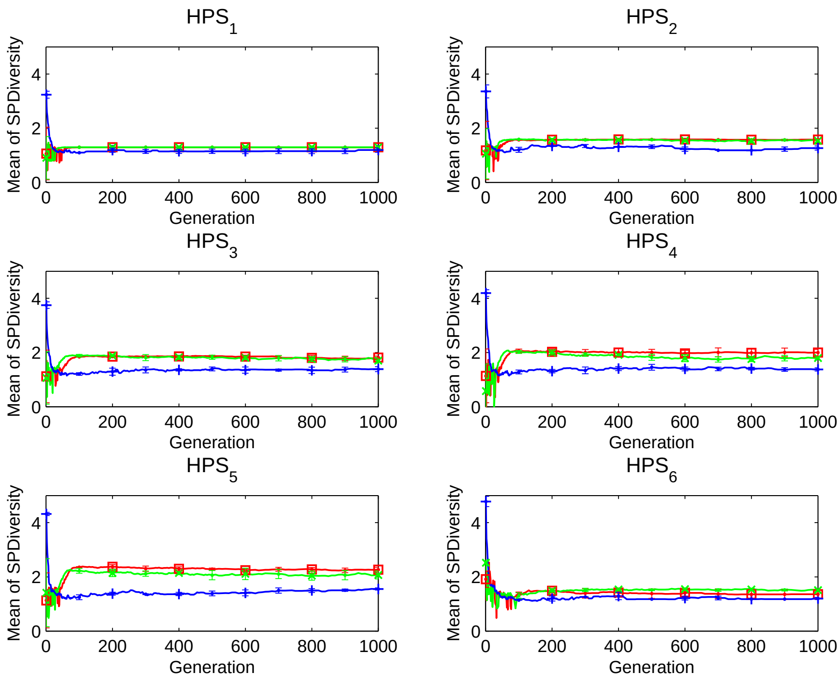
To sum up, IGD is popular in literature and has wide application in evolutionary computation for performance evaluation. We also used IGD here to compare the performance of algorithms regarding new test problems. The information IGD provides relates only to objective space. Next sections investigate decision space performance as well.

3.4.2.2 Decision Space Performance Comparison using SPD

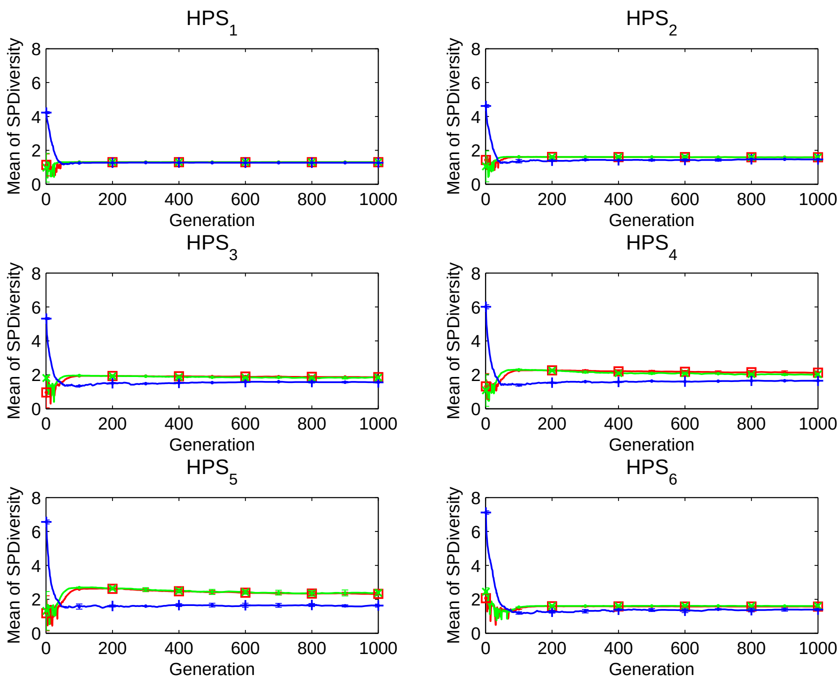
Table 3.3: Average SPD at Last Generation

Population Size		100	500	1000	5000	10000
HPS_1	NSGA3	1.288e+00	1.297e+00	1.299e+00	1.298e+00	1.297e+00
	NSGA2	1.298e+00	1.299e+00	1.299e+00	1.300e+00	1.300e+00
	MOEA/D	1.100e+00	1.218e+00	1.195e+00	1.254e+00	1.267e+00
HPS_2	NSGA3	1.309e+00	1.500e+00	1.579e+00	1.592e+00	1.591e+00
	NSGA2	1.509e+00	1.572e+00	1.539e+00	1.596e+00	1.594e+00
	MOEA/D	1.276e+00	1.286e+00	1.267e+00	1.432e+00	1.462e+00
HPS_3	NSGA3	1.332e+00	1.631e+00	1.813e+00	1.792e+00	1.860e+00
	NSGA2	1.430e+00	1.757e+00	1.704e+00	1.792e+00	1.837e+00
	MOEA/D	1.312e+00	1.301e+00	1.388e+00	1.494e+00	1.557e+00
HPS_4	NSGA3	1.435e+00	1.712e+00	1.998e+00	2.050e+00	2.124e+00
	NSGA2	1.513e+00	1.914e+00	1.798e+00	1.969e+00	2.033e+00
	MOEA/D	1.261e+00	1.364e+00	1.378e+00	1.558e+00	1.649e+00
HPS_5	NSGA3	1.600e+00	1.900e+00	2.265e+00	2.153e+00	2.322e+00
	NSGA2	1.461e+00	1.919e+00	2.071e+00	2.253e+00	2.385e+00
	MOEA/D	1.325e+00	1.476e+00	1.554e+00	1.579e+00	1.631e+00
$HPS-6$	NSGA3	1.244e+00	1.479e+00	1.359e+00	1.572e+00	1.576e+00
	NSGA2	1.401e+00	1.558e+00	1.509e+00	1.589e+00	1.610e+00
	MOEA/D	1.221e+00	1.203e+00	1.203e+00	1.342e+00	1.392e+00

The SPD, as introduced in Chapter 2 Section 2.5.4, is dedicated for the evalua-

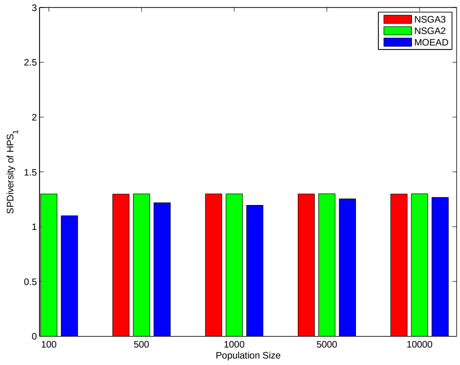


(a) Average SPDiveristy Over Time When Population Size is 1000

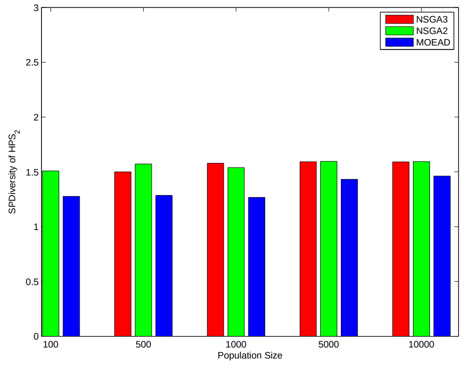


(b) Average SPD Over Time When Population Size is 10000

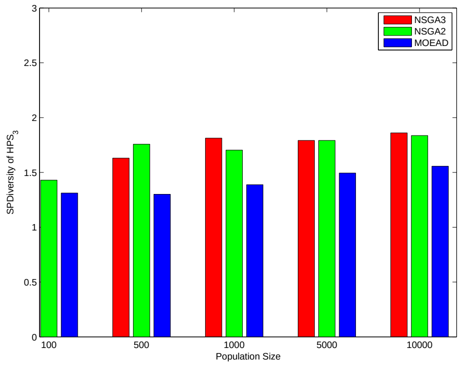
Figure 3.9: Average SPD Over Time with Error Bars



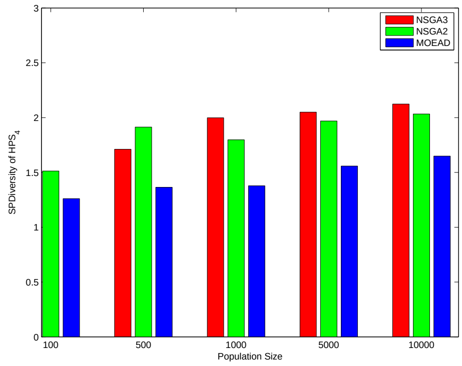
(a) HPS_1



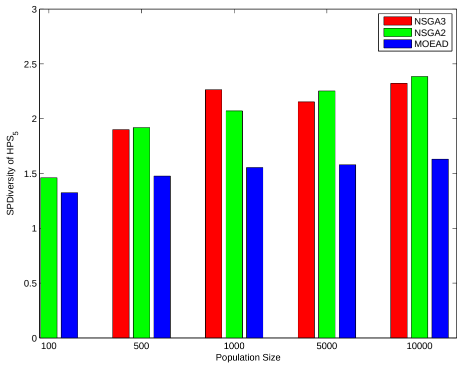
(b) HPS_2



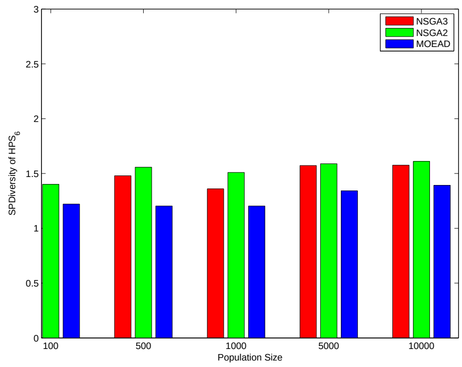
(c) HPS_3



(d) HPS_4



(e) HPS_5

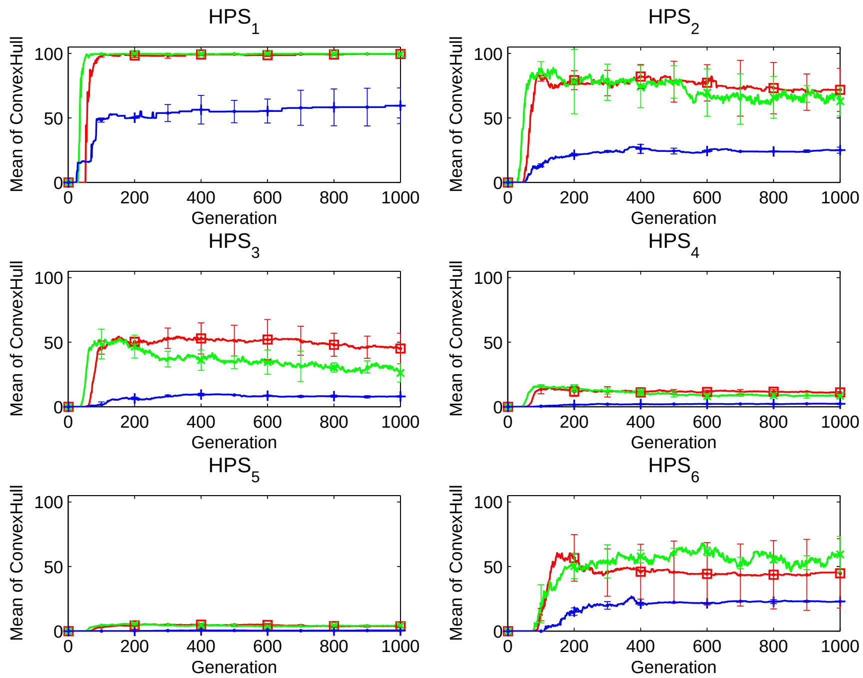


(f) HPS_6

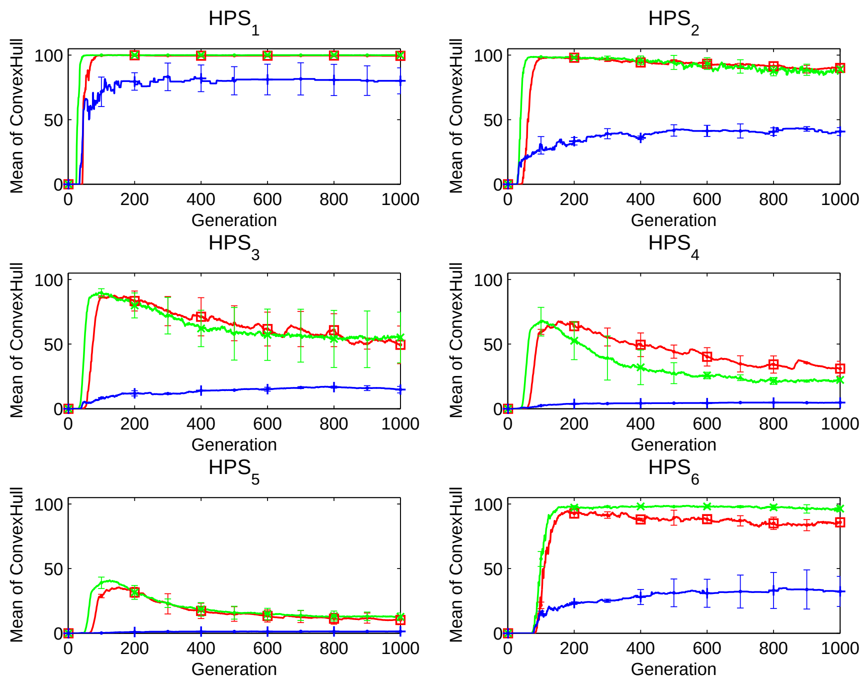
Figure 3.10: Average SPD Values achieved by the three algorithms in each of the six problems at the last generation for different population sizes.

tion of solution diversity in decision space. Usually, higher metric value corresponds to better decision space diversity. From Table 3.3 and Figure 3.10, generally the performance of all three algorithms improves with the increase in population size. When comparing between the three algorithms, the results are similar to the IGD metric and it can be seen that MOEA/D performs the poorest of the three algorithms, even with increasing population sizes. NSGA2 and NSGA3 perform equivalently in most cases based on this metric, with NSGA3 showing slightly better performance overall. The most interesting and counter-intuitive observation is that almost all algorithms achieve higher SPD scores going from problem HPS_1 to HPS_5. This is in contrast with IGD values which generally became worse in this direction for all algorithms. However, this makes sense because SPD is calculated based on pair-wise distances between solutions and the values tend to grow when the solution set is sparse. In our test problem set, the problem sparsity increases from HPS_1 to HPS_5 due to number of dimensions and hence the sparsity of Pareto sets also increases leading to higher values for this metric. In summary we can say that SPD, like IGD, is a problem oriented metric. The values only are comparable for the same problem but not across problems.

The change of SPD value over time is shown in Figure 3.9. For all test problems, a similar pattern can be observed where small fluctuations in the values are observed for all algorithms at the start of the evolution but the trends flat out in the end, showing a convergence. Notice that in the beginning we observe the best SPD values, but this is due to the random initialization of the solutions and does not mean that the optimal solutions have been identified. This is another artefact of this measure. Moreover, the quality of solution set in decision space over time is generally not static (e.g. as shown by IGD or our PSV metric in the next section), however as mentioned before SPD metric tends to lose this sensitivity. This can be clearly seen from the levelling of SPD curves over time in Figure 3.9. Nonetheless, SPD is perhaps the main metric available currently to measure decision space performance for EMO.

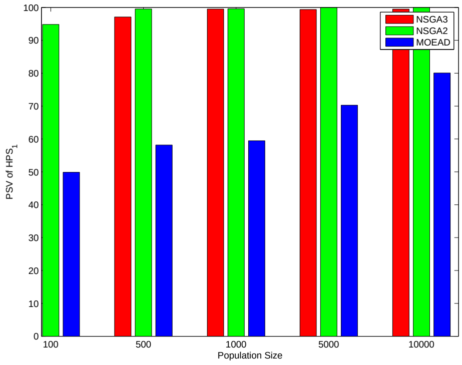


(a) Average PSV When Population Size is 1000

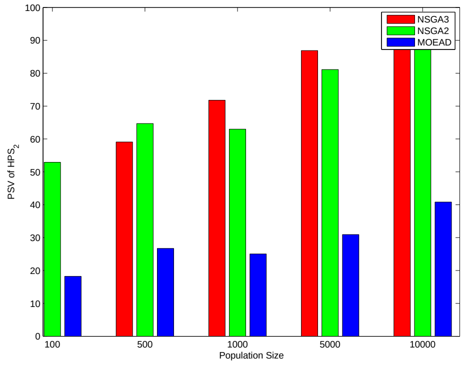


(b) Average PSV When Population Size is 10000

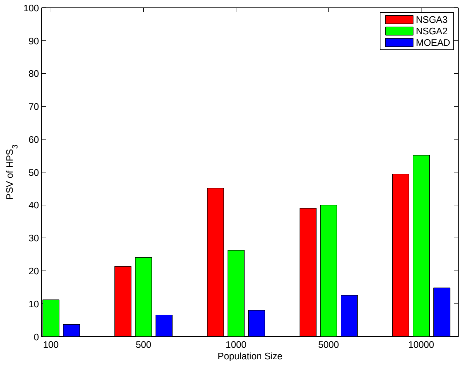
Figure 3.11: Average PSV Over Time With Error Bars



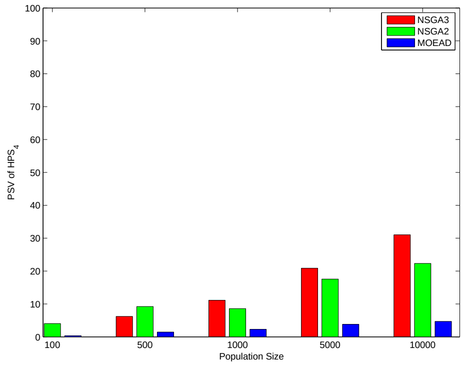
(a) HPS_1



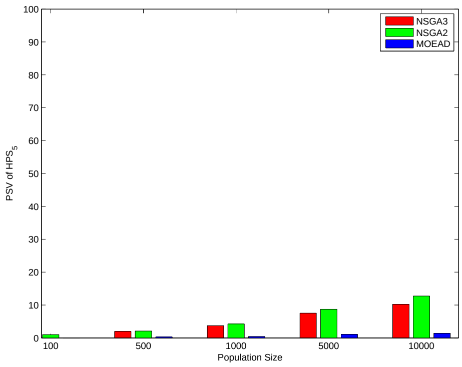
(b) HPS_2



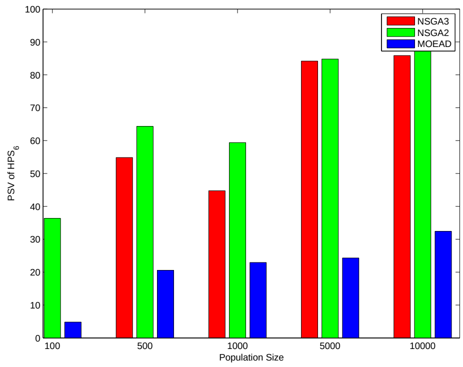
(c) HPS_3



(d) HPS_4



(e) HPS_5



(f) HPS_6

Figure 3.12: Average PSV Values achieved by the three algorithms in each of the six problems at the last generation for different population sizes.

3.4.2.3 Decision Space Performance Comparison using PSV

Table 3.4: Average PSV at Last Generation

Population Size		100	500	1000	5000	10000
HPS_1	NSGA3	9.214e+01	9.711e+01	9.952e+01	9.937e+01	9.949e+01
	NSGA2	9.484e+01	9.953e+01	9.960e+01	9.993e+01	9.996e+01
	MOEA/D	4.989e+01	5.817e+01	5.946e+01	7.028e+01	8.006e+01
HPS_2	NSGA3	3.215e+01	5.908e+01	7.179e+01	8.688e+01	9.002e+01
	NSGA2	5.291e+01	6.470e+01	6.299e+01	8.110e+01	8.921e+01
	MOEA/D	1.823e+01	2.672e+01	2.505e+01	3.093e+01	4.082e+01
HPS_3	NSGA3	9.693e+00	2.134e+01	4.515e+01	3.903e+01	4.943e+01
	NSGA2	1.121e+01	2.401e+01	2.625e+01	3.999e+01	5.516e+01
	MOEA/D	3.702e+00	6.573e+00	8.007e+00	1.259e+01	1.481e+01
HPS_4	NSGA3	2.503e+00	6.213e+00	1.113e+01	2.087e+01	3.102e+01
	NSGA2	4.042e+00	9.213e+00	8.578e+00	1.757e+01	2.232e+01
	MOEA/D	3.389e-01	1.448e+00	2.280e+00	3.821e+00	4.706e+00
HPS_5	NSGA3	4.812e-01	1.974e+00	3.728e+00	7.524e+00	1.024e+01
	NSGA2	1.012e+00	2.095e+00	4.252e+00	8.707e+00	1.274e+01
	MOEA/D	2.743e-02	2.937e-01	4.364e-01	1.092e+00	1.398e+00
HPS_6	NSGA3	2.585e+01	5.484e+01	4.475e+01	8.419e+01	8.585e+01
	NSGA2	3.635e+01	6.431e+01	5.939e+01	8.480e+01	9.647e+01
	MOEA/D	4.812e+00	2.056e+01	2.293e+01	2.431e+01	3.242e+01

The PSV evaluates the percentage of the area of convex hull determined by solutions over the overall area of optimal patterns. Higher value is preferred. If more than one regions are involved, they are treated separately and averaged in the final measurement. The definition of PSV requires that the obtained solutions are close enough to the optimal region. Based on the construction of test problems, we only consider solutions with the first k variables to be in $[0.2, 0.4]$ or $[0.6, 0.8]$ and the rest $n - k$ variables to be in $[0.5 - p, 0.5 + p]$ when calculating the convex hull. Here, p is the precision of obtained solutions and $p = 0.001$ is adopted.

From Table 3.4 and Figure 3.12 we can see that generally all algorithms show improved performance with respect to the population size. It clearly reveals that population size is a key factor to improve performance in decision space for current algorithms. Among the three algorithms adopted in this experimentation, NSGA algorithms perform better against MOEA/D with Tchebycheff approach. This is consistent with the observations in the two metrics above. From the problems perspective, PSV shows that HPS_5 is the most difficult problem for the algorithms to identify the optimal patterns. It can also be observed that the PSV metric is sensitive to the problem complexity as the PSV values drop significantly going from HPS_1 to HPS_5. For HPS_2 and HPS_6, where HPS_6 is more difficult for the number of variables in Δ function to challenge the convergence, the algorithms can achieve equivalent performance when population size is 10000. But for other population size settings, they are better when dealing with HPS_2 rather than HPS_6. This further supports that PSV is more sensitive to problem complexity than SPD.

Figure 3.11 shows the variation of PSV values over time with different population sizes. The PSV usually starts with a 0 value at the start of the runs and gradually improves when the obtained sets get closer the optimal regions.

An interesting observation is that in many cases, the best PSV value may be obtained before the last generation, especially for NSGA algorithms. This means that this metric can be used effectively for a stopping criteria instead of a fixed number of generations.

Overall, these results show that the PSV metric being more sensitive to problem difficulty is a more suitable metric to be used for comparing the performance across different problems, especially when decision space performance analysis is a major concern.

3.5 Summary

MOEAs are popular approaches to deal with MOPs. A number of test problems and metrics exist that support MOEA development as well as benchmark their performance. In comparison to MOEAs performance evaluation in objective space, research in performance evaluation of MOEAs in the decision space is rather scant. Attention to decision space analysis has important implications for the field of EMO in general. Specifically, such a research effort may help designing of additional test problems that are closer to real world MOPs, gaining better understanding of the search process and the rules or patterns that lead to optimal and robust designs, and in the design and evaluation of knowledge-based optimization algorithms.

This chapter complements the existing research in this area. We proposed a mechanism to generate test problems where the solutions belonging to efficient or Pareto frontier are mapped to defined hyper-rectangular patterns in the decision space. A modular design approach is employed which allows generating a number of test problem instances with varying degrees of complexity in terms of number of decision variables, number of optimal patterns and other interesting features. We also proposed a new metric to evaluate the performance of MOEAs in the decision space. The proposed metric relies on computing the ratio of volume covered by the solution set obtained by an MOEA to the total volume occupied by the defined Pareto sets in decision space. Compared to existing test problems and metrics which mainly focus on objective space performance evaluation, the new test problems and metric proposed in this chapter can be used to benchmark the performance of MOEAs from decision space performance perspective.

Experimental results are presented to compare the performance of three leading MOEAs, NSGA2, NSGA3 and MOEA/D using the proposed test problems and metric. The results show that the test problems pose new learning challenges for MOEAs and suggest that new mechanisms are probably needed to improve algorithm performance when dealing with such problems. In future, we aim to focus on exploring such mechanisms and algorithms that may better suit when dealing with such problems.

Chapter 4

Online Knowledge-based Evolutionary Multi-objective Optimization

This chapter introduces the main idea of knowledge-based evolutionary multi-objective optimization (*KB-EMO*). To reiterate, here we refer *knowledge* as *patterns in the design space that lead to Pareto optimal solutions in the objective space*. While the traditional EMO algorithms aim at searching for individual solutions that are Pareto optimal, the aim of KB-EMO is to search for Paretor optimal patterns or areas in the design space. In other words, the aim of KB-EMO is to move from specific to general optimization solutions. This can primarily be done in two ways: 1) by using an MOEA to solve an optimization problem, retrieving the set of most optimal solutions and using a machine learning or statistical technique to find the patterns that contain this set of solutions; 2) by designing an MOEA that search for optimal patterns while solving the optimization problem. These approaches can be classed as post-optimization and online KB-EMO techniques, respectively.

The techniques introduced in this chapter fall under the latter category and hence referred as *online KB-EMO*. The contents of this chapter are organized as follows: A generic framework, to design online knowledge-based multi-objective optimization algorithms (*KB-MOEAs*) that evolve MOP solutions in the form of interpretable rules representing optimal design patterns, is first presented in Section 4.1. A specific instantiation of this framework is then presented, in Section 4.2, that relies on commonly used non-dominated sorting principle for evolving a solution to a MOP in the form of hyper-rectangular or simple if-then rules. The resulting algorithm is referred to as $RB - MOEA_{NS}$. The experimental setup to conduct a thorough investigation of the proposed online KB-MOEA with classic benchmarks and new problems introduced in Chapter 3 is then reported in Section 4.3 and the results are discussed in Section 4.4. Conclusions of this chapter are provided in Section 4.5.

4.1 The Online KB-EMO Framework

The core idea behind the KB-EMO framework is that the evolutionary process is applied on a population of n-dimensional bounding hypervolumes (or rules), where n corresponds to the number of variables or dimensions in the design space. The rule population is then evolved towards Pareto optimal areas instead of evolving towards individual Pareto solutions. The fitness of rules is partly dependent upon the quality of the sampled solutions from the bounded volume of the rule according to a multi-objective criterion, such as non-dominated sorting, and partly on other characteristics, such as, the relative volume of the rules and proportion of good solutions.

An algorithmic description of the proposed online KB-EMO framework is provided in Algorithm 3. The framework requires initializing a KB-MOEA with a population of rules covering parts of the design space. Different representations can be used for rules including a simple axis-parallel hyper-rectangular representation or a more complex kernel-based representation. Obviously there are important tradeoffs that affect choice of representation such as interpretability, computational effort, etc. Similarly, different schemes can be used in the initialization step; for instance, rules can be initialized completely randomly in terms of their volumes and locations in the search space with or without allowing overlapping between rules and constraining the initialization to fully or partially cover parts of the design space. Alternatively, a fully regular scheme with a grid-like rule initialization can also be employed. Again the design of the initialization operator will be influenced by the type of problems and available resources, etc.

This initial population of rules is then evolved iteratively to generate the final rule-based solution to the underlying optimization problem. This involves:

- generating new rules through evolutionary operations;
- sampling a set of solutions from each rule in the population, where the sampling method may vary from simple random sampling to more systematic sampling methods that would ensure coverage of bounded space by a rule. Furthermore the amount of sampling may also vary based on the problem as well as computational resource requirements;
- evaluating the quality or fitness of rules which in turn is computed as a function of volume of the rule and the relative fitness of locally sampled solutions, i.e. within a rule, to the quality of complete sample, i.e. over entire rule population;
- keeping an archive of best overall solutions discovered during the entire search at any point in time; and
- evolving next generation of rule population through systematically selecting fitter rules.

The complete algorithm is shown in Algorithm 3.

KB-EMO is generic in a sense that any existing MOEA can be adapted to evaluate the solution and in-turn the rule quality in the objective space based on the sampled solutions from the bounded space represented by each rule. Moreover, the framework is representation independent in that different hyper-volume representations can be used, including hyperrectangles which allow expressing rules in a simple if-then form. Lastly, other components, such as solution sampling scheme

Algorithm 3: An online Knowledge-Based Evolutionary Multi-objective Optimization (KB-EMO) Framework

Intput : An MOP;
A stopping criteria;
Output: A set of rules approximating the Pareto optimal patterns in the decision space;
An archive of solutions;

```
Initialization: Generate a population P_1^R of rules according to a given representation and a given initialization method;
Solution Archive: Initialize an empty archive to keep a set of best found solutions;
while no stopping criterion is satisfied do
    Rule Evolution: Apply genetic operations (selection, crossover, mutation) to generate a new rule population P_2^R;
    Solution Sampling: Select a given number of solutions from the space bounded by each rule in both P_1^R and P_2^R according to a given sampling scheme;
    Solution Evaluation: Evaluate quality or fitness of the sampled solutions, including those stored in the solution archive if it is not empty, according to a given mechanism, such as Pareto dominance ranking;
    Solution Archive Updating: Update archive based on current ranking of solutions;
    Rule Adjustment: Adjust rule boundaries by applying specific rule operations such as generalization, specialization and pruning;
    Rule Evaluation: Evaluate adjusted rules in the population based on the quality of the sampled solutions and other rule characteristics, such as rule volume;
    Environmental Selection: Apply a selection operator to choose the next generation population P_1^R;
end
return The final rule population and the solution archive.
```

and rule evolution, of the framework are also open in design and can take advantage of different techniques proposed in the literature.

KB-EMO is different from the traditional solution-based multi-objective evolutionary algorithms and promises additional advantages. First, it explores the design space using a rule-based representation. Second, the online multi-objective evolution of rules means that a set of rules are available at every step of the evolution capturing the current state of the optimization process. This is especially useful for the dynamic optimization problems where the decision makers might need a set of solutions and hence the knowledge about them at any point in time. Third, the hypervolume representation, and in particular hyper-rectangular representation, provides a powerful and intuitive way of capturing knowledge. Lastly, the rule-based representation of the optimal design space further provides decision makers a greater flexibility in exercising their preferences.

The following subsections further explain the overall principles, different operations and the implementation in detail.

4.1.1 Interaction Between Rules and Solutions

The introduction of rule based evolutionary optimization implies the evolution deals with optimization at two levels, the evolution of rules to identify optimal patterns in decision space and the evolution of solutions to be Pareto optimal. In fact, the rules and their solutions are evolved interdependently. On one hand, the quality of a rule is determined mainly by the solutions sampled from this rule. More outstanding solutions a rule contains the better it is. On the other, the generation of high quality solutions also depends on the relative location and coverage of rules to optimal areas. The richness of non-dominated solutions is determined by how close a rule is to optimal areas since the solutions are sampled from parent rules, not created by parent solutions. A rule and its sampled solutions support each other to achieve optimization goals.

The evolution at both rule and solution levels requires two evaluation methods. For solutions, their evaluation is naturally conducted using original objective functions of MOPs. For rules, a new evaluation mechanism is needed that can differentiate good rules from bad one. In other words, what we are looking for is an indicator that can show the relative optimality to distinguish rules. The comparison against the chosen indicator determines which rules are fitter for reproduction and surviving to the next generation.

The evolution at both rule and solution levels requires coordination. First, rules rely on solutions to provide the direction for evolving since the original objective function can only apply on solutions. In traditional evolutionary computation methods, this is usually done by employing elitism mechanism for solutions. In rule based evolutionary optimization, solution elitism has to be implemented explicitly when ranking the solutions. The necessity is demonstrated in Figure 4.1 using non-dominated sorting based solution evaluation. Red dots and blue dots are solutions belonging to rule 1 and rule 2 respectively, in the objective space. Rule 2 is the offspring of 1 and if there is no solution elitism, it would conclude that the red rule is better than the blue parent just because it has more non-dominated solutions.

4.1.2 The Role of Solution Archive and Rule Adjustment Operators

In KB-EMO framework, there is no solution population, therefore a solution archive is introduced to store elite solutions. When new solutions are sampled from rules, these solutions have to be estimated against the archived ones to confirm the improvement and hence provide direction information for rules to keep optimizing.

Second, a rule adjustment mechanism is introduced to facilitate convergence. The sampling of solutions brings about several problems for rules to improve. One is uncertainty. The sampling of solutions is somewhat random and the current state of rule is just an occasional visit. It cannot promise consistent evaluation over generations. The other is the counteraction between solutions. Since a group of solutions are maintained by a rule, the quality of several good solutions can be

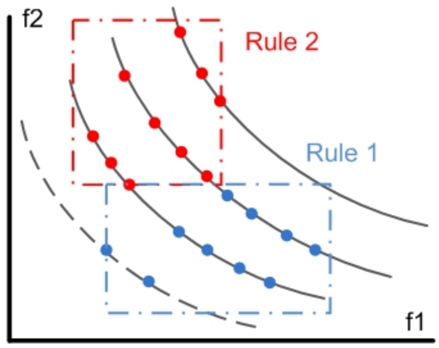


Figure 4.1: Solution Elitism for Rule Evaluation.

cancelled out by the poor quality of others when, for instance, averaging the quality of solutions to evaluate rules. These two cases make the evaluation over rules noisy and unstable and may cause stagnation during evolution process, as shown in Figure 4.2. If we simply resample Rule 1 two more times, two possible results are shown in Rule 1-1 and Rule 1-2. The left one deteriorate and the right one happens to grasp a best solution found ever but its overall performance could still be worse than the original Rule 1.

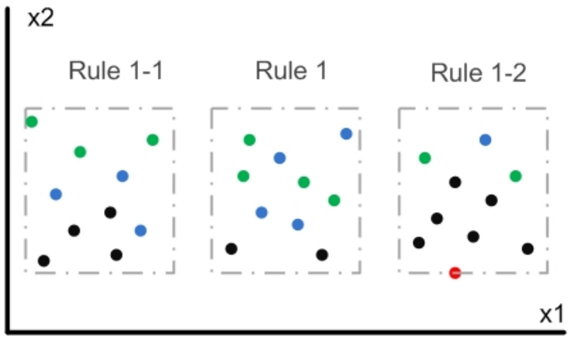


Figure 4.2: Rule resampling dilemma: Red, Green, Blue and Black indicate the solution quality from high to low.

Hence, adjustment, either pruning or shrinking, is necessary for the convergence of rules. In our algorithm, a simple shrinking method is integrated, which enables a 2 phase optimization. Firstly, the rules will investigate where the optimal solutions are. And after that, it will grow itself in size to fully occupy the optimal area. But the seeming inaccuracy caused by the random sampling doesn't lead the evolution to astray since a competent rule will be challenged over generations, only the real good rules can survive. Shrinking enables the identification of optimal areas more effectively by excluding non-optimal areas.

4.1.3 Rule Diversity in Decision Space

The target of the rule evolution is finding the optimal patterns in decision space. Hence, the distribution of optimal solutions is an important issue for the optimization of rules. This determines what rules will survive to the last. Generally, the knowledge based algorithm would be affected by the continuity, axis parallel, degeneration and size of the distribution of optimal solutions in the decision space.

If the optimal area in decision space is continuous, it implies only one optimal area exists, such as the simple rules in ZDT and DTLZ problems; otherwise, multiple optimal patterns should be expected, such as the HPS problems. The KB-MOEA also needs to consider multiple optimal areas of different sizes. If the optimal area is axis parallel, then exact identification is expected; if non-axis-parallel, the set of rules in evolution are supposed to approximate the area. Moreover, non-axis-parallel area can be of various shapes including concave, convex, etc. Degeneration means the optimal area is of lower dimension than the decision space in which it is embedded, minus one, e.g., a line segment in three dimensional space or a point in 2 dimensional space. The degeneration demands the handling of rules to be compatible with special cases, in which, for instance, there could be no optimal areas, only isolated optimal solutions existed.

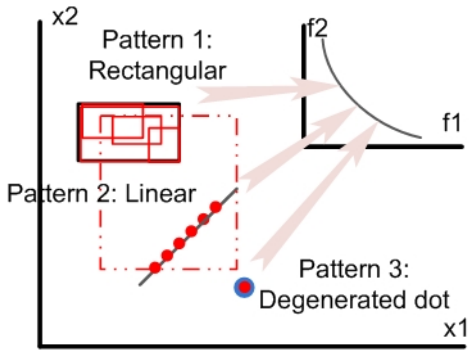


Figure 4.3: Distribution Patterns of Optimal Solutions in Decision Space.

All of the issues above are new challenges to the knowledge based evolutionary optimizers. Sometimes the algorithm has to deal with the combination of such issues, as demonstrated in Figure 4.3. There are three distribution patterns of optimal solutions: the first group of solutions clusters in a rectangle; the second group share linear relationship and the last group has only 1 isolated dot. The traditional MOEAs put an emphasis in objective space where a set of discrete and well spread solutions are searched. They usually do not take into account the distribution of optimal solutions in the parameter space, which contains important patterns in a multi-objective context. Instead, the KB-MOEA has to take both parameter space and objective space under consideration.

One concern regarding the rule diversity is an extreme case where the distribution of optimal solutions degenerates to a line or a discrete set of points. In this case, we say there exists no rules of more than 1 dimension in the MOP. The countermeasure for this is the introduction of the archive which stores the representatives of high quality solutions explored from the beginning of the evolution. With the archive, if we couldn't get the explicit rules in the end, we have a fruitful archive as a compromise, plus the rules also allowed to shrink to non-dominated solutions.

4.2 Instantiation of KB-EMO Framework

As an instantiation, a rule based MOEA ($RB - MOEA_{NS}$) is presented and discussed below with the algorithmic procedure shown in Algorithm 4, which comes with hyper-rectangular formed rules, random rule initialization, Latin Hyper Cube based solution sampling and non-dominated sorting based solution evaluation, etc. The detailed design of the operations of $RB - MOEA_{NS}$ is elaborated in following subsections.

4.2.1 Rule Representation and Initialization

The hyper-rectangular rules use an interval-based representation. In specific, a rule consists of n intervals representing n decision variable using lower bounds and the

Algorithm 4: $RB - MOEA_{NS}$

Intput : An MOP;
A stopping criteria;
Output: A set of rules approximating the Pareto optimal patterns in the decision space;
An archive of solutions;

Initialization: Initialize rule population $P_1^R = (R_1, R_2, \dots, R_N)$;
while $gen \leq Gen$ **do**
    Perform real crossover pairwise on P_1^R and mutate the rules to generate a new rule population P_2^R;
    Sample S_R solutions for each rule in P_1^R and P_2^R using Latin Hypercube method;
    Rank all $(2 \times N \times S_R)$ solutions in population P_1^R and P_2^R plus the non-dominated solutions in the archive S_A if not empty using the non-dominated sorting;
    Update the solution archive S_A;
    Shrink the rules in P_1^R and P_2^R;
    Evaluate the quality of all the rules in P_1^R and P_2^R;
    Combine P_1^R and P_2^R to generate the offspring rule population P_c^R;
    $P_1^R{=}P_c^R$;
end

interval lengths, represented as

$$(x_1^l, \delta_1, x_2^l, \delta_2, \cdots, x_n^l, \delta_n)$$

where x_i^l and δ_i denote the lower bound and the length of interval for variable x_i, respectively. Hence, the upper bound will be $x_i^u = x_i^l + \delta_i$ for i-th dimension.

The implementation of $RB{-}MOEA_{NS}$ requires initializing the rules first. Since the rules follow a hyper-rectangular representation, for problems with fewer design variables, grid based initialization is preferred. A grid allows covering the entire decision space systematically without leaving a gap or an overlap between initial rules and enables effective search over the whole design space. But the grid-based initialization is not scalable with the number of dimensions. It is usually impossible to partition a high dimensional design space, where the divisions on each dimension collectively increases the number of overall rules exponentially. For high dimensional cases, random selection of rules is suggested to initialize the very first population after partitioning of the whole search space.

4.2.2 Sampling of Solutions

While individual solutions can be sampled from the rules uniform randomly. This would require a large number of samples to get an accurate representation of the bounded area. A high number of samples however means high computational time in our framework. On the other hand fewer samples could lead to a biased sampling. So an effective sampling mechanism is important for the working of our algorithms. While sampling is a rich area of research, in this thesis we chose to use Latin Hypercube sampling (LHS) [71] which allows stratified and steady sampling of the search space using a divide and sample approach. When implemented, for a more precisely representation, only $S_R - 2$ solutions are sampled with LHS ($S - R$ is the predefined sampling size for rules). The other samples are two vertices of the hyper-rectangular rule: $(x_1^l, x_2^l, \cdots, x_n^l)$ and $(x_1^l + \delta_1, x_2^l + \delta_2, \cdots, x_n^l + \delta_n)$.

4.2.3 Rule Fitness

For knowledge-based evolution, we prefer the rules which enclose more non-dominated solutions first and then we encourage larger or more general rules over specific rules. In $RB - MOEA_{NS}$, a rule has two quality indicators, ρ and ν. ρ is evaluated as the average Pareto dominance ranking of sampled solutions from the rule and ν is used to reflect the volume of rule. The definitions emphasize rules with smaller ρ values and larger volume ν.

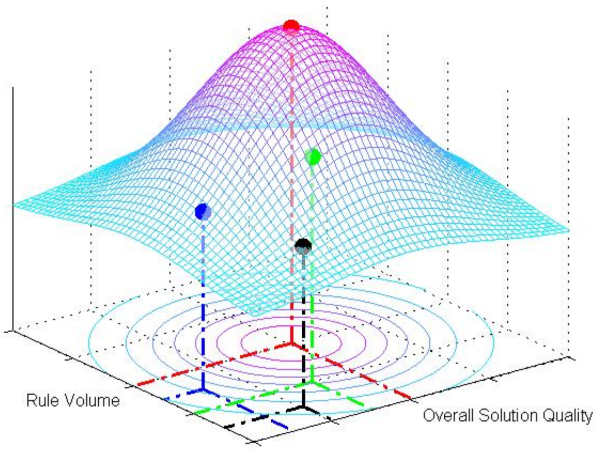
The first indicator ρ is calculated as following:

$$\rho = \frac{1}{S_R} \sum_{i=1}^{S_R} r_i \tag{4.1}$$

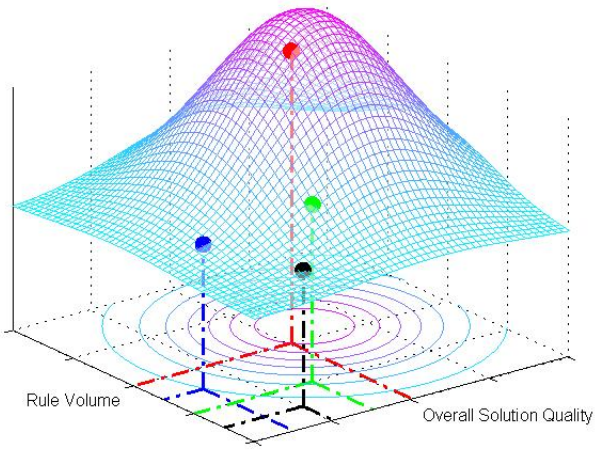
where S_R is the number of solutions sampled from a rule and r_i corresponds to the ranking values of solution i after the non-dominated sorting.

The second indicator, size of a rule ν, is computed as the summation of interval ranges in each dimension. Based the representation of rules, it is simply evaluated by:

$$\nu = \sum_{i=1}^{n} \delta_i \tag{4.2}$$



(a) The quality of 4 rules at generation t with the peak representing the best rule



(b) The quality of same 4 rules at generation $t+1$ when the peak represent new better rule identified

Figure 4.4: Using an Artificial Fitness landscape to Represent the Rule Quality

When doing environmental selection, we first prefer the rules with smaller better average ranking ρ and if they have the same ρ values, the rules with larger size will be

Table 4.1: Possibilities of How the Rule Quality Varies After Solution Archive Updating (↑, increased; -, unaltered; ↓, reduced)

Conditions	Possibilities of Rule Quality Changes
$n_A > n_A{}'$	$(\rho\uparrow,\nu\uparrow)$, $(\rho\uparrow,\nu-)$, $(\rho\uparrow,\nu\downarrow)$, $(\rho-,\nu\uparrow)$, $(\rho-,\nu-)$,$(\rho-,\nu\downarrow)$,$(\rho\downarrow,\nu\uparrow)$, $(\rho\downarrow,\nu-)$, $(\rho\downarrow,\nu\downarrow)$
$n_A = n_A{}'$	$(\rho\uparrow,\nu\uparrow)$, $(\rho\uparrow,\nu-)$, $(\rho\uparrow,\nu\downarrow)$, $(\rho-,\nu\uparrow)$

preferred. The overall quality determination mechanism of rules can be visualized as shown in Figure 4.4(a). The peak of this artificial fitness landscape indicates rules with highest ρ value and largest size ν found so far. All the other rules will be located relatively to the top. We always prefer rules with better ρ to facilitate convergence first. In the figure, the black dot will defeat the blue dot although the latter has a smaller distance to the top. As the evolution goes, the rules will gradually deteriorate down the hill as better rules will be identified and occupy the top as shown in Figure 4.4(b). Since we do not a priori know the exact optimal patterns, this artificial fitness landscape will be dynamically changing over time. When better rule is found, as shown in Figure 4.4(b), where both ρ and ν are increased, the quality of the same rules will deteriorate against the new rule.

4.2.4 Solution Elitism and Solution Archive Updating

Since the selection operation only occurs at rule level, solution elitism must be implemented explicitly, as explained above. On the solution level, in order to keep the elite, it requires to rank solutions across two consecutive generations and the solution archive to avoid isolated and misleading ranking in Figure 4.1. This means, during an evolution loop, the solutions in P_1^R, P_2^R and S_A must be pooled together for ranking rather than independently to ensure the identification and preservation of elite in current generation. By doing so, the evolution information in solutions can be transferred to rules by promising accurate rule evaluation.

The updating of solution archive is done as following: first, rank the solutions sampled from both rule populations P_1^R, P_2^R and the current archive S_A if not empty. If the number of non-dominated solutions is within the predefined limit of archive size, move all the non-dominated solutions to archive. If the number of non-dominated solutions is bigger than the archive size, then choose the non-dominated solutions with better crowding distance estimation to the archive until it is full.

Suppose the number of non-dominated solutions in solution archive S_A is n_A and after the non-dominated ranking, the number of non-dominated solutions in S_A is $n_A{}'$ (before the archive gets updated). Since it's impossible that $n_A < n_A{}'$, the possible changes to the rule quality that can be expected are listed in Table 4.1.

When $n_A > n_A{}'$, that means the original non-dominated solutions in archive are no longer the best trade-off. This means at least part of the front has been either moved forward or better diversified. In this case, the convergence of rules doesn't necessarily imply the improvement over its quality. In fact, it means comparatively more optimal areas has been touched by the rules and been explored using sampling. But the resultant rule can possibly own more, less or equal high quality solutions compared to rules in last generation.

When $n_A = n_A{}'$, that means the current non-dominated solutions are still the best trade-off. No more efficient front has been discovered. The convergence of rules is just looking for rules with better quality. If a rule with more high quality solutions has been found, the increase in ρ value is preferred no matter how ν changes. If the best rule still share the same ρ value, then the convergence is prone to the larger rule.

On the rule level, a rule's fitness is not fixed, but variable according to the updating of the efficient front among generations. As long as the solution archive keeps the best objective vectors, it will promise that better rules enjoy comparatively higher fitness over the rest.

4.2.5 Environmental Selection and Rule Diversity

In $RB-MOEA_{NS}$, when doing the environmental selection, extra attention has to be paid on the rule diversity. After the rule evaluation, if the number of rules having non-dominated solutions is not greater than the population size, keep track of all of them and select the rules with better quality to fill the rest vacancies. Otherwise, sort the rules having non-dominated solutions into two queues. The first queue is based on the average distance of its non-dominated solutions to the rest non-dominated solutions; the other is based on quality (ρ first, ν second). Each queue contributes to the offspring generation by 50 percent. Then check the first queue by setting a replaceable flag to the rules. After shrinking, if a rule still has dominated solutions inside, then it's replaceable; otherwise, it's non-replaceable. Then all the replaceable rules in the first queue will also be replaced by subsequent rules in the second queue. The aim of survival is to facilitate the convergence of rules by the second queue while keep the diversity of rules by the first queue.

4.3 Experimental Setup

4.3.1 Test Problems

The design of $RB-MOEA_{NS}$ aims to evolve knowledge directly based on traditional evolutionary operations. There are a few concerns regarding the determination of test suite for the evaluation of the performance of knowledge-based multi-objective evolutionary algorithms.

- The test functions have pre-determined optimal area(s) to test the convergence over time;
- The test functions have multiple optimal areas to test the diversity maintenance of evolution;
- The test functions are scalable to test the search ability of algorithms in comparatively high dimensional space, at least in design space;
- The test functions should also take scalable benchmark problems for comparison;

Based on the discussion above, the ZDT [169] and DTLZ [50] benchmark problems are adopted. These two suites are widely adopted for the analysis of multi-objective computation in the literature. Their well-designed Pareto front and optimal solutions facilitate comparisons for the performance of new evolutionary algorithms. The ZDT and DTLZ problems used are listed in Chapter 2. Besides, the HPS problems proposed in Chapter 3 is employed in order to test the multiple rule identification and rule diversity maintenance.

4.3.2 Performance Metrics

When assessing the performance of evolutionary algorithm in objective space, what we are expecting is not only the intuitive approximation of Pareto solutions and front by generation, but also the quantitative comparison regarding certain performance metrics. In this chapter, the generational distance (GD) and inverse generational distance (IGD) are adopted for the performance analysis in objective space.

The generational distance measure the the closeness of the approximate set of optimal solutions returned by MOEA to the real Pareto front and hence reflect the convergence level, but is not capable to report the distribution of optimal solutions over the entire front. The inverse generational distance, on the contrary, can measure both the convergence and diversity. But IGD can not reflect the quality of the whole approximate set since it's calculated inversely. For example, if you add some bad solutions to the obtained front, the IGD value won't change. But the GD value will be increased. Here, we take advantage of both GD and IGD values as our performance metric for objective space.

Regarding the decision space, the PSV metric proposed in Chapter 3 is employed. It evaluates the percentage of the convex hull area of obtained set of solutions over the overall area of optimal region. Higher values are preferred. It's more suitable to reflect the decision space concerns such as the diversity of solutions compared to other metrics, as shown in Chapter 3.

4.3.3 Competing Algorithms

The performance of $RB-MOEA_{NS}$ is compared against with other two influential algorithms in the literature, NSGA2 and MOEA/D. For all the algorithms, they share the same operations and common parameter settings. For example, Simulated Binary Crossover (SBX) with default distribution index 20 and polynomial mutation with distribution index 10 are adopted for all. The rates for crossover and mutation are 1 and $1/n$, where n is the number of variables in decision space.

The maximum generation number is set to 1000 and the evolution is recorded with different population size settings. For NSGA2 and MOEA/D, the population sizes are 100, 500, 1000, 5000 and 10000 respectively. The size limit for solution archive in $RB-MOEA_{NS}$ is also set to these values accordingly. For rule evolution, $RB-MOEA_{NS}$ maintains the rule population size of 10, 50, 100, 500 and 1000 respectively and sample 10 solutions for each rule. For each algorithm with a specific population size, we triggered the evolution 10 times with different seeds for random number generation.

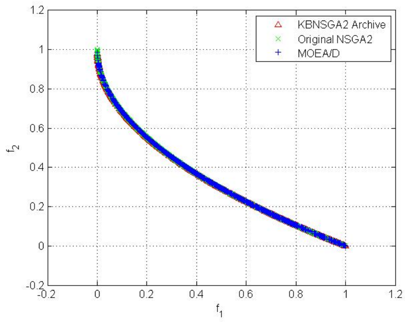
The MOEA/D version with Tchebycheff approach [162] is implemented and used here. For problems with 2 objectives, the weight vectors are generated as

$$\left\{\Big(\frac{i}{n_p-1},\frac{n_p-1-i}{n_p-1}\Big)\Big|i=0,\ldots,n_p-1\right\}$$

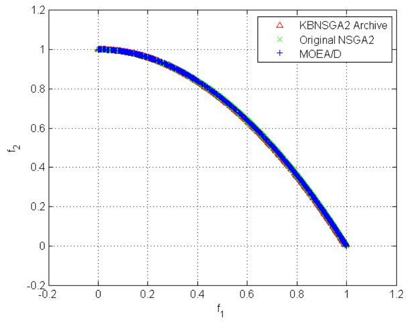
where n_p is the population size. For problems with 3 objectives, the weight vectors are generated randomly from the objective space. A weight vector is set to have 60 neighbours.

4.4 Experiment Results

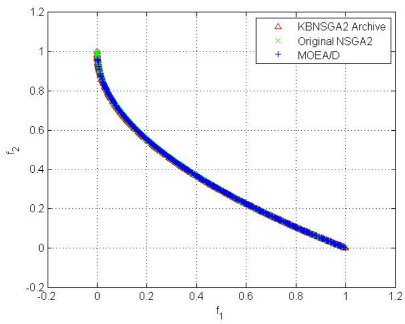
This section presents the performance analysis of $RB-MOEA_{NS}$, compared to the original NSGA2 and MOEA/D, in both objective space and decision space. The objective space performance is discussed in Section 4.4.1 covering the visualization of obtained fronts, performance measurement using GD and IGD. The decision



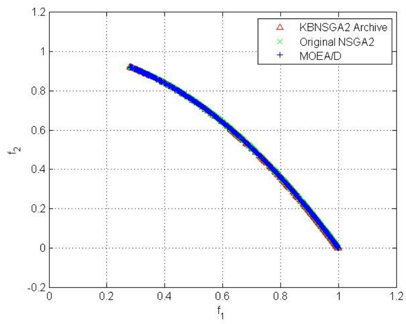
(a) ZDT1



(b) ZDT2



(c) ZDT4



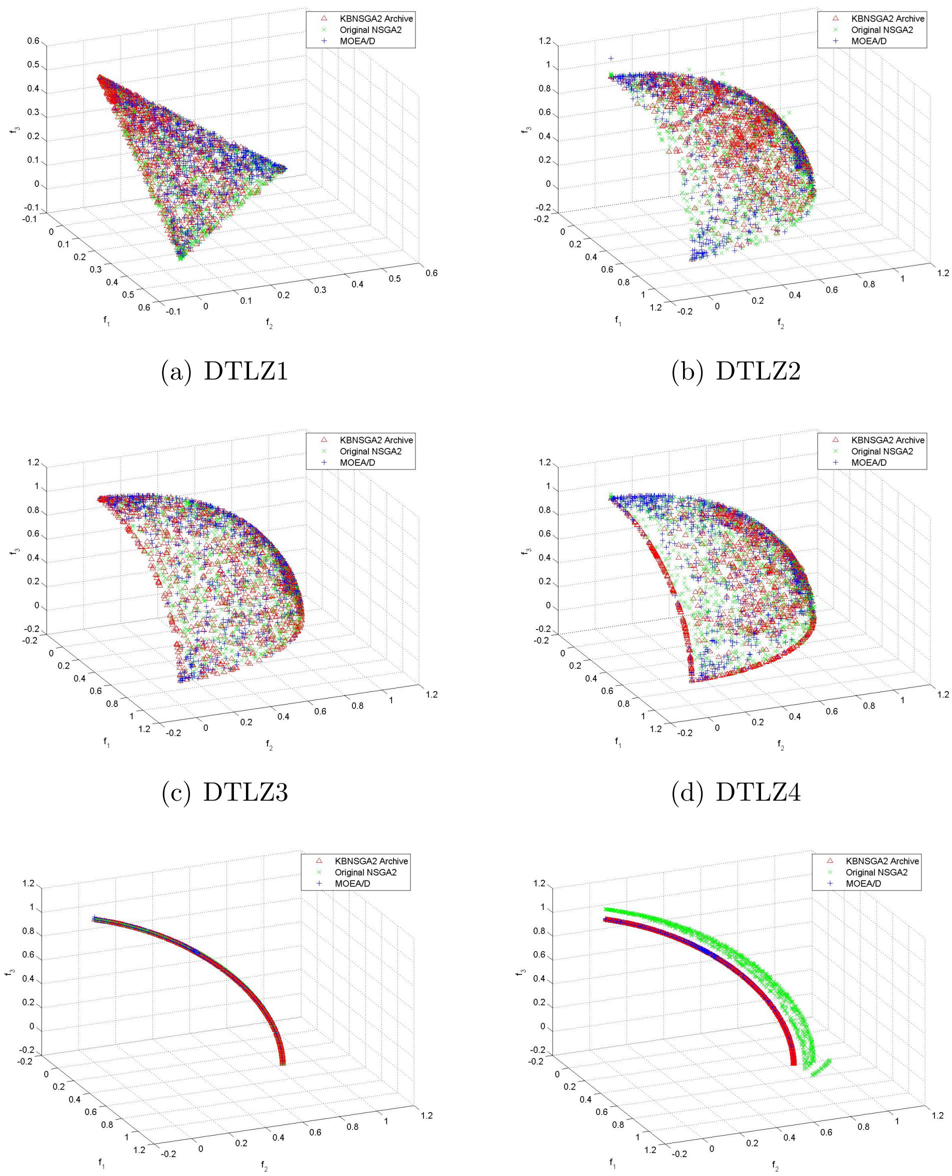
(d) ZDT6

Figure 4.5: Obtained Front of $RB - MOEA_{NS}$, NSGA2 and MOEA/D at Last Generation of Run 1 for ZDT Problems respectively.

space performance, is analysed in Section 4.4.2 presenting the rules returned by $RB - MOEA_{NS}$ and result comparison with the PSV metric, proposed in Chapter 3. The legend is consistent with $RB - MOEA_{NS}$ result in cyan, NSGA2 result in green and MOEA/D result in blue.

4.4.1 Objective Space Performance

The objective space performance analysis focuses on the convergence to the Pareto front and diversity along this front. In objective space, the obtained fronts of these algorithms are first visualized in Section 4.4.1.1 and then quantitative performance measure and comparison using GD and IGD are presented in Section 4.4.1.2 and Section 4.4.1.3.



(a) DTLZ1 (b) DTLZ2

(c) DTLZ3 (d) DTLZ4

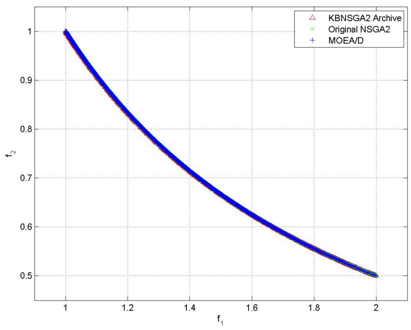
(e) DTLZ5 (f) DTLZ6

Figure 4.6: Obtained Front of $RB - MOEA_{NS}$, NSGA2 and MOEA/D at Last Generation of Run 1 for DTLZ Problems respectively.

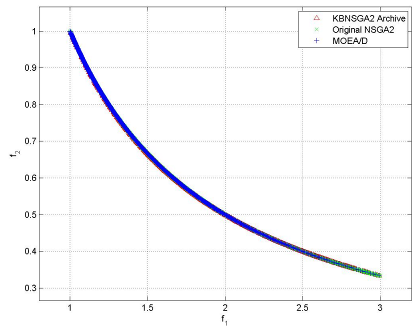
4.4.1.1 Fronts

For the three groups of test problems, ZDT, DTLZ and HPS, the fronts obtained are shown in Figure 4.5, Figure 4.6 and Figure 4.7 respectively. Only the result of the first run is plot.

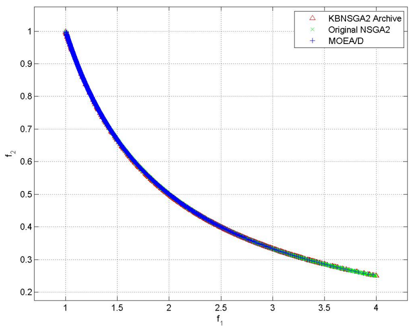
ZDT problems are bi-objective and $RB - MOEA_{NS}$ succeed to deal with con-



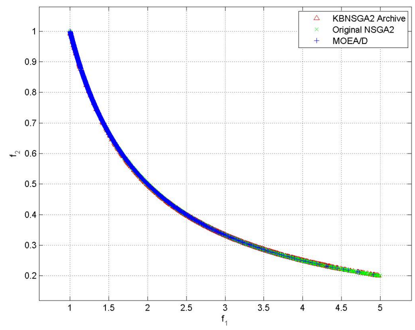
(a) HPS_1



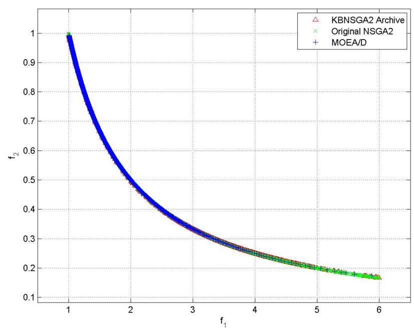
(b) HPS_2



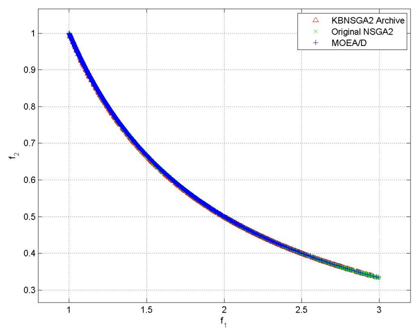
(c) HPS_3



(d) HPS_4



(e) HPS_5



(f) HPS_6

Figure 4.7: Obtained Front of $RB - MOEA_{NS}$, NSGA2 and MOEA/D at Last Generation of Run 1 for HPS Problems respectively.

vex and concave shaped front identification in ZDT1 and ZDT2, multimodality in ZDT4 and nonuniformity in ZDT6, comparable to the performance of NSGA2 and MOEA/D.

DTLZ problems are all 3-objective and have enhanced difficulties in search space. $RB - MOEA_{NS}$ not only returned competitive results over DTLZ1-DTLZ5, but also outperforms NSGA2 in DTLZ6, where the front degenerates to a single curve in 3 dimensional objective space.

HPS problems are bi-objective with mechanism diversifying the Pareto set in decision space. All the three algorithms succeed to identify the Pareto front in objective space.

Table 4.2: Average GD of ZDT Problems at Last Generation with ♦ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
	Archive	4.245e-04	3.986e-04	4.339e-04♦	4.282e-04	4.255e-04
$ZDT1$	NSGA2	5.681e-04	1.906e-01	5.937e-04	6.016e-04	5.842e-04
	MOEA/D	8.471e-02	1.601e-02	8.850e-03	9.182e-03	8.674e-03
	Archive	3.707e-04	3.663e-04	3.819e-04♦	3.678e-04	3.582e-04
$ZDT2$	NSGA2	1.954e-04	2.708e-01	3.907e-04	4.011e-04	3.936e-04
	MOEA/D	6.755e-02	1.176e-02	8.710e-03	8.592e-03	8.329e-03
	Archive	5.938e-02	5.478e-04	4.562e-04♦	4.693e-04	4.642e-04
$ZDT4$	NSGA2	9.175e-04	5.300e-04	5.962e-04	6.228e-04	5.961e-04
	MOEA/D	2.082e+00	4.155e-01	2.010e-01	1.937e-01	1.876e-01
	Archive	1.393e-02	3.427e-03	3.257e-03	3.356e-03	3.296e-03
$ZDT6$	NSGA2	1.670e-03	2.853e-01	1.677e-03♦	1.627e-03	1.605e-03
	MOEA/D	1.046e-02	2.999e-03	2.545e-03	2.411e-03	2.482e-03

4.4.1.2 GD

GD evaluates the convergence of the obtained front to the Pareto optima in objective space. The lower GD values are preferred. The GD values of $RB - MOEA_{NS}$, NSGA2 and MOEA/D over three groups of test problems with different population sizes are shown in Table 4.2, Table 4.3 and Table 4.4. A ♦ means it's significantly better than the rest using a Wilcoxon signed-rank test at the default 95% significance level. The change of GD values over time (generation) is shown in Figure 4.4.1.1, Figure 4.4.1.1 and Figure 4.4.1.1.

When comparing the performance of the three algorithms in terms of this metric, it can be observed that $RB-MOEA_{NS}$ almost always performs relatively better in most problems. In DTLZ3, the MOEA/D with Tchebycheff Approach is much poorer that the other two competitors where although most of its solutions con-

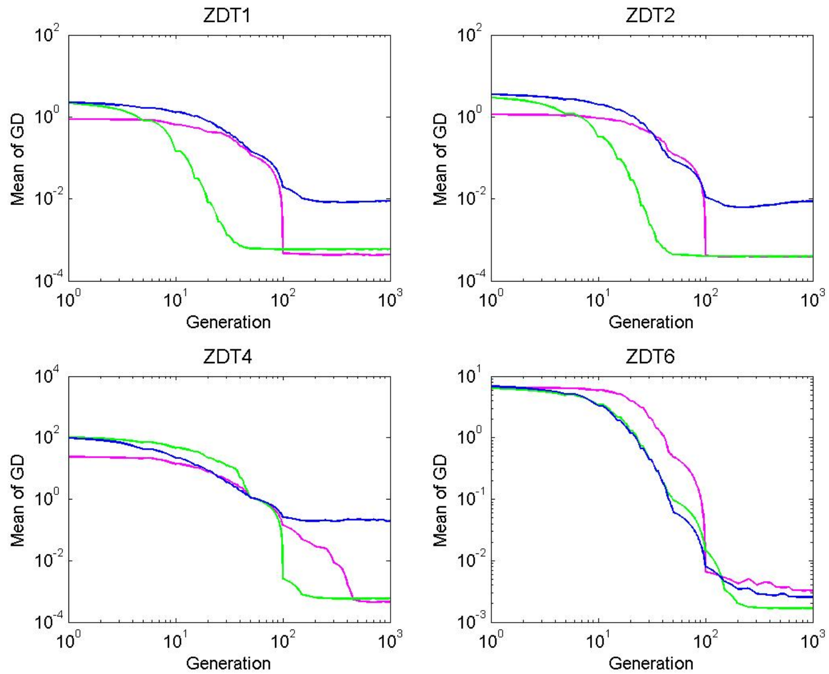


Figure 4.8: Average GD Over Time for Each ZDT Problem When Population Size is 1000

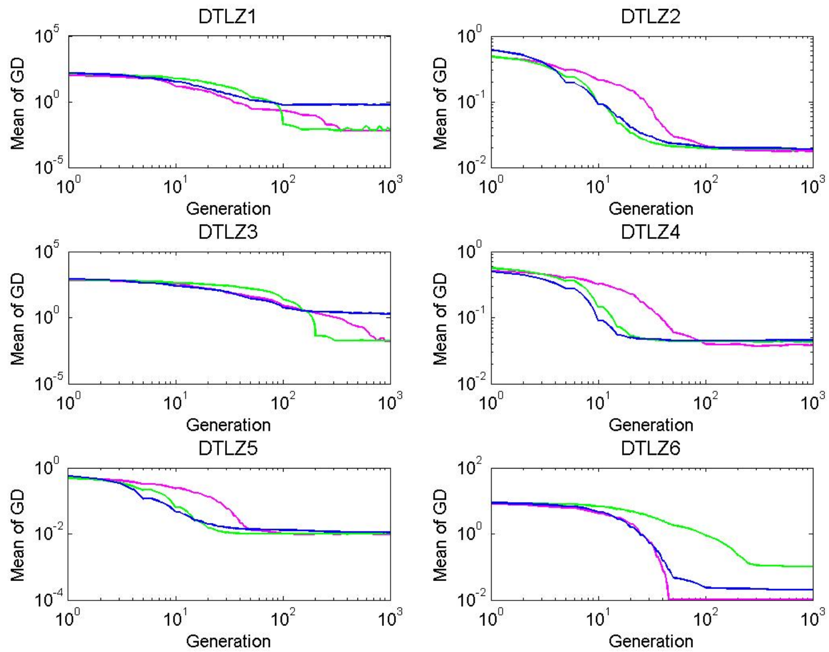


Figure 4.9: Average GD Over Time for Each DTLZ Problem When Population Size is 1000

Table 4.3: Average GD of DTLZ Problems at Last Generation with ⧫ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
	Archive	5.779e-01	4.236e-02	6.059e-03⧫	6.002e-03	5.862e-03
$DTLZ1$	NSGA2	2.415e-02	6.903e-03	6.930e-03	6.763e-03	7.086e-03
	MOEA/D	6.045e+00	1.183e+00	6.066e-01	6.028e-01	5.884e-01
	Archive	2.199e-02	1.808e-02	1.760e-02⧫	1.765e-02	1.736e-02
$DTLZ2$	NSGA2	2.498e-02	1.995e-02	1.873e-02	1.883e-02	1.784e-02
	MOEA/D	4.933e-02	2.228e-02	1.891e-02	1.848e-02	1.834e-02
	Archive	4.534e+00	1.145e-01	1.637e-02⧫	1.657e-02	1.663e-02
$DTLZ3$	NSGA2	5.883e-02	1.871e-02	1.898e-02	1.974e-02	1.855e-02
	MOEA/D	1.954e+01	3.523e+00	2.044e+00	2.020e+00	1.056e+00
	Archive	3.881e-02	3.699e-02	3.833e-02⧫	3.693e-02	3.501e-02
$DTLZ4$	NSGA2	4.306e-02	4.326e-02	4.319e-02	4.535e-02	4.447e-02
	MOEA/D	5.983e-02	4.945e-02	4.548e-02	4.622e-02	4.590e-02
	Archive	9.948e-03	1.006e-02	9.939e-03	9.671e-03	9.218e-03
$DTLZ5$	NSGA2	1.016e-02	1.003e-02	1.002e-02	1.053e-02	1.032e-02
	MOEA/D	2.831e-02	1.291e-02	1.117e-02	1.115e-02	1.106e-02
	Archive	1.013e-02	1.010e-02	1.008e-02⧫	1.028e-02	9.779e-03
$DTLZ6$	NSGA2	9.667e-02	9.975e-02	1.034e-01	1.055e-01	9.784e-02
	MOEA/D	1.415e-01	3.621e-02	2.010e-02	2.034e-02	2.083e-02

verged very well as shown in Figure 4.6(c), it has several solutions out of the scale of the figure worsening the overall GD value. It has similar situation in DTLZ1. In DTLZ6, NSGA2 is shown to be the poorest to identify the degenerated front.

Generally all algorithms show improved performance in terms of GD values with increasing population sizes, although not very obvious as IGD in next section. In most cases, the results are comparable especially between $RB - MOEA_{NS}$ and the original NSGA2 with an exception in ZDT6.

Overall, the GD curves keep decreasing over time as the evolution approaches the Pareto front. It can be seen that all algorithms converge nicely as the number of generations increase. There is no clear winner to show which algorithm converges fastest across most test problems.

To sum up, GD is popular to measure the convergence. We also used GD here to compare the performance of algorithms. It reveals that $RB - MOEA_{NS}$ enjoys some advantage over the other two mainstream algorithms in terms of this metric.

Table 4.4: Average GD of HPS Problems at Last Generation with ⧫ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
HPS_1	Archive	2.927e-03	2.909e-03	2.897e-03	2.865e-03	2.860e-03
	NSGA2	3.103e-03	2.848e-03	2.905e-03	2.899e-03	2.905e-03
	MOEA/D	9.209e-03	2.987e-03	2.995e-03	2.998e-03	2.998e-03
HPS_2	Archive	5.418e-03	5.369e-03	5.309e-03⧫	5.119e-03	5.140e-03
	NSGA2	6.397e-03	5.474e-03	5.558e-03	5.574e-03	5.588e-03
	MOEA/D	7.187e-03	5.876e-03	5.822e-03	5.822e-03	5.822e-03
HPS_3	Archive	7.677e-03	7.793e-03	7.817e-03⧫	7.581e-03	7.430e-03
	NSGA2	8.728e-03	8.171e-03	8.181e-03	8.202e-03	8.198e-03
	MOEA/D	1.235e-02	8.619e-03	8.653e-03	8.642e-03	8.635e-03
HPS_4	Archive	1.064e-02	1.020e-02	1.031e-02⧫	1.042e-02	9.969e-03
	NSGA2	1.169e-02	1.085e-02	1.082e-02	1.088e-02	1.090e-02
	MOEA/D	3.332e-02	1.153e-02	1.152e-02	1.146e-02	1.146e-02
HPS_5	Archive	1.248e-02	1.274e-02	1.290e-02⧫	1.271e-02	1.269e-02
	NSGA2	1.322e-02	1.358e-02	1.358e-02	1.351e-02	1.351e-02
	MOEA/D	3.891e-02	1.433e-02	1.430e-02	1.428e-02	1.429e-02
HPS_6	Archive	5.237e-03	5.256e-03	5.340e-03⧫	5.310e-03	5.304e-03
	NSGA2	7.961e-03	5.583e-03	5.544e-03	5.580e-03	5.567e-03
	MOEA/D	2.947e-01	5.894e-03	5.848e-03	5.823e-03	5.823e-03

Table 4.5: Average IGD of ZDT Problems at Last Generation with ⧫ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
ZDT_1	Archive	5.738e-03	1.306e-03	4.557e-04	3.325e-04	1.848e-04
	NSGA2	5.145e-03	1.151e-01	2.989e-04⧫	2.155e-04	9.308e-05
	MOEA/D	4.161e-03	9.599e-04	5.366e-04	7.946e-06	1.765e-06
ZDT_2	Archive	6.579e-03	1.415e-03	5.719e-04	2.300e-04	9.250e-05
	NSGA2	2.471e-01	9.194e-02	2.713e-04⧫	1.082e-04	8.111e-05
	MOEA/D	4.319e-03	1.138e-03	5.167e-04	1.606e-04	7.642e-06
ZDT_4	Archive	6.347e-02	1.520e-03	4.598e-04	1.902e-04	7.450e-05
	NSGA2	3.393e-01	9.807e-04	3.145e-04⧫	9.742e-05	5.901e-05
	MOEA/D	8.375e-03	8.512e-04	4.646e-04	5.263e-05	1.534e-06
ZDT_6	Archive	5.392e-03	1.158e-03	5.261e-04	2.005e-04	1.578e-04
	NSGA2	4.201e-03	2.888e-03	1.290e-04⧫	5.219e-05	4.035e-05
	MOEA/D	3.568e-03	6.515e-04	1.533e-04	1.646e-05	3.382e-06

4.4.1.3 IGD

IGD evaluates both convergence and diversity in objective space. The lower IGD values are considered better. The IGD values of $RB-MOEA_{NS}$, NSGA2 and

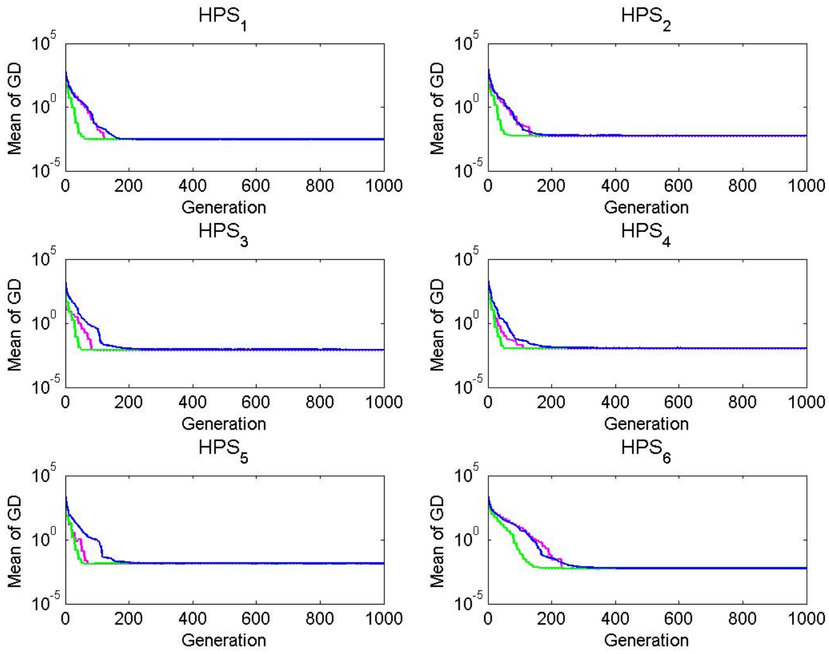


Figure 4.10: Average GD Over Time for Each HPS Problem When Population Size is 1000

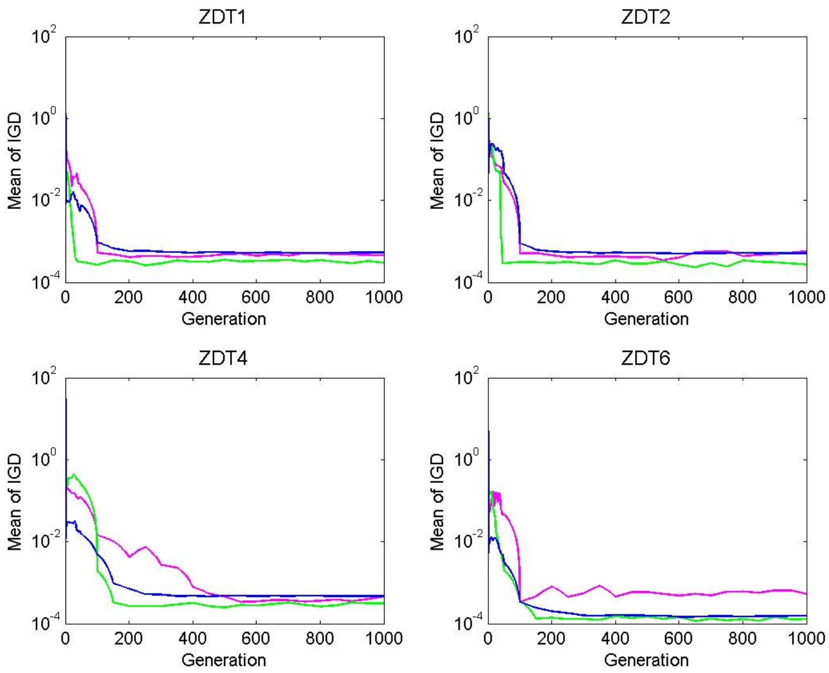


Figure 4.11: Average IGD Over Time for Each ZDT Problem When Population Size is 1000

Table 4.6: Average IGD of DTLZ Problems at Last Generation with ♦ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
	Archive	3.653e-01	6.325e-02	2.322e-04	1.727e-04	7.276e-05
*DTLZ*1	NSGA2	2.550e-02	1.108e-02	1.620e-04♦	1.233e-04	2.549e-05
	MOEA/D	2.718e-02	1.207e-02	2.901e-04	1.393e-05	1.838e-06
	Archive	7.139e-02	3.576e-02	6.183e-04	1.835e-04	9.892e-05
*DTLZ*2	NSGA2	7.144e-02	3.162e-02	3.430e-04♦	2.597e-04	5.477e-05
	MOEA/D	7.994e-02	3.311e-02	7.267e-04	3.340e-05	2.226e-06
	Archive	3.598e+00	2.770e-02	4.332e-04	3.310e-04	1.699e-04
*DTLZ*3	NSGA2	6.962e-02	2.967e-02	3.428e-04♦	1.391e-04	8.477e-05
	MOEA/D	1.018e+00	3.376e-02	7.429e-04	3.830e-05	3.320e-05
	Archive	9.230e-02	4.230e-02	3.793e-04	6.694e-05	5.820e-05
*DTLZ*4	NSGA2	1.581e-01	3.110e-02	3.271e-04♦	1.069e-04	5.895e-05
	MOEA/D	1.318e-01	3.798e-02	6.157e-04	2.975e-05	2.917e-05
	Archive	8.268e-03	2.156e-03	3.909e-04	1.533e-04	1.218e-04
*DTLZ*5	NSGA2	5.607e-03	1.030e-03	1.954e-04♦	5.848e-05	3.071e-05
	MOEA/D	1.307e-02	2.360e-03	3.580e-04	1.073e-04	1.727e-05
	Archive	1.360e-02	2.073e-03	4.005e-04	1.166e-04	9.836e-05
*DTLZ*6	NSGA2	8.440e-02	8.654e-02	3.071e-04♦	2.225e-04	7.201e-05
	MOEA/D	1.053e-02	3.917e-03	3.763e-04	8.351e-06	1.205e-06

MOEA/D over three groups of test problems with different population sizes are shown in Table 4.5, Table 4.6 and Table 4.7. A ♦ means it's significantly better than the rest using a Wilcoxon signed-rank test at the default 95% significance level. The change of IGD values over time (number of generation) is shown in Figure 4.4.1.2, Figure 4.4.1.2 and Figure 4.4.1.2.

Regarding the performance of the three algorithms in terms of this metric, it can be observed that $RB-MOEA_{NS}$ just sits between the original NSGA2 and MOEA/D with Tchebycheff Approach, although $RB-MOEA_{NS}$ is very close to the original NSGA2. In some cases of HPS problems, $RB-MOEA_{NS}$ performs better. Overall, $RB-MOEA_{NS}$ has comparable performance with NSGA2 sharing the same order of magnitudes with the competitors, but are slightly worse than the best. The reason for this is during the evolution process, we have 2 fixed sampling solutions $(x_1^l, x_2^l, \cdots, x_n^l)$ and $(x_1^l+\delta_1, x_2^l+\delta_2, \cdots, x_n^l+\delta_n)$ consisting of all the lower bounds and upper bounds of the rules. When the rules approaches the optimal area, we have many duplicates of same solutions across the rule population in spite of nominally

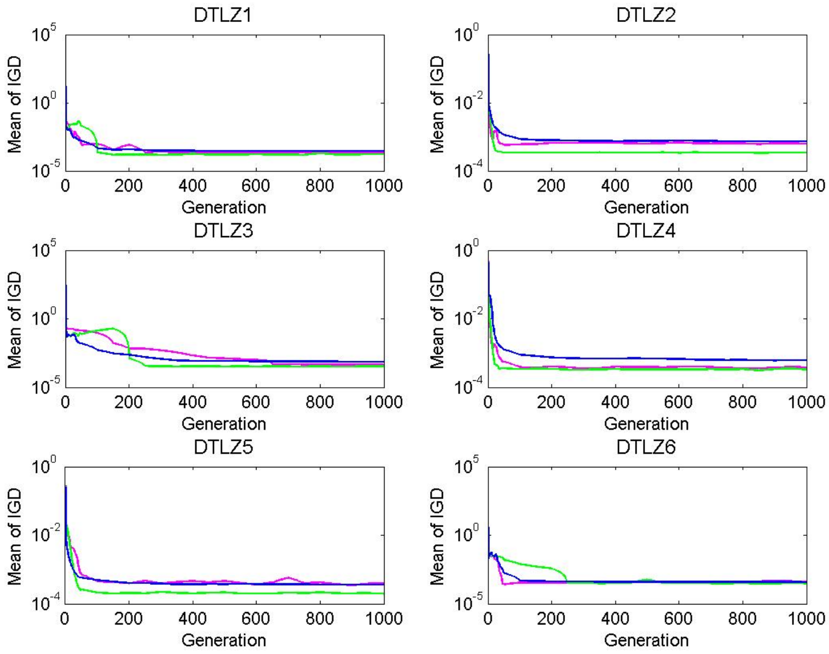


Figure 4.12: Average IGD Over Time for Each DTLZ Problem When Population Size is 1000

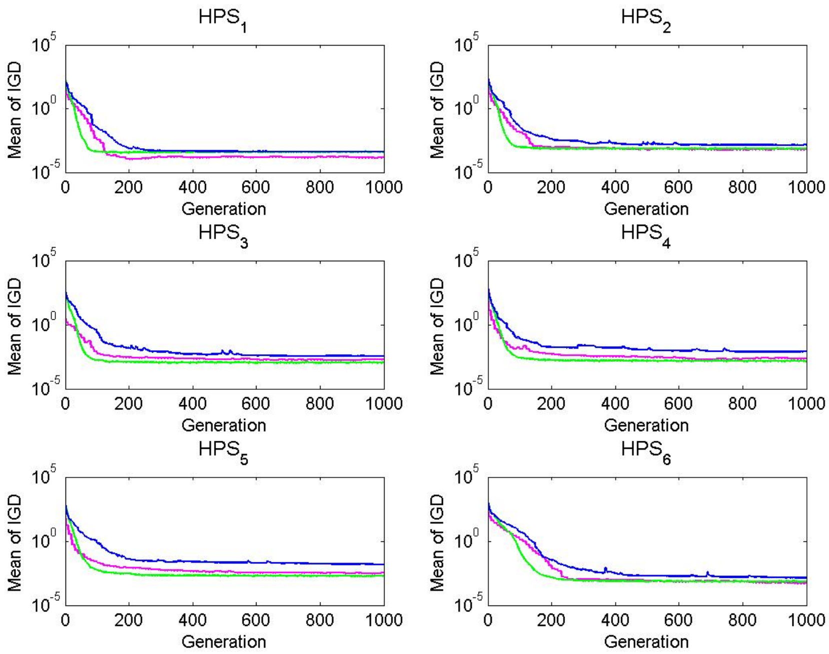


Figure 4.13: Average IGD Over Time for Each HPS Problem When Population Size is 1000

Table 4.7: Average IGD of HPS Problems at Last Generation with ♦ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
HPS_1	Archive	1.471e-03	3.378e-04	1.364e-04♦	3.040e-05	1.606e-05
	NSGA2	4.191e-03	6.834e-04	3.939e-04	7.817e-05	3.764e-05
	MOEA/D	9.757e-03	7.121e-04	4.209e-04	7.227e-05	4.218e-05
HPS_2	Archive	4.544e-03	1.383e-03	5.904e-04♦	1.959e-04	9.124e-05
	NSGA2	8.396e-03	1.491e-03	7.846e-04	1.396e-04	7.278e-05
	MOEA/D	1.568e-02	3.013e-03	1.376e-03	2.932e-04	1.420e-04
HPS_3	Archive	1.114e-02	2.853e-03	1.897e-03	4.541e-04	2.285e-04
	NSGA2	1.178e-02	2.253e-03	1.043e-03♦	2.215e-04	1.078e-04
	MOEA/D	3.901e-02	7.398e-03	3.587e-03	8.341e-04	4.054e-04
HPS_4	Archive	2.783e-02	4.099e-03	2.272e-03	5.211e-04	4.379e-04
	NSGA2	1.665e-02	3.130e-03	1.489e-03♦	3.095e-04	1.653e-04
	MOEA/D	8.153e-02	1.933e-02	8.371e-03	1.996e-03	9.350e-04
HPS_5	Archive	3.000e-02	9.054e-03	3.252e-03	1.094e-03	3.389e-04
	NSGA2	1.980e-02	3.891e-03	2.030e-03♦	4.038e-04	2.315e-04
	MOEA/D	1.361e-01	3.639e-02	1.517e-02	4.249e-03	2.127e-03
HPS_6	Archive	8.601e-03	1.654e-03	5.151e-04♦	1.665e-04	4.699e-05
	NSGA2	9.369e-03	1.519e-03	6.990e-04	1.485e-04	8.128e-05
	MOEA/D	3.181e-01	2.964e-03	1.420e-03	3.070e-04	1.551e-04

the total number of solutions maintained in the rule population is equivalent to the population size in NSGA2 in the experiment. This duplication hinders the solution diversity in the archive and hence the IGD values, which take both convergence and diversity into account.

Generally all algorithms show improved performance in terms of IGD values with increasing population sizes, more obvious than GD in the section above. The IGD curves over time or number of generations also converge nicely as the number of generations increase.

Similar to GD, IGD is also popular and is frequently employed in evolutionary computation literature to evaluate the performance the convergence. In this section, it reveals that $RB-MOEA_{NS}$ is comparable but slightly worse than the best performer in terms of this metric. Take both GD and IGD into account, we can see that $RB-MOEA_{NS}$ converges very well but sacrifices a little bit regarding diversity in objective space.

4.4.2 Decision Space Performance

The $RB - MOEA_{NS}$ algorithm directly evolves rules in decision space expecting to occupy interesting patterns of the distribution of optimal parameters. Here, we first represent the rules that $RB - MOEA_{NS}$ returned in Section 4.4.2.1 and then evaluate the performance using PSV metric in Section 4.4.2.2.

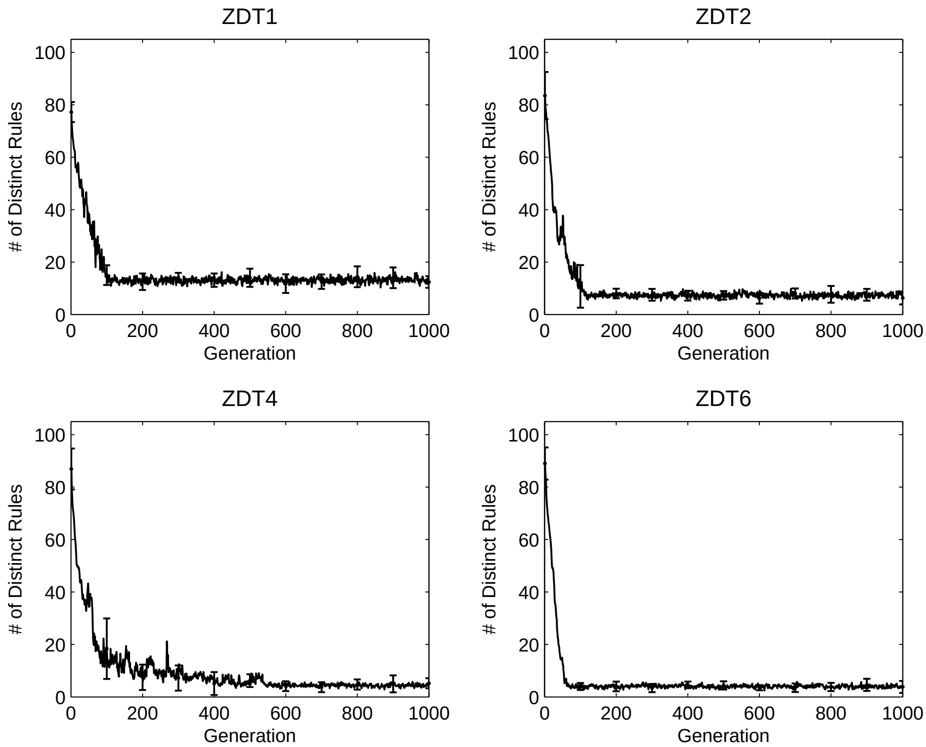
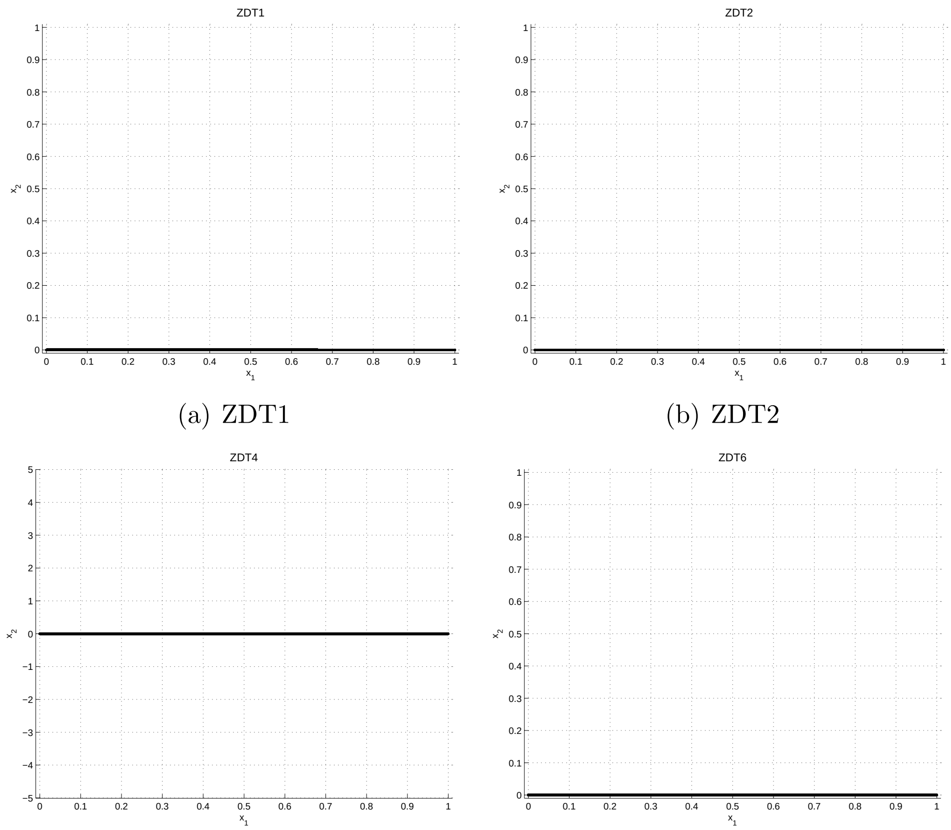


Figure 4.14: The number of distinct rules over time for ZDT problems when rule population size is 100

4.4.2.1 Rules

For knowledge-based NSGA2, not only a set of high quality non-dominated solutions are returned, but also the rules converging to the optimal patterns in decision space are expected. The latter is even more important since this is the strong motivation for the design of knowledge-based algorithms. The rule population at last generation of 1st Run for ZDT, DTLZ and HPS problems are plot in Figure 4.15, Figure 4.17 and Figure 4.19 respectively. As the rules evolve, how the number of distinct rules changes over time of 10 Runs is shown in Figure 4.14, Figure 4.16 and Figure 4.18



(a) ZDT1 (b) ZDT2

(c) ZDT4 (d) ZDT6

Figure 4.15: The rules achieved by $RB - MOEA_{NS}$ for ZDT problems at the last generation of Run 1.

with respect to different groups of problems.

For ZDT problems, the Pareto set is determined by the pattern $x_1 \in [0, 1]$ and $x_{2,\ldots} = 0$, forming a simple line segment in decision space. Figure 4.15 shows the rule population at last generation of the 1st run of $RB - MOEA_{NS}$ for ZDT problems. We only pick the first 2 decision variables for visualization (the total number of decision variables for ZDT problems are 30, 30, 10, 10 respectively). We can see that the Pareto set has been identified exactly. All the rules have converged to the optimal pattern and most of them evolved to capture the whole area. The number of distinct rules in rule population is calculated and averaged over 10 runs shown in Figure 4.14 with error bars. It is shown that $RB - MOEA_{NS}$ is capable to identify the optimal areas with minimum number of rules.

For DTLZ problems, the Pareto set is determined by the pattern $x_{1,2} \in [0, 1]$

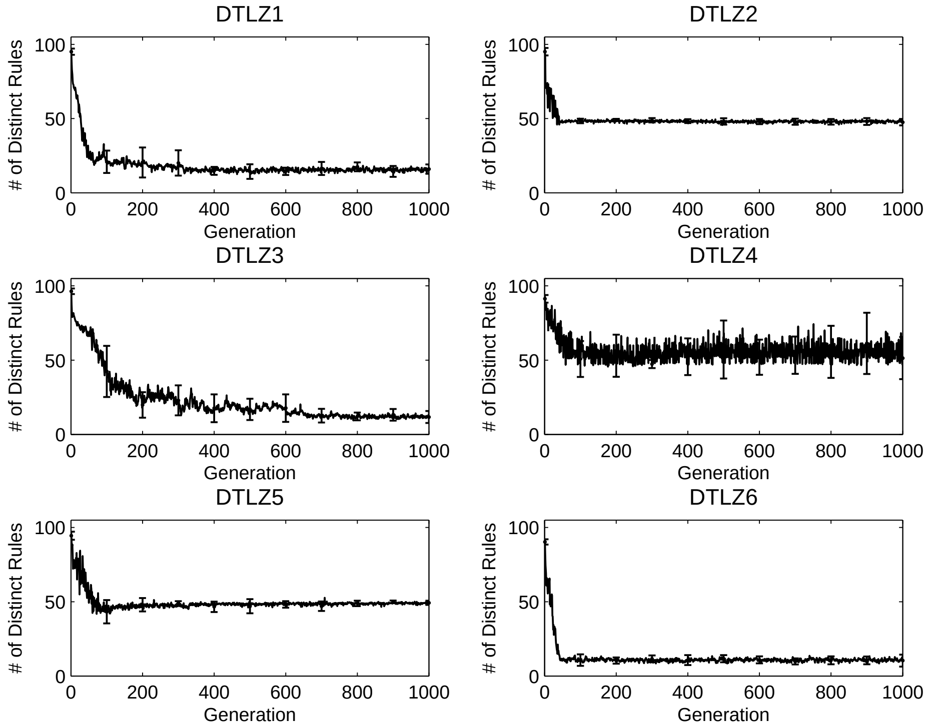
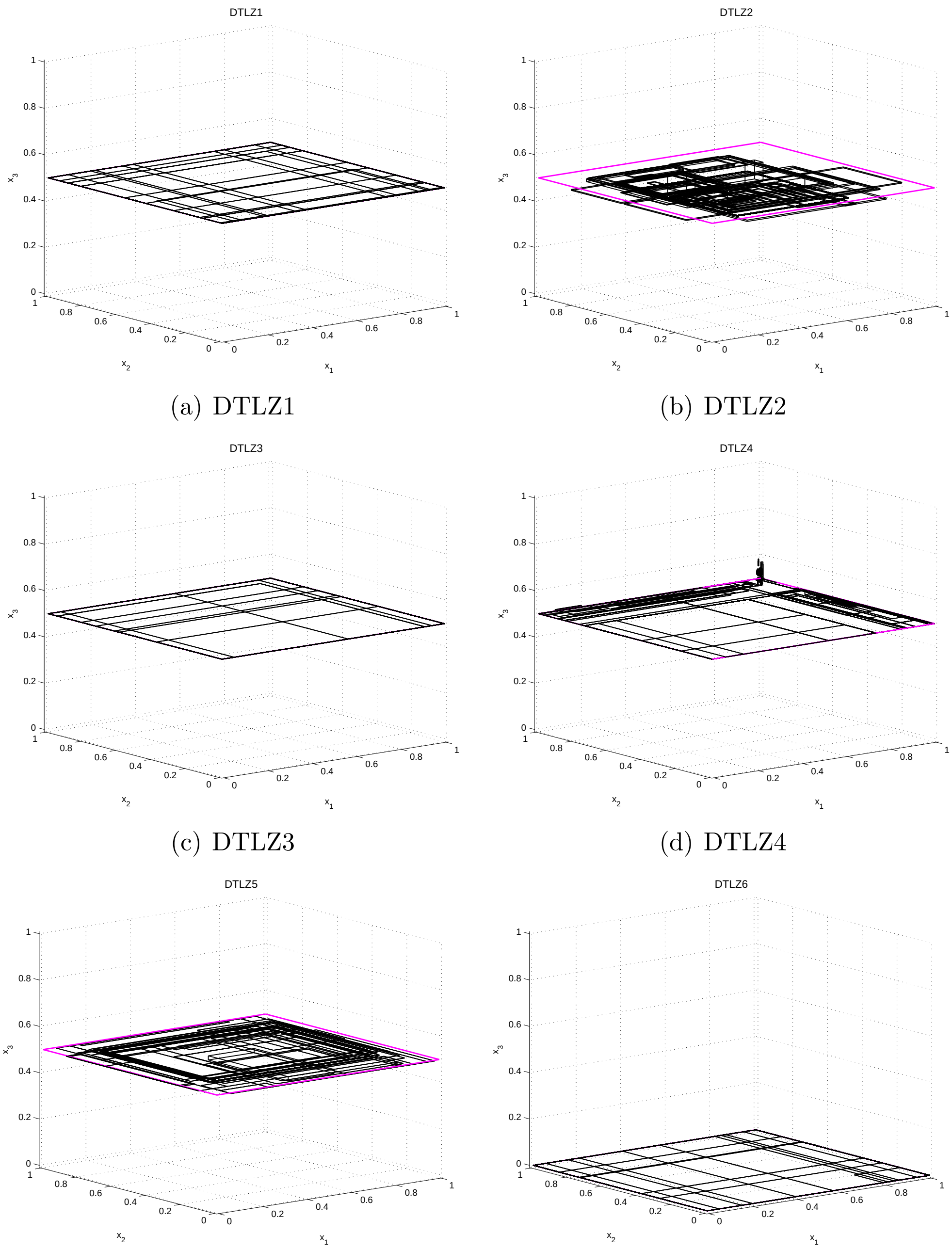


Figure 4.16: The number of distinct rules over time for DTLZ problems when rule population size is 100

and $x_{3,\dots} = 0.5$ for DTLZ1∼5 and $x_{1,2} \in [0, 1]$ and $x_{3,\dots} = 0$ for DTLZ6. Figure 4.17 shows the rule population at last generation of the 1st run of $RB - MOEA_{NS}$ for DTLZ problems. We only pick the first 3 decision variables for visualization. We can see that the Pareto set has been identified exactly for DTLZ1, DTLZ3 and DTLZ6. The perfect rules are extracted to cover the whole optimal pattern and a number of other rules located in the optimal area are provided. For DTLZ2, DTLZ4 and DTLZ5, we have a number of rules collectively occupying the majority of optimal area, where the $RB - MOEA_{NS}$ performs worst in DTLZ2. The number of distinct rules in rule population is calculated and averaged over 10 runs shown in Figure 4.16 with error bars. It is also shown that $RB - MOEA_{NS}$ is capable to converge to a handful of rules for DTLZ1, DTLZ3 and DTLZ6 compared to DTLZ2, DTLZ4 and DTLZ5.

For $HPS1 \sim 6$ problems, the number of rules is 2, 4, 8, 16, 32, 4 respectively with the first k decision variables satisfying $x_{1,\dots,k} \in [0.2, 0.4]$ or $x_{1,\dots,k} \in [0.6, 0.8]$



(a) DTLZ1 (b) DTLZ2

(c) DTLZ3 (d) DTLZ4

(e) DTLZ5 (f) DTLZ6

Figure 4.17: The rules achieved by $RB-MOEA_{NS}$ for DTLZ problems at the last generation of Run 1.

and the rest $n-k$ decision variables satisfying $x_{k+1,\dots,n} = 0.5$, as introduced in Section 3.2.3 of Chapter 3. Figure 4.19 shows the rule population at last generation of the 1st run of $RB-MOEA_{NS}$ for HPS problems. For HPS_1, HPS_2 and HPS_6, we only visualize the first 2 variables and use the first 3 variables for HPS_3. For

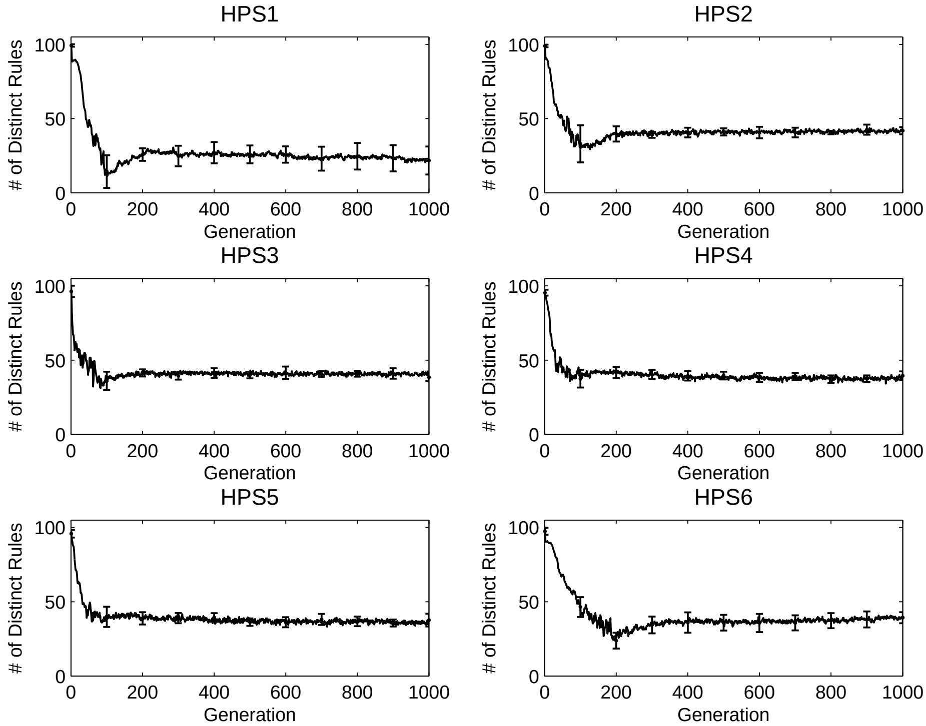
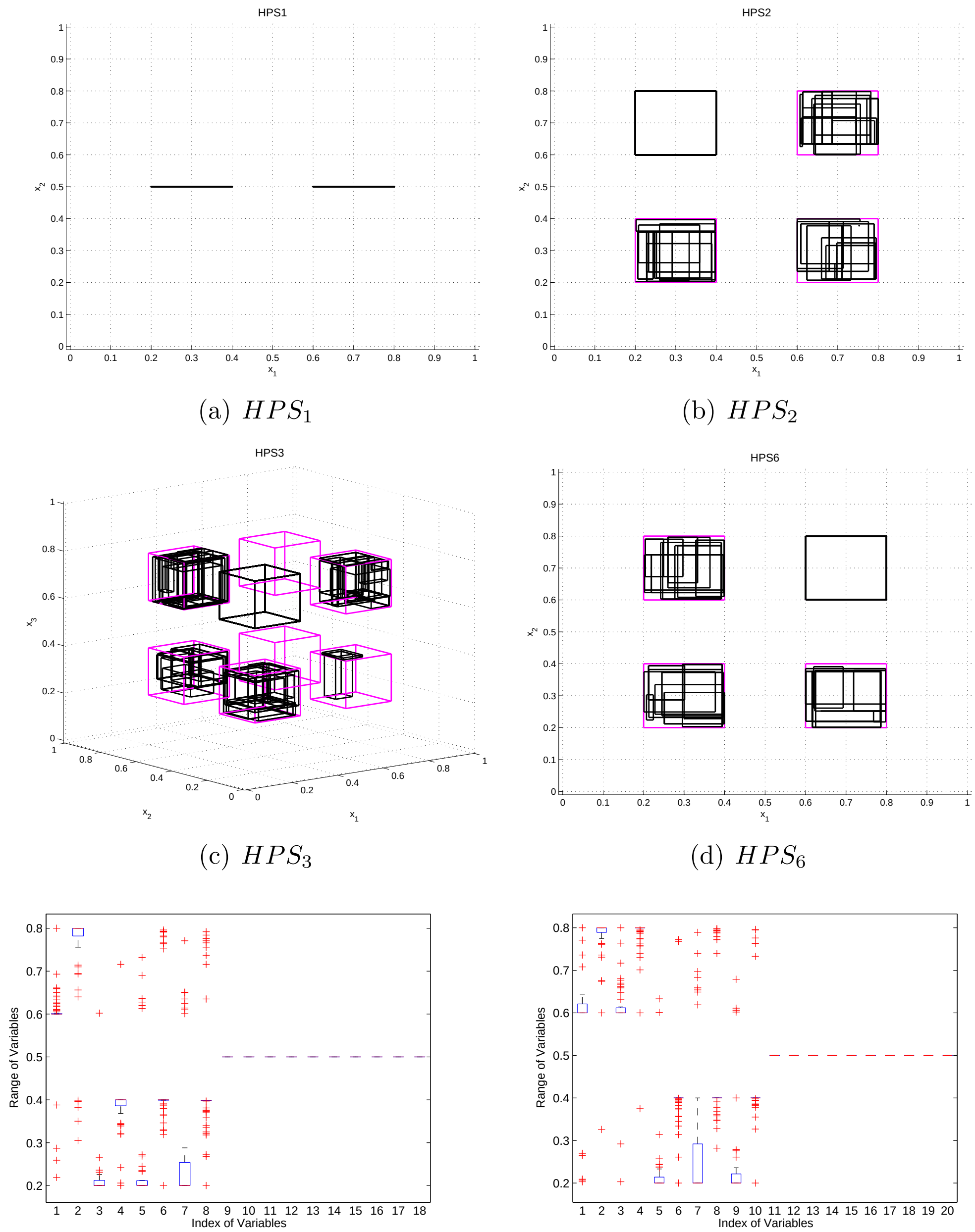


Figure 4.18: The number of distinct rules over time for HPS problems when rule population size is 100

HPS_4 and HPS_5, we use the box plot to show the range of all rule variables. The predefined line segment formed optimal pattern in HPS_1 has been fully identified. The four hyper-rectangle formed optimal pattern in HPS_2 and HPS_6 are explored efficiently with a number of rules cover these four isolated optimal areas. For HPS_3, HPS_4 and HPS_5 with 8 and more isolated patterns, $RB - MOEA_{NS}$ is able to capture part of the optimal patterns. The performance is further quantified using PSV metric in the following section. The number of distinct rules in rule population is calculated and averaged over 10 runs shown in Figure 4.18 with error bars. Since the number of optimal patterns is relatively higher than that in ZDT and DTLZ, many rules exist in the final population to capture different optimal areas.

Overall, $RB - MOEA_{NS}$ can successfully deal with the challenges in the test problems, such as comparatively high dimensionality, different Pareto front landscape, multi-modality, nonuniformity, degeneration, etc. The convergence of rules in decision space is consistent across the test suites with either perfect rules or



(a) HPS_1 (b) HPS_2

(c) HPS_3 (d) HPS_6

(e) HPS_4 (f) HPS_5

Figure 4.19: The rules achieved by $RB-MOEA_{NS}$ for HPS_1, HPS_2, HPS_3, HPS_4 problems and the box plot of Rules achieved by $RB-MOEA_{NS}$ for HPS_4, HPS_5 problems at the last generation of Run 1.

multiple rules to capture the optimal patterns in decision space.

These optimized rules returned by knowledge-based algorithm are the most inspiring. The rules provide profound understanding towards the optimization problem and great flexibility for solution implementation in a multi-objective environment. Since the optimal area has been identified by the final rule population, no matter whether a supreme rule emerged or multiple rules returned, the mapping from patterns in decision space and the Pareto optimality in objective space is therefore determined. We can further utilize rule formed knowledge to support decision making, investigate the diversity issue in both decision space and objective space since you can sample solutions from the bounded areas of rules, research the robustness concepts for more reliable engineering design or other practical problems, and so on. At the same time, the knowledge based evolutionary algorithm still keeps track of an archive of non-dominated solutions to represent the front just in case that the convergence of rules in the decision space is immature or even worse there exists no rules.

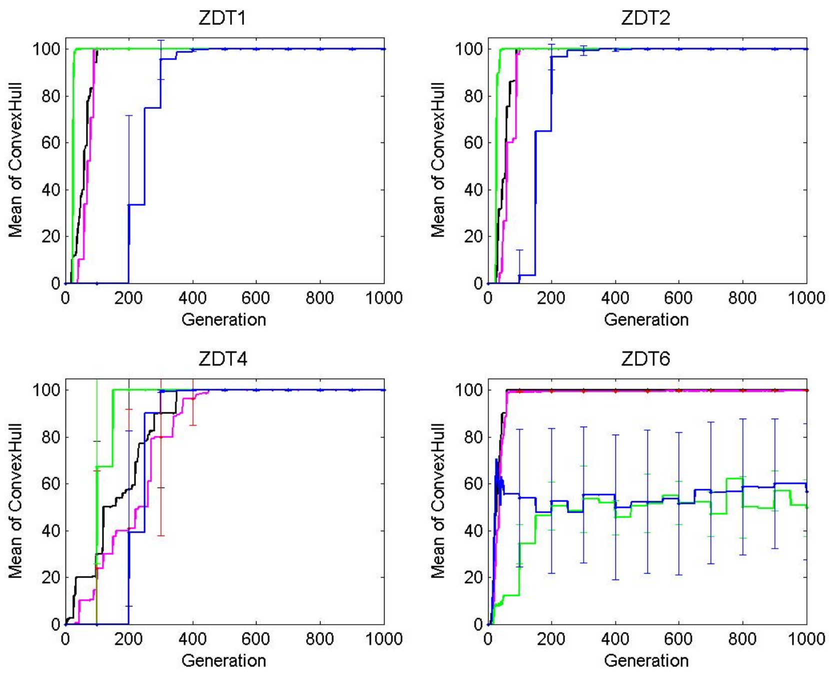


Figure 4.20: Average PSV Over Time for Each ZDT Problem When Population Size is 1000

4.4.2.2 PSV

The PSV evaluates the percentage of the area of convex hull determined by solutions over the overall area of optimal patterns. Higher value is preferred. If more

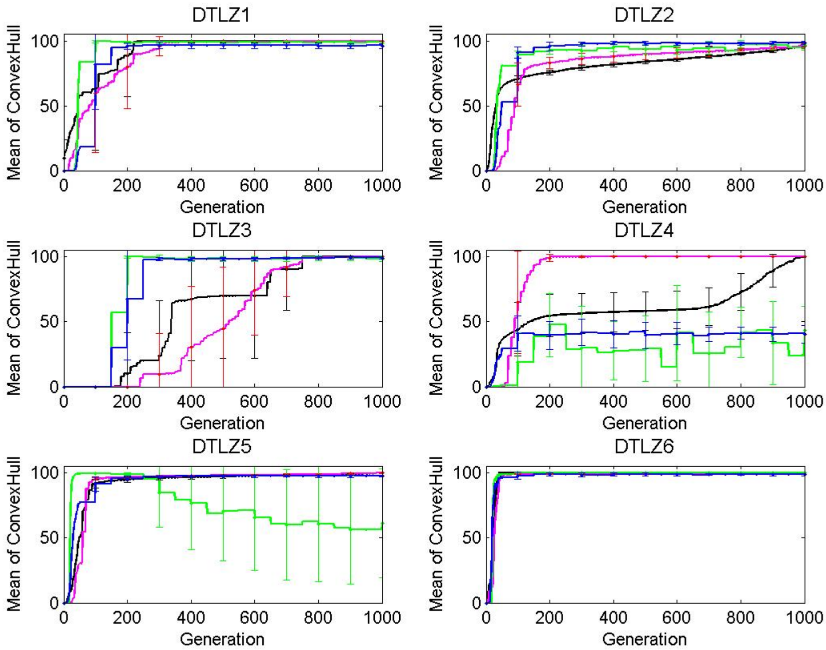


Figure 4.21: Average PSV Over Time for Each DTLZ Problem When Population Size is 1000

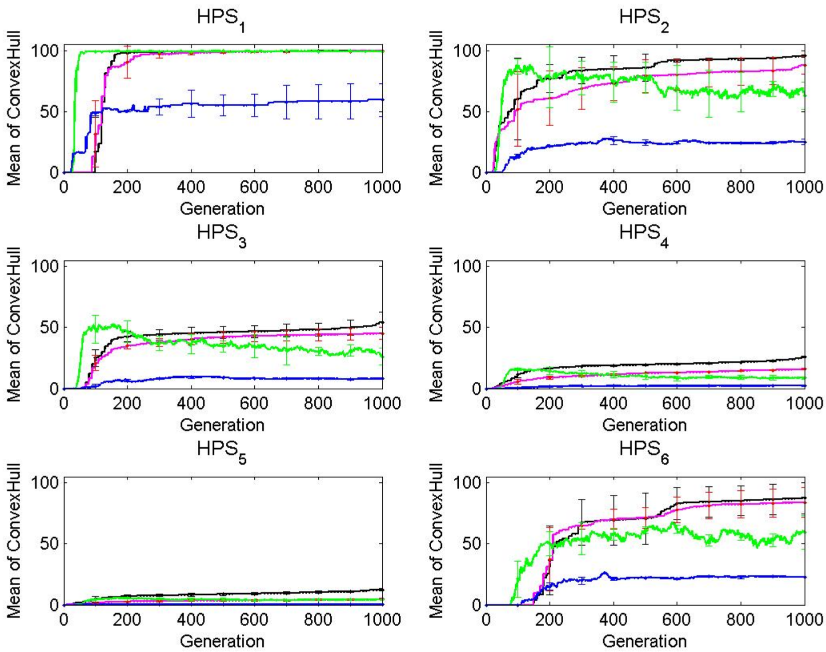


Figure 4.22: Average PSV Over Time for Each *HPS* Problem When Population Size is 1000

Table 4.8: Average PSV of ZDT Problems at Last Generation with ⧫ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
$ZDT1$	Rules	1.000e+02	1.000e+02	1.000e+02	1.000e+02	1.000e+02
	Archive	9.945e+01	9.979e+01	1.000e+02	1.000e+02	1.000e+02
	NSGA2	1.000e+02	6.954e+01	1.000e+02	1.000e+02	1.000e+02
	MOEA/D	9.131e+01	9.975e+01	9.993e+01	1.000e+02	1.000e+02
$ZDT2$	Rules	1.000e+02	1.000e+02	1.000e+02	1.000e+02	1.000e+02
	Archive	9.942e+01	9.969e+01	1.000e+02	1.000e+02	1.000e+02
	NSGA2	5.000e+01	4.391e+01	1.000e+02	1.000e+02	1.000e+02
	MOEA/D	9.087e+01	9.947e+01	9.991e+01	1.000e+02	1.000e+02
$ZDT4$	Rules	8.726e+01	9.999e+01	1.000e+02	1.000e+02	1.000e+02
	Archive	5.922e+01	9.967e+01	1.000e+02	1.000e+02	1.000e+02
	NSGA2	4.000e+01	9.000e+01	1.000e+02	1.000e+02	1.000e+02
	MOEA/D	5.683e+00	9.979e+01	9.988e+01	1.000e+02	1.000e+02
$ZDT6$	Rules	1.000e+02	1.000e+02	1.000e+02⧫	1.000e+02	1.000e+02
	Archive	9.368e+01	9.849e+01	9.986e+01	1.000e+02	1.000e+02
	NSGA2	3.933e+01	4.474e+01	4.961e+01	4.944e+01	6.324e+01
	MOEA/D	9.261e+01	8.252e+01	5.661e+01	7.946e+01	8.787e+01

than one regions are involved, they are treated separately and averaged in the final measurement. The PSV values of $RB - MOEA_{NS}$, NSGA2 and MOEA/D over three groups of test problems with different population size are shown in Table 4.8, Table 4.9 and Table 4.10. A ⧫ means it's significantly better than the rest using a Wilcoxon signed-rank test at the default 95% significance level. The change of PSV values over time (number of generations) is shown in Figure 4.4.2.1, Figure 4.4.1.1 and Figure 4.4.2.1. When evaluating the $RB - MOEA_{NS}$, we calculate not only the PSV of solutions in the archive, but also the PSV of rule vertices. They have two rows in the PSV tables called Rules and Archive to differentiate and black marks or curves are used in the figures to show the result of PSV calculation using rules.

From the tables, it can be observed that $RB - MOEA_{NS}$ outperforms NSGA2 and MOEA/D in terms of PSV. Further, the rules are usually better than the archive. The result reveals the outstanding advantage of rule based algorithms regarding the identification of optimal patterns and maintaining the diversity in decision space. This is clearly shown in some HPS problems where multiple or many patterns existed. $RB - MOEA_{NS}$ also obtains much better results in ZDT6

Table 4.9: Average PSV of DTLZ Problems at Last Generation with ⧫ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
DTLZ1	Rules	5.031e+01	9.832e+01	1.000e+02⧫	1.000e+02	1.000e+02
	Archive	1.872e+01	8.889e+01	9.988e+01	1.000e+02	1.000e+02
	NSGA2	9.505e+01	9.861e+01	9.855e+01	1.000e+02	9.949e+01
	MOEA/D	8.286e+01	9.371e+01	9.628e+01	1.000e+02	1.000e+02
DTLZ2	Rules	3.180e+01	7.326e+01	9.693e+01	1.000e+02	1.000e+02
	Archive	5.899e+00	7.503e+01	9.603e+01	1.000e+02	1.000e+02
	NSGA2	5.664e+01	8.716e+01	9.649e+01	1.000e+02	9.942e+01
	MOEA/D	5.342e+01	9.805e+01	9.827e+01⧫	1.000e+02	1.000e+02
DTLZ3	Rules	9.960e+00	9.500e+01	1.000e+02⧫	1.000e+02	1.000e+02
	Archive	0.000e+00	8.761e+01	9.974e+01	1.000e+02	1.000e+02
	NSGA2	9.641e+01	9.678e+01	9.807e+01	1.000e+02	1.000e+02
	MOEA/D	5.494e+01	9.734e+01	9.874e+01	1.000e+02	1.000e+02
DTLZ4	Rules	1.561e+01	5.668e+01	9.975e+01	1.000e+02	1.000e+02
	Archive	8.095e+01	9.316e+01	9.999e+01	1.000e+02	1.000e+02
	NSGA2	3.116e+01	2.813e+01	4.316e+01	7.015e+01	9.161e+01
	MOEA/D	1.284e+01	1.094e+01	4.024e+01	6.356e+01	6.465e+01
DTLZ5	Rules	7.705e+01	9.304e+01	9.967e+01	1.000e+02	1.000e+02
	Archive	8.646e+01	9.533e+01	9.967e+01	1.000e+02	1.000e+02
	NSGA2	1.003e+01	4.827e+01	6.101e+01	6.371e+01	8.295e+01
	MOEA/D	5.267e+01	9.349e+01	9.748e+01	1.000e+02	1.000e+02
DTLZ6	Rules	1.000e+02	1.000e+02	1.000e+02⧫	1.000e+02	1.000e+02
	Archive	7.977e+01	9.802e+01	9.980e+01	1.000e+02	1.000e+02
	NSGA2	9.965e+01	9.999e+01	1.000e+02	1.000e+02	1.000e+02
	MOEA/D	7.810e+01	9.426e+01	9.837e+01	1.000e+02	1.000e+02

and DTLZ4, which feature nonuniformity with biased density of solutions in the search space.

Generally, all algorithms show improved performance values regarding PSV with increasing population sizes. The PSV curves keep increasing over time as the evolution approaches the Pareto front.

Overall, the rule-based algorithm can not only return comparable results in objective space, but also improve the decision space performance significantly against the traditional multi-objective evolutionary algorithms.

Table 4.10: Average PSV of HPS Problems at Last Generation with ⧫ Showing Statistical Significance Over the Other Two at 0.95 Level Using Wilcoxon Test When Population Size is 1000

Population Size		100	500	1000	5000	10000
HPS_1	Rules	9.425e+01	9.942e+01	1.000e+02⧫	1.000e+02	1.000e+02
	Archive	7.309e+01	9.924e+01	9.985e+01	9.781e+01	1.000e+02
	NSGA2	9.484e+01	9.953e+01	9.960e+01	9.993e+01	9.996e+01
	MOEA/D	4.989e+01	5.817e+01	5.946e+01	7.028e+01	8.006e+01
HPS_2	Rules	3.206e+01	8.006e+01	9.589e+01⧫	1.000e+02	1.000e+02
	Archive	3.179e+01	5.867e+01	8.818e+01	1.000e+02	8.930e+01
	NSGA2	5.291e+01	6.470e+01	6.299e+01	8.110e+01	8.921e+01
	MOEA/D	1.823e+01	2.672e+01	2.505e+01	3.093e+01	4.082e+01
HPS_3	Rules	1.759e+01	3.498e+01	5.398e+01⧫	1.000e+02	1.000e+02
	Archive	8.194e+00	2.161e+01	4.523e+01	4.666e+01	6.209e+01
	NSGA2	1.121e+01	2.401e+01	2.625e+01	3.999e+01	5.516e+01
	MOEA/D	3.702e+00	6.573e+00	8.007e+00	1.259e+01	1.481e+01
HPS_4	Rules	5.745e+00	1.337e+01	2.575e+01⧫	5.477e+01	7.988e+01
	Archive	2.769e+00	8.187e+00	1.583e+01	3.296e+01	2.601e+01
	NSGA2	4.042e+00	9.213e+00	8.578e+00	1.757e+01	2.232e+01
	MOEA/D	3.389e-01	1.448e+00	2.280e+00	3.821e+00	4.706e+00
HPS_5	Rules	2.795e+00	6.671e+00	1.240e+01⧫	1.780e+01	2.631e+01
	Archive	5.522e-01	2.270e+00	4.649e+00	5.940e+00	1.189e+01
	NSGA2	1.012e+00	2.095e+00	4.252e+00	8.707e+00	1.274e+01
	MOEA/D	2.743e-02	2.937e-01	4.364e-01	1.092e+00	1.398e+00
HPS_6	Rules	2.711e+01	6.598e+01	8.736e+01⧫	1.000e+02	1.000e+02
	Archive	2.779e+01	6.248e+01	8.397e+01	1.000e+02	1.000e+02
	NSGA2	3.635e+01	6.431e+01	5.939e+01	8.480e+01	9.647e+01
	MOEA/D	4.812e+00	2.056e+01	2.293e+01	2.431e+01	3.242e+01

4.5 Summary

This chapter presents a novel knowledge-based multi-objective optimization framework which aims at searching for optimal areas of the design space instead of individual solutions using a rule-based representation. A population of rules, corresponding to the bounding areas in the design space, are evolved to search for the optimal areas. The rules are evaluated based on the quality of sampled solutions from their bounded area.

An implementation of the framework using a hyper-rectangular rule representation and non-dominated sorting based rule evaluation is presented in this chapter. The resulting algorithm is tested on a few standard benchmarks and newly design problems. The experimental results show that the algorithm is comparable to other MOEAs and able to successfully identify the optimal areas with minimum number of rules.

We believe that the knowledge-based MOEA framework presented in this chapter has important implications for many domains including engineering design and decision support systems and has a broad scope for extensions. There are a number of directions stemming from this work for future research including extensions of the framework using a number of rule and solution evaluation techniques in the literature; extensions of the framework using a number of other rule representations including fuzzy representations and thorough evaluation of the framework with robust optimization and dynamic objective functions under the above extension.

Chapter 5

Hybrid Knowledge-Based Evolutionary Many-Objective Optimization

The previous chapter defined a general framework for the design of knowledge-based evolutionary algorithms for multi-objective optimization. An implementation of the framework was presented as $RB - MOEA_{NS}$. This chapter extends the work in the previous chapter to problems with four and more objectives, commonly referred to as many-objective optimization problems. The increase in the number of objectives introduce additional challenges to the MOP. A number of well-established MOEAs exists today that handle optimization problems with two or three objectives successfully. However, the performance of these algorithms deteriorates fast when dealing with problems involving higher number of objectives. A new breed of evolutionary optimization algorithms has emerged to deal with such problems. The work presented in this chapter builds on these developments and extends the knowledge-based evolutionary optimization approach to deal with many-objective optimization problems.

Similar to knowledge-based algorithms proposed in Chapter 4 which were built upon multi-objective evolutionary algorithms, the knowledge-based evolutionary many-objective optimization algorithms proposed and investigated in this chapter mainly leverage on NSGA3, a state-of-the-art many-objective evolutionary optimization algorithms.

The chapter is organized as follows: Section 5.1 introduces many-objective optimization problems and highlights the challenges they pose for evolutionary optimization algorithms. NSGA3 is described and analyzed in Section 5.2. The corresponding proposed knowledge-based algorithms are discussed in Section 5.3 and Section 5.4. Section 5.5 provides the details of experimental setup used for the evaluation of the algorithms and the results are presented in Section 5.6. Finally, the chapter is concluded in Section 5.7.

5.1 Introduction of Many-objective Optimization

Many-objective optimization problems are a subset of MOPs which deal with four or more objectives. Such problems emerge frequently in research and engineering fields, such as industrial system design [29], air traffic control [66], and so on. Effective methods are required to handle such optimization problems.

Dealing with a large number of objectives in optimization is one of the most active research areas within the evolutionary computation community today that poses new challenges. The challenges for the scalability of optimization algorithms in many objective environments are discussed in detail in Chapter 2. A summary of these challenges using references from existing work [78][42] is presented below.

A major challenge in dealing with these problems is that the number of non-dominated solutions increase exponentially as more objectives are involved. The non-dominated solutions of an m-objective optimization problem remain non-dominated when an extra objective is introduced but bring new trade-offs among old objectives and this new objective. Since this can lead to a population that is full of non-dominated solutions, it affects Pareto-dominance based evolutionary algorithms significantly weakening their convergence properties.

Suppose $F(X) = (f_1, f_2, ..., f_m)$ and $F(X') = (f'_1, f'_2, ..., f'_m)$ are two different solutions in objective space, since $F(X) \neq F(X')$, there must exist an $i \in \mathbb{N} \cap [1, m]$ that $f_i \neq f'_i$. Suppose $f_i > f'_i$, then only in the extreme case where all the rest objectives satisfying $f_j \geq f'_j, j \in \mathbb{N} \cap [1, m] \setminus i$, a more optimized solution $F(X')$ compared to $F(X)$ can be expected. Otherwise, these two solutions are non-dominated to each other. The number of cases in this 'otherwise' category can be calculated as:

$$\sum_{k=1}^{m-1} \binom{m-1}{k} = 2^{m-1} - 1 \tag{5.1}$$

This implies that the non-dominated solutions will soon occupy all the slots in population and provide no direction information for convergence. Garza-Fabre et al. [61] shows that from generation 1, the proportion of Pareto non-dominated solutions in a randomly generated population has exceeded 90% and reached 100% from generation 3 when optimizing DTLZ1 with more than 10 objectives.

In order to address the vague selection pressure caused by the pervasive existence of non-dominated solutions, a set of reference points evenly spread in the objective space can be utilized. After clustering the non-dominated solutions based on the perpendicular distance to the reference line (from origin to reference points), the preference over solutions can be established by considering the density of solutions surrounding reference lines. Hence the reference points can be regarded as a systematic mechanism to divide the objective space and the selection is constrained within the subdivisions.

The second difficulty is about maintaining diversity in high dimensional objective space. On the one hand, evaluation of diversity will be more computationally expensive when dealing with more objectives [42]. On the other, concepts, such as crowding distance in NSGA2, can be misleading for selection and diversity maintenance in many objective environment [78]. However, after the introduction of reference points, if we are interested in the solutions closest to reference lines in each subdivision, a good distribution of objective vectors can be incurred over time.

Lastly, recombination operation can be inefficient for many-objective problems [42]. In order to search more effectively for optimal objective vectors in high dimensional environment, corresponding changes over parameter setting for evolutionary operations are required to improve the search performance. Take the SBX crossover operator [40] as an example, a relatively larger value of distribution index is suggested to mitigate the deterioration of search capability in many-objective context.

The challenge in many-objective optimization problem requires new algorithms or specific modification to current MOEAs for applications extended to many-objective environments.

5.2 NSGA3 for Many-objective Optimization

NSGA3 is essentially an extension of NSGA2 for scaling to many-objective optimization. It has shown to work well with problems up to 15 objectives [42]. It utilizes a set of predefined reference points to alleviate difficulties encountered by NSGA2 in many-objective environment.

The fitness assignment in NSGA3 is still based on Pareto dominance. In fact, NSGA3 tries to take full advantage of the search ability of Pareto dominance with the support of reference points. The set of reference points not only provide directions to guide the convergence, but also help maintain a well-spread distribution of optimized solutions. NSGA3 only keeps non-dominated solutions closest to reference lines (that extend from the origin to reference points). On the one hand, the diversity of solutions is promising once the reference points are evenly spread across the objective space. On the other, the reference lines restrict the search within its scope of directions to avoid the disturbance of pervasive non-dominated solutions. Generally, these reference points can be considered as a method to partition the objective space and mitigate the deterioration of the partial-order Pareto dominance when checking the solutions surrounding corresponding reference points individually. NSGA3 still adopts SBX crossover operation and Polynomial mutation as NSGA2, but recommends larger values for distribution indexes.

The algorithmic description of NSGA3 is shown in Algorithm 5. As can be seen, the fitness assignment is still based on dominance ranking, but the crowding distance concept to choose from competing solutions for survival has been replaced by a reference point based selection.

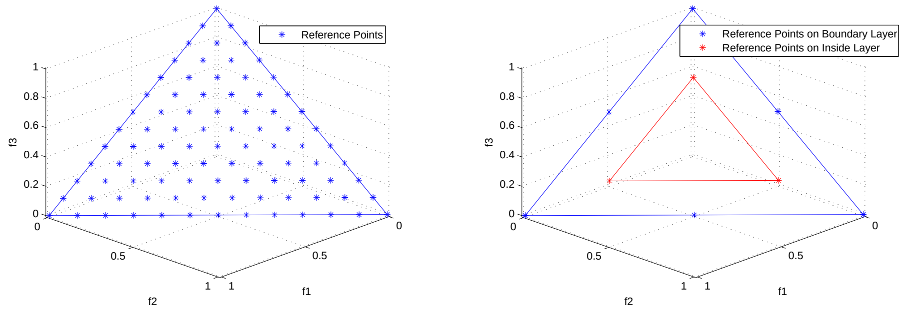
Algorithm 5: NSGA3 Algorithmic Description.

Intput : A set of reference points Z;
A stopping criterion, i.e., the maximum number of generations is Gen;
Output: A population of optimized solutions.

Initialization: Initialize solution population P_1 of size n_P;
while $gen \leq Gen$ **do**
  Evolutionary Operation: Perform crossover and mutation on P_1 to generate a new population P_2;
  Pareto Dominance Ranking: Rank all solutions in populations P_1 and P_2 based on the non-dominated sorting, $(F_1, F_2, \ldots) = Non-dominated-sort(P_1 \cup P_2)$;
  From $(F_1, F_2, \ldots)$, find the last ranked set of solutions to be included, denoted as F_l, which satisfies $\sum_{i=1}^{l} |F_i| \geq n_P$;
  if $\sum_{i=1}^{l} |F_i| = n_P$ **then**
    the next generation $P_c = \cup_{i=1}^{l} F_i$;
  else
    The next generation $P_c = \cup_{i=1}^{l-1} F_i$ and choose $k = n_P - \sum_{i=1}^{l-1} |F_i|$ solutions from F_l for P_c;
    Adaptive Normalization of solutions in $\cup_{i=1}^{l} F_i$;
    Associate solutions in $\cup_{i=1}^{l} F_i$ with a reference point based on perpendicular distance to reference lines;
    Choose k solutions from F_l one at a time to P_c based on association count of reference point;
  end
  $P_1 = P_c$;
  $gen = gen + 1$;
end

5.2.1 Generation of Reference Points

The generation of reference points plays an important role in NSGA3. A systematic approach [36] that places points evenly on a m-dimensional hyperplane, which has an intercept of 1 on each dimension, is recommended. Hence, for the coordinates of reference points $r_1, r_2, \ldots, r_m$, they satisfy $\sum_{i=1}^{m} r_i = 1$. Given the number of dimensions m and the number of divisions p on each dimension, the total number of points is



(a) Reference Points When $m = 3, p = 12$. (b) Transformation Based Inside Layer.

Figure 5.1: Generation of Reference Points

$$\binom{m+p-1}{p} \tag{5.2}$$

For instance, when $m = 4, p = 3$, there are 20 reference points shown in Table 5.1. When $m = 3, p = 12$, the number reference points generated is 91, as shown in Figure 5.1(a).

Table 5.1: Reference Points Generated When $m = 4, p = 3$

r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4	r_1	r_2	r_3	r_4
0	0	0	1	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	0	$\frac{2}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	0	0
0	0	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{3}$	$\frac{2}{3}$	0	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	0	0	$\frac{1}{3}$
0	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{2}{3}$	0	$\frac{2}{3}$	0	$\frac{1}{3}$	0
0	0	1	0	0	$\frac{2}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{1}{3}$	0	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	0	0
0	$\frac{1}{3}$	0	$\frac{2}{3}$	0	1	0	0	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	0	1	0	0	0

5.2.2 Two Layers of Reference Points

The simple issue regarding the method used here is the combinatorial number of reference points will grow too fast when raising the number of objectives. For example, when $m = 10, p = 6$, the number of reference points will be 5005. To avoid the pressure on population size caused by the explosion of reference points, NSGA3 adopts two layers of reference points when the number of objectives is greater than or equal to 8.

The first one is the boundary layer, where the reference points are all located on the intersection of Pareto optima and coordinate plane and leave the in-between space empty. The second layer is the inside layer, which is generated by performing a transformation on point $(r_1, r_2, \ldots, r_m)$:

$$r_i = \frac{1-\tau}{m} + \tau \times r_i \tag{5.3}$$

τ is set to 0.5 in implementation as suggested. Based on this, the vertices of boundary layer reference points will be transformed to the inside of the hyperplane. A 3D demonstration is shown in Figure 5.1(b). Why we use two layers of reference points is simply because we cannot afford the increase of the number of reference points if we still use the one layer method when there are only a few objectives.

Take a problem with 15 objectives as a demonstration. The number of divisions for boundary layer is 2 and we have 120 reference points on boundary lines. The number of divisions for inside layer is 1 and we have 15 reference points, which are then transformed to the inside space. This example clearly shows the distribution of reference points is not even and very sparse and restricted to represent the entire objective space since we have only 135 reference points in total for a problem with 15 objectives and most of them (88.89%) reflects only trade-off of two objectives.

Overall, the introduction of reference points in NSGA3 help maintain the diversity and constrain evolution along certain directions. The drawback is that it limits the representation of the trade-off surface in many-objective optimization problems. Since the combinatorial number grows faster compared to the fixed size of population, although a set of high quality and well spread solutions can be expected, they locate very sparsely in the objective space. This may get the decision maker to a dilemma where he has a set of optimized solutions, but find no one can be implemented.

5.3 KB-MOEA for Many-objective Optimization

In Chapter 4, a KB-EMO framework is proposed and implemented as $RB{-}MOEA_{NS}$. The resulting algorithm evolves a population of rules directly and is evaluated with test problems having 2 or 3 objectives. In this section, the framework is implemented with the utilization of a set of reference points based on NSGA3 to facilitate the online knowledge extraction from many-objective optimization problems.

5.3.1 $RB - MOEA_{REF}$

The resulting algorithm of KB-EMO Framework using reference points is termed as $RB - MOEA_{REF}$. The algorithmic description of $RB - MOEA_{REF}$ is provided in Algorithm 6.

Algorithm 6: Algorithmic Description of $RB - MOEA_{REF}$

Intput : A set of evenly spread reference points in objective space;
Initialization: Initialize rule population P_1^R of size N;
for $gen \leftarrow 1$ **to** Gen **do**
    Evolutionary Operations: Perform real crossover and mutation on pairwise rules in P_1^R to generate a new rule population P_2^R;
    Solution Sampling and Evaluation: Sample S_R solutions using each rule in P_1^R and P_2^R using Latin Hypercube method;
    Dominance Ranking: Rank all $(2 \times N \times S_R)$ solutions in population P_1^R and P_2^R plus the non-dominated solutions in the archive S_A if not empty using the non-dominated sorting to generate front levels $(F_1, F_2, \ldots)$;
    Solution Archive Updating: Updating the solution archive S_A using Algorithm 7 with the non-dominated ranking result $(F_1, F_2, \ldots)$ and the set of reference points;
    Rule Shrinking: Shrink the rules in P_1^R and P_2^R;
    Rule Quality Evaluation: Evaluate the quality of all the rules in P_1^R and P_2^R;
    Survival: Combine P_1^R and P_2^R to generate the offspring population P_c^R;
    $P_1^R = P_c^R$;
end

The biggest difference between $RB - MOEA_{REF}$ in this section and $RB - MOEA_{NS}$ in Chapter 4 is the mechanism for solution archive updating. Although the solution quality in $RB - MOEA_{REF}$ is still based on Pareto dominance, the selection is many-objective environment is assisted by a set of reference points. It calculates the density of solutions associated to reference points and the less crowded reference points has a larger probability to contribute to the archive update. On the contrary, in $RB - MOEA_{NS}$, the solution updating is based on Pareto dominance

Algorithm 7: Solution Archive Updating

Intput : Front levels of solutions after non-dominated ranking $(F_1, F_2, \ldots)$;
A set of evenly spread reference points in objective space Z;

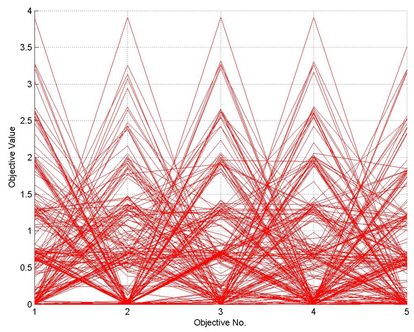
if $|F_1| <=$ *size (the predefined size limit of solution archive)* **then**
  Solution Archive $S_A = F_1$ of actual size $|F_1|$;
  return;
else
  Normalize objective vectors in F_1;
  Associate each solution in F_1 with a reference point in Z, supposing the number of solutions associated with j-th reference point is ϵ_j;
  while *the solution archive is not full* **do**
    Exclude the reference points without associated solutions ($\epsilon_j = 0$);
    Identify the reference points having the least number of solutions $J_{min} = \{j : argmin_j \epsilon_j\}$;
    Choose one reference point randomly from J_{min}, denoted as the $\bar{j}$-th reference point;
    Move the solution having the shortest perpendicular distance to the reference line to the solution archive;
    $\epsilon_{\bar{j}} = \epsilon_{\bar{j}} - 1$;
  end
end

and crowding distance.

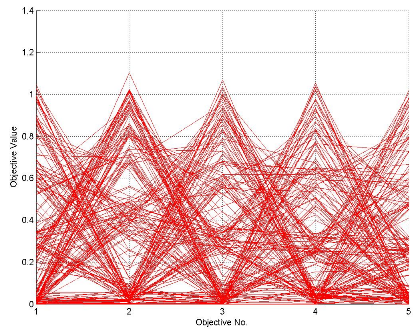
For other components, $RB-MOEA_{REF}$ adopts the same designs as $RB-MOEA_{NS}$ including rule representation, solution sampling mechanism, rule quality evaluation, rule shrinking mechanism and environmental selection for rules.

5.3.2 Experimentation

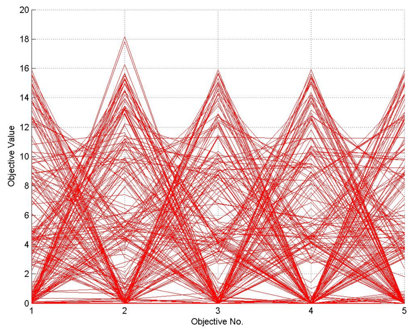
The performance of $RB-MOEA_{REF}$ is test with DTLZ1∼4 problems with 5, 8, 10 and 15 objectives respectively, as shown in Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5. For DTLZ1 problems, the objective vectors are supposed to satisfy $\sum f_i = 0.5$ and for the other $\sum f_i^2 = 1$. $RB-MOEA_{REF}$ can achieve fair performance for DTLZ2 and DTLZ4 instances, but not satisfying with DTLZ1 and DTLZ3, where the objective vectors are far away from the optima.



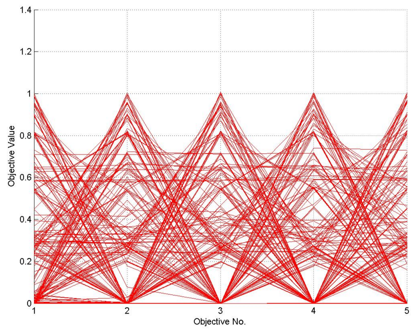
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



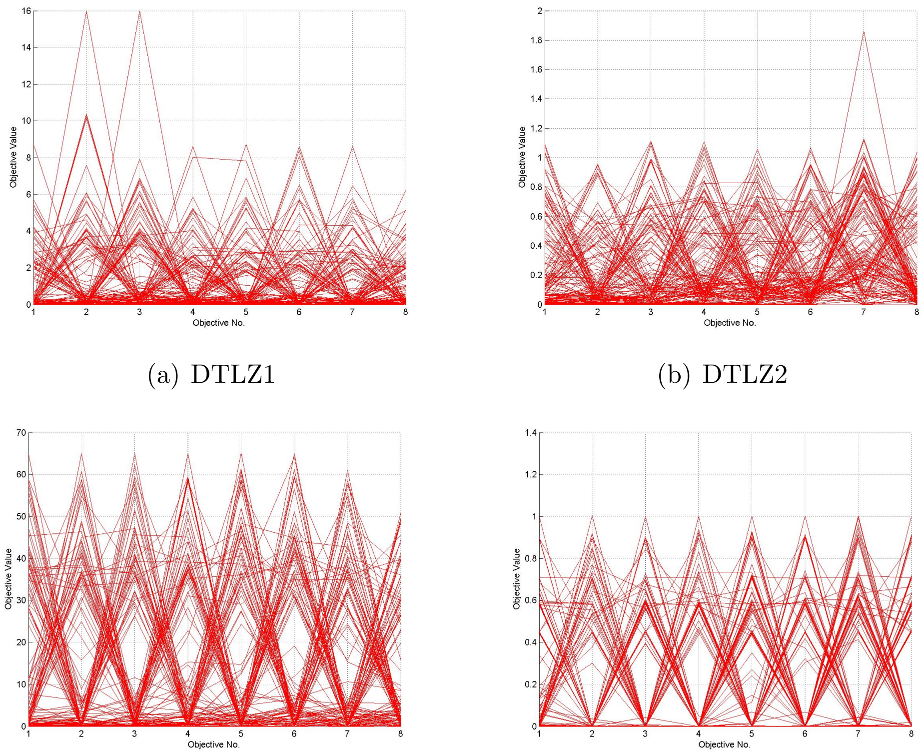
(d) DTLZ4

Figure 5.2: Obtained Objective Vectors of $RB - MOEA_{REF}$ for DTLZ problems with 5 Objectives

5.3.3 Discussion Over $RB - MOEA_{REF}$

The KB-MOEA, $RB - MOEA_{REF}$, evolves rule-represented knowledge, aims at investigating the optimal distribution patterns in the decision space when many objectives are involved. However, the rule-represented knowledge evolution still has to deal with the same problems, such as the pervasive non-dominated solutions in objective space. Rules with a comparatively smaller sampling size are more likely to be full of non-dominated solutions when they are still far away from the optimal area. Generally, the $RB - MOEA_{REF}$ suffers from finding convergence direction.

This can be mitigated in two ways. First, we can modify Pareto dominance to decrease the number of non-dominated solutions or assign different ranks to non-dominated solutions, as proposed in research [118][91][34]. Second, we can adopt a hybrid design sitting on top of the original NSGA3. The hybrid approach starts



(a) DTLZ1 (b) DTLZ2

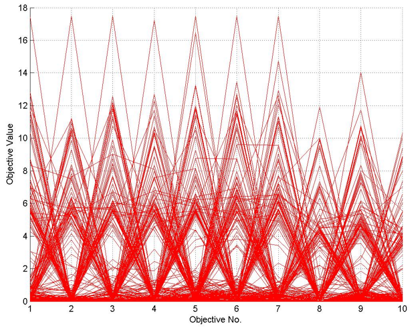
(c) DTLZ3 (d) DTLZ4

Figure 5.3: Obtained Objective Vectors of $RB - MOEA_{REF}$ for DTLZ problems with 8 Objectives

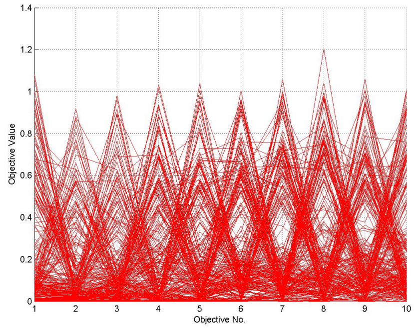
NSGA3 from the very beginning and then after a period of evolution, say half of the generations have completed, switch to knowledge-based evolution. In other words, this hybrid handling starts to evolve rules from a comparatively high level. The purpose is simply to promise the benefits of knowledge evolution but avoid the stagnation when the rules are immature. The following section will elaborate the hybrid design of knowledge-based many-objective evolutionary optimization.

5.4 Hybrid Approach for $RB - MOEA_{REF}$

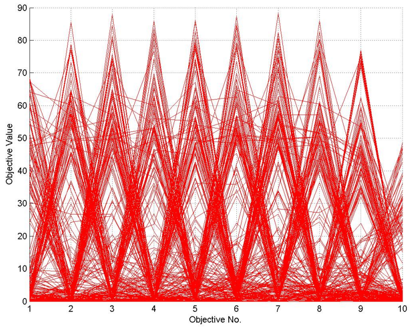
This section introduces the hybrid approach combining the latest development in many-objective optimization and knowledge based evolution. The resulting algorithm using this hybrid design starts original NSGA3 first but switches to knowledge based evolution at some generation, to identify optimal patterns in decision space.



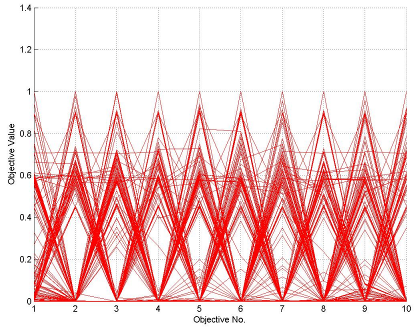
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



(d) DTLZ4

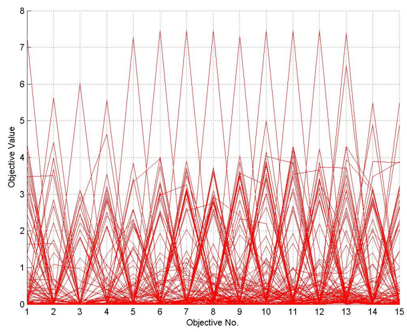
Figure 5.4: Obtained Objective Vectors of $RB - MOEA_{REF}$ for DTLZ problems with 10 Objectives

5.4.1 Hybrid $RB - MOEA_{REF}$

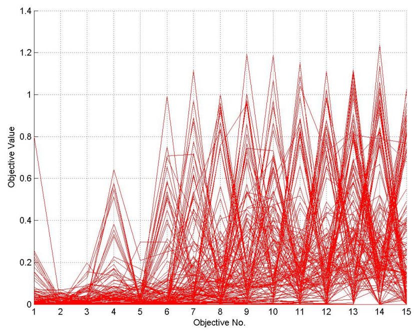
Based on the selection of switching point between solution-based NSGA3 evolution and rule-based knowledge evolution, we adopt two setups in the design of the hybrid algorithm. First, we start original NSGA3 for the first half of generations and then switch to the knowledge evolution from halfway to the end. Second, we alternate original NSGA3 and knowledge evolution every 100 generations.

5.4.2 Hybrid $RB - MOEA_{REF}$ - Single Switch

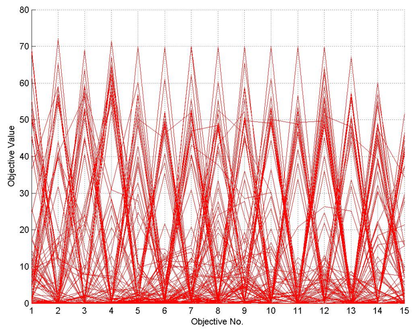
For the first setup, the Hybrid $RB - MOEA_{REF}$ starts with normal NSGA3 procedure, such as population initialization, crossover, mutation and environmental selection until to the middle of the whole optimization process. The middle point is selected for switching to the rule based search, $RB-MOEA_{REF}$. It treats the solu-



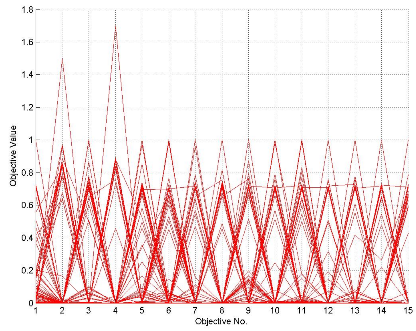
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



(d) DTLZ4

Figure 5.5: Obtained Objective Vectors of $RB - MOEA_{REF}$ for DTLZ problems with 15 Objectives

tion population as the initial archive and utilizes k-means clustering to group these solutions. The bounds of these groups are further used to initialize the rule population and then the rule evolution starts. The whole process is shown in Algorithm 8.

This hybrid approach is still able to bring the benefits that a knowledge-based algorithm is expected to present:

- A set of rules that reveal the patterns of Pareto optimal solutions in decision space;
- A set of optimized solutions with comparable quality compared to the result of original algorithm.

Here, the first benefit is especially useful in many-objective optimization. It's clear that even when traditional MOEAs can converge in a many-objective envi-

Algorithm 8: Hybrid $RB-MOEA_{REF}$ with Single Switch Algorithmic Description.

Intput : A set of reference points Z;
A stopping criterion, i.e., the maximum number of generations is Gen;
Output: A set of optimized rules;
An archive of optimized solutions.

Initialize solution population P_1^S of size n_P;
while $gen \leq Gen/2$ **do**
  Perform crossover and mutation on P_1^S to generate a new population P_2^S;
  Utilize NSGA3 operations in Algorithm 5 to obtain next generation P_C^S;
  $P_1^S = P_C^S$;
end
Initialize solution archive S_A: $S_A = P_1^S$;
Initialize rule population P_1^R of size N using bounds of clusters from archive S_A generated using k-Means method;
while $Gen/2 < gen \leq Gen$ **do**
  Perform real crossover and mutation on P_1^R to generate a new rule population P_2^R;
  Sample S_R solutions from each rule in P_1^R and P_2^R;
  Rank all $(2 \times N \times S_R)$ solutions in population P_1^R and P_2^R plus the solutions in the archive S_A using the non-dominated sorting;
  Update the solution archive S_A using Algorithm 7;
  Shrink the rules in P_1^S and P_2^S;
  Evaluate the quality of all the rules in P_1^S and P_2^S;
  Combine P_1^R and P_2^R to generate the offspring population P_c^R. Select the rules with better quality first and then the rules with bigger volume if sharing the same quality;
  $P_1^R = P_c^R$;
end

ronment after improvement, its representation capability of the Pareto optimality is very limited if it's individual point based. In this case, patterns, instead of solutions, are more suitable to describe the Pareto optimality.

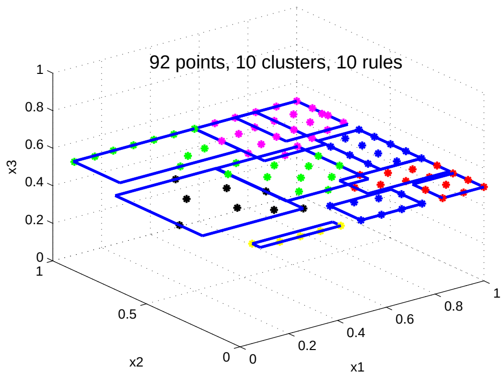
An important question here is why we should start rule evolution from the middle? First, the reason why we make it hybrid is to start the rule evolution at a comparatively high level to mitigate the influence of pervasive non-dominated solutions in the many-objective environment. Hence, it will make little sense to introduce hybridization if we switch to rule evolution too early, for example, at 50th generation (if the maximum generation number is 500). Second, the rule evolution will be immature if we switch too late. In the evolution, rules will be evaluated again and again by resampling solutions over generations before being eliminated from the rule population. After the optimization, the rules returned are all winners not only over bad rules, but also over their previous immature forms. Overall, it's not a must to switch to rule evolution at the middle point, but a moderate switching point is recommended.

5.4.3 Hybrid $RB - MOEA_{REF}$ - Alternating Switch

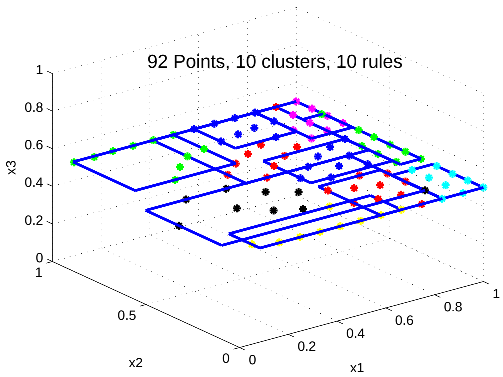
The second setup executes NSGA3 process and knowledge evolution $RB-MOEA_{REF}$ alternatively. It has more time points for switching. When switching from NSGA3 to knowledge evolution, the same method as above is used. When switching back, we just use the solutions in the archive of knowledge evolution as the initial population of NSGA3.

5.4.4 Rule Initialization at Switching Generation

The initialization of rules at switching point is based on k-means clustering. Two examples are shown in Figure 5.6 with balanced or unbalanced size of clustering. When using balanced k-means, one issue is the convergence of clustering. We set the maximum number of iterations for identifying stable centroids is 100.



(a) Unbalanced Clustering



(b) Balanced Clustering

Figure 5.6: Rule Initialization based on k-Means Clustering

5.5 Experimental Setup

5.5.1 Algorithm Setups

Overall, we have four hybrid setups for knowledge based evolution in many-objective optimization: Hybrid $RB-MOEA_{REF}$ with single Halfway switch setup using Unbalanced k-means clustering (HHU), Hybrid $RB-MOEA_{REF}$ with single Halfway switch using Balanced k-means clustering (HHB), Hybrid $RB-MOEA_{REF}$ with Alternating switch setup using Unbalanced k-means clustering (HAU) and Hybrid $RB-MOEA_{REF}$ with Alternating switch setup using Balanced k-Means clustering (HAB). The experiments compare these four hybrid designs to the original NSGA3 algorithm.

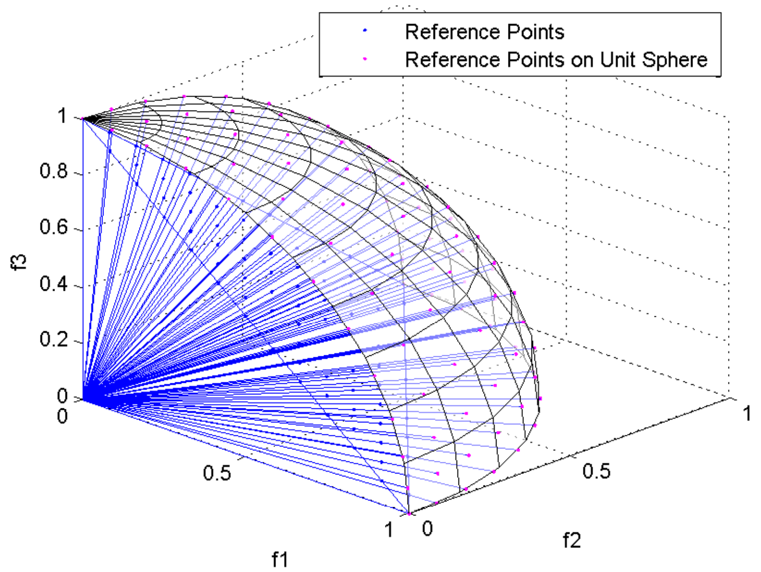


Figure 5.7: Projection of Reference Points to Unit Sphere.

5.5.2 Test Problems

We use DTLZ test problems to evaluate the performance of algorithms, which are scaled to 3, 5, 8, 10 and 15 objectives respectively in the experiments.

5.5.3 Performance Metric

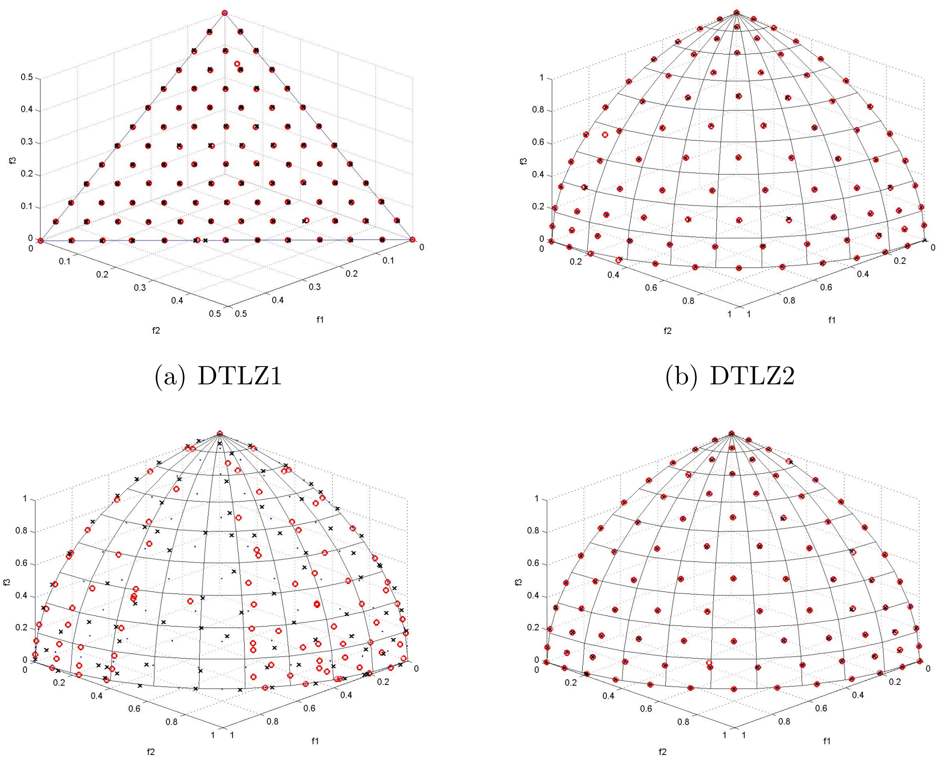
The performance of algorithms are analysed with respect to the objective space and decision space. For the objective space, the IGD metric is used to measure the outcome of the corresponding algorithm. When calculating IGD, a set of optimal solutions are required. For DTLZ 2-4 problems, the Pareto optimal solutions are all located on the unit sphere in the first octant. In order to get an even spread solutions on unit sphere, we just project the reference points to the unit sphere to get a sample of optimal solutions, as shown in Figure 5.7.

5.5.4 Parameter Settings

Since the reference points used in NSGA3 are generated structurally, we use the same settings as in its official paper for our experiments for clear comparison, as shown in Table 5.2. The parameters m, p_B, p_I, n_{REF}, n_P and N stand for number of objectives, number of divisions (Boundary Layer), number of divisions (Inside Layer), number of reference points, NSGA3 population size and rule population size, respectively. The number of solutions sampled from each rule S_R is 10.

Table 5.2: Number of Reference Points and Population Size Settings for NSGA3 and Hybrid $RB-MOEA_{REF}$ (m: number of objectives; p_B: number of divisions on Boundary Layer; p_I: number of divisions on Inside Layer; n_{REF}: number of reference points; n_P: NSGA3 population size; N: rule population size.)

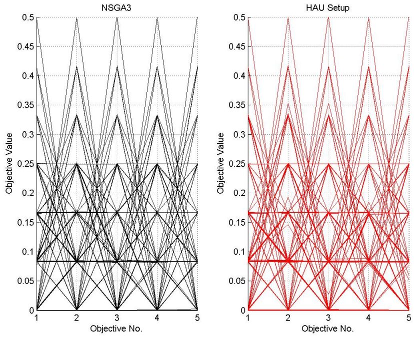
m	p_B	p_I	n_{REF}	n_P	N
3	12	-	91	92	10
5	6	-	210	212	22
8	3	2	156	156	16
10	3	2	275	276	28
15	2	1	135	136	14



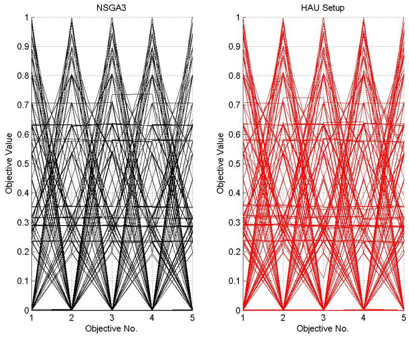
(a) DTLZ1 (b) DTLZ2

(c) DTLZ3 (d) DTLZ4

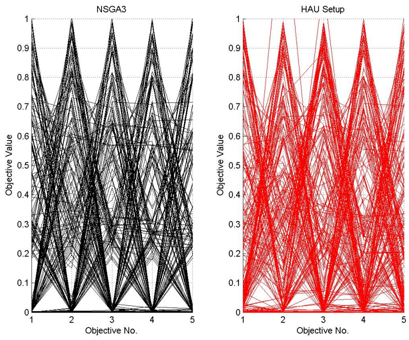
Figure 5.8: Obtained Front of NSGA3 in Black and HAU Setup in Red for DTLZ problems with 3 Objectives



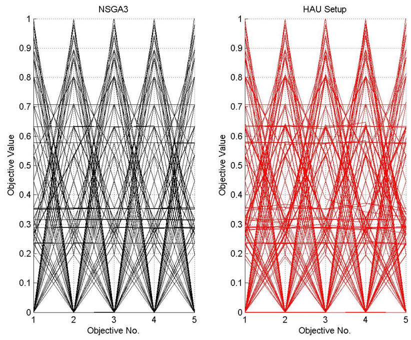
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



(d) DTLZ4

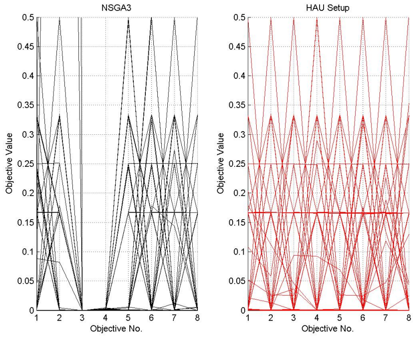
Figure 5.9: Obtained Front of NSGA3 in Black and HAU Setup in Red for DTLZ problems with 5 Objectives

5.6 Experiment Result

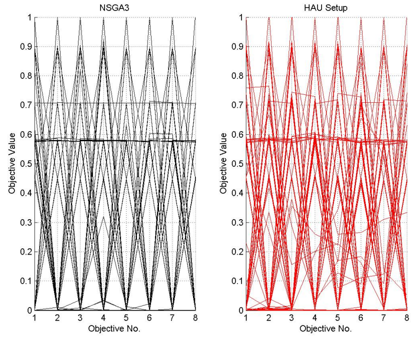
This section presents the performance analysis of the hybrid design for knowledge evolution in many-objective optimization, compared to the original NSGA3, in both objective space and decision space. The objective space performance is discussed in Section 5.6.1 covering the visualization of obtained objective vectors and performance measurement using IGD. The decision space performance, is analysed in Section 5.6.2 presenting the rules returned by knowledge evolution and result comparison with the PSV metric, proposed in Chapter 3.

5.6.1 Objective Space

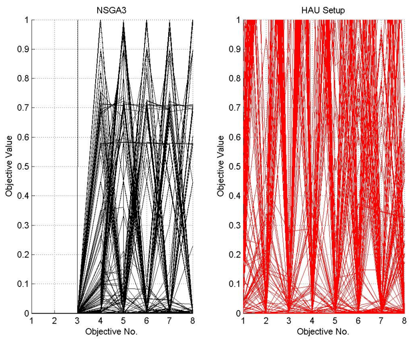
The objective space performance analysis focus on the convergence to the Pareto front and diversity along this front. In objective space, the obtained fronts with 3



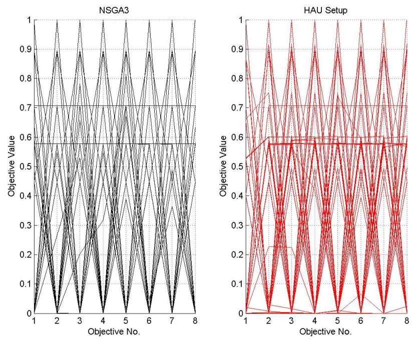
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



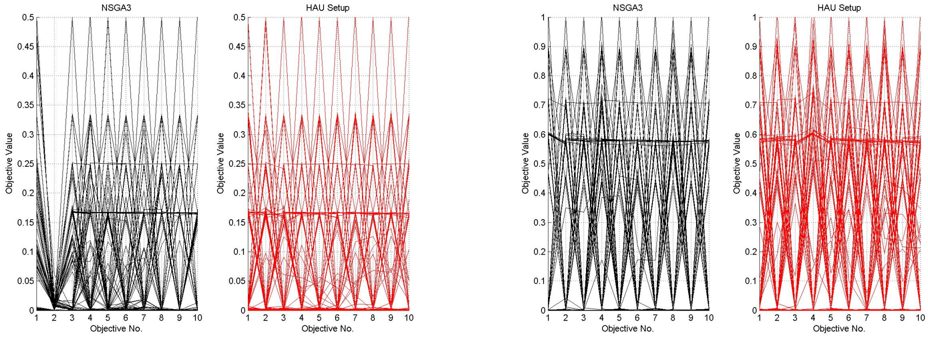
(d) DTLZ4

Figure 5.10: Obtained Front of NSGA3 in Black and HAU Setup in Red for DTLZ problems with 8 Objectives

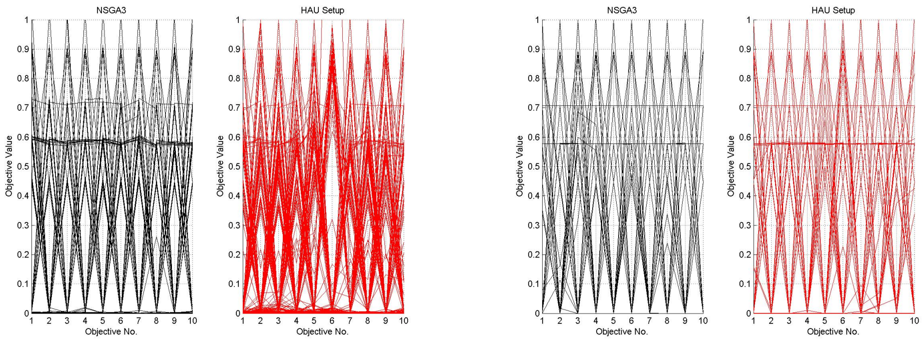
objectives are first visualized in Section 5.6.1.1 and for problems with 5, 8, 10 and 15 objectives, the parallel coordinates plot is used. The parallel coordinates plot is widely used in many-objective documents and is a straight-forward multivariate alternative that display all the variables together, allowing to investigate higher-dimensional relationships among variables. In this plot, the coordinate axes are all laid out horizontally, instead of using orthogonal axes as in the usual Cartesian graph. Each observation is represented in the plot as a series of connected line segments. The quantitative performance measure and comparison using IGD are presented in Section 5.6.1.2.

5.6.1.1 The distribution of Objective Optimal Vectors

For DTLZ problems with 3 objectives, the distribution of solutions in objective space is shown in Figure 5.8. Among these four subproblems, the 3-objective DTLZ3



(a) DTLZ1 (b) DTLZ2



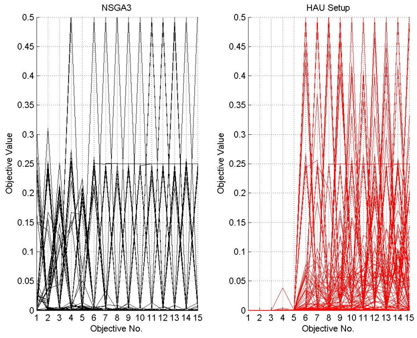
(c) DTLZ3 (d) DTLZ4

Figure 5.11: Obtained Front of NSGA3 in Black and HAU Setup in Red for DTLZ problems with 10 Objectives

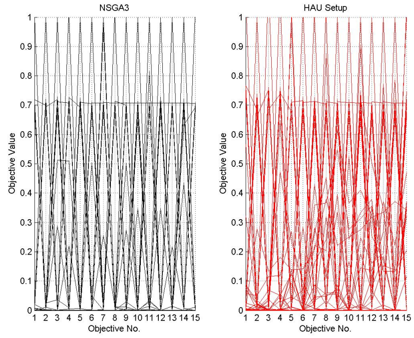
(DTLZ3F3) seems to be the most difficult one compared to other problems. In Figure 5.8(c), the objectives vectors successfully converged to the unit sphere, but not evenly spread compared to solutions in other problems.

For DTLZ problems with 5, 8, 10 and 15 objectives, the objective vectors of original NSGA3 and HAU hybrid design are presented in Figure 5.9, 5.10, 5.11 and 5.12 respectively. For most problems, both algorithms can return well distributed optimal solutions. For DTLZ3 problems, however, outliers can be observed in Figure 5.9(c), 5.10(c) and 5.11(c). Biased distribution exists in DTLZ1 and DTLZ3 with 15 objectives, as shown in Figure 5.12(a) and 5.12(c).

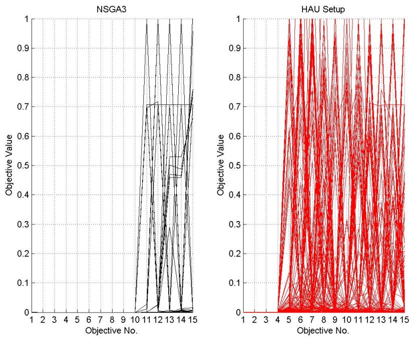
Overall, both the NSGA3 and the hybrid design are successfully providing satisfying outcomes with respect to increase in the number of objectives. A quantitative analysis is performed in the following section.



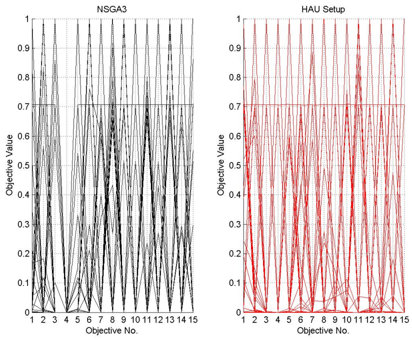
(a) DTLZ1



(b) DTLZ2



(c) DTLZ3



(d) DTLZ4

Figure 5.12: Obtained Front of NSGA3 in Black and HAU Setup in Red for DTLZ problems with 15 Objectives

5.6.1.2 IGD

IGD evaluates both convergence and diversity in objective space. The lower IGD values are considered better. Since the reference points in the calculation is determined by the reference points used in the algorithm, IGD here evaluates the convergence and distribution of solutions to the projection of reference points on the unit hyper-plane in DTLZ1 or unit hyper-sphere in other problems. The average IGD values (μ) and their standard deviation (σ) of original NSGA3 and four hybrid setups over DTLZ problem with different number of objectives are shown in Table 5.3. The value in bold means the best in this row and a ♦ means it's significantly better than the original NSGA3 using a Wilcoxon signed-rank test at the default 95% significance level. The change of IGD values over time (number of generation) is shown in Figure 5.13.

Regarding the performance of the algorithms in terms of this metric, it can be observed that in most cases, the best outcomes belong to the hybrid approaches. The original NSGA3 outperforms the four hybrid knowledge based evolution setups in 6 subproblems while in the rest 14 problems it fails to compete with the best value of hybrid approaches. When compared original NSGA3 to single hybrid setup, it outperforms with HAU, HAB, HHU and HHB setups with 7, 8, 11 and 12 subproblems. Its performance just sits between the alternating approaches and halfway-switching approaches.

Among the hybrid setups themselves, the alternating setups are slightly better than the halfway-switching approaches with respect to IGD. The alternating setups enjoys the both benefit from original NSGA3 evolution and knowledge-based evolution for convergence and diversity.

All the IGD curves over time or number of generations converge nicely as the number of generations increase.

Generally, for the objective space analysis, the hybrid knowledge based NSGA3 evolution shows comparable performance to the original NSGA3 in terms of IGD metric.

Table 5.3: Average IGD and Standard Deviation at Last Generation with ♦ Showing Statistical Significance at 0.95 Level Using Wilcoxon Test

Prob	m	Gen		NSGA3	HAU	HAB	HHU	HHB
DTLZ1	3	600	μ	1.785e-03	1.277e-03	**8.916e-04♦**	6.109e-03	2.265e-03
			σ	3.833e-03	3.021e-03	7.224e-04	2.547e-02	3.428e-03
	5	1000	μ	**9.632e-04**	4.317e-03	3.431e-03	5.251e-03	3.123e-03
			σ	6.910e-04	7.126e-03	4.440e-03	3.891e-03	1.512e-03
	8	1600	μ	1.409e-01	1.443e-02	**4.341e-03♦**	5.319e-02	4.121e-02
			σ	9.161e-02	2.720e-02	4.469e-03	2.763e-02	2.199e-02
	10	2000	μ	1.518e-01	**3.667e-03♦**	4.630e-03	6.372e-02	6.064e-02
			σ	7.952e-02	3.513e-03	4.723e-03	3.517e-02	3.734e-02
	15	3000	μ	3.078e-01	9.949e-02	**1.745e-02♦**	1.700e-01	1.417e-01
			σ	6.790e-02	4.650e-02	2.328e-02	4.117e-02	3.335e-02
DTLZ2	3	600	μ	**1.312e-03**	2.688e-03	2.427e-03	4.671e-03	3.874e-03
			σ	8.779e-04	1.291e-03	1.461e-03	3.538e-03	3.025e-03
	5	1000	μ	**3.537e-03**	7.440e-03	6.579e-03	8.252e-03	8.013e-03
			σ	1.530e-03	1.933e-03	9.110e-04	1.759e-03	2.894e-03
	8	1600	μ	3.997e-02	1.391e-02	**1.177e-02♦**	1.467e-02	1.627e-02
			σ	3.822e-02	5.413e-03	2.692e-03	4.375e-03	4.273e-03
	10	2000	μ	4.610e-02	2.411e-02	**1.810e-02♦**	4.021e-02	3.154e-02
			σ	3.151e-02	1.137e-02	5.253e-03	3.502e-02	2.682e-02
	15	3000	μ	1.701e-01	1.061e-01	**4.509e-02♦**	1.686e-01	2.277e-01
			σ	3.896e-02	3.464e-02	4.006e-02	4.570e-02	4.091e-02
DTLZ3	3	600	μ	**3.117e-02**	1.310e-01	2.753e-01	6.623e-01	5.601e-01
			σ	1.752e-02	1.780e-01	3.884e-01	6.422e-01	6.533e-01
	5	1000	μ	**5.970e-02**	1.890e-01	1.521e-01	1.555e-01	1.519e-01
			σ	5.097e-02	5.333e-02	4.541e-02	3.981e-02	1.228e-01
	8	1600	μ	3.408e-01	3.761e-01	6.372e-01	4.688e-01	**3.199e-01♦**
			σ	2.427e-01	3.317e-01	1.025e+00	5.841e-01	1.031e-01
	10	2000	μ	5.670e-01	2.479e-01	**1.637e-01♦**	4.022e-01	3.909e-01
			σ	1.517e-01	1.237e-01	9.159e-02	1.509e-01	1.366e-01
	15	3000	μ	9.734e-01	4.472e-01	**3.883e-01♦**	6.977e-01	7.240e-01
			σ	7.427e-02	2.739e-01	2.409e-01	1.217e-01	8.738e-02
DTLZ4	3	600	μ	5.969e-02	2.706e-02	**2.292e-02♦**	6.241e-02	7.315e-02
			σ	7.914e-02	2.748e-02	1.511e-02	9.944e-02	1.207e-01
	5	1000	μ	1.350e-02	**8.298e-03♦**	1.086e-02	1.712e-02	1.480e-02
			σ	1.132e-02	1.508e-02	1.416e-02	1.501e-02	1.697e-02
	8	1600	μ	7.845e-03	**3.036e-03♦**	3.536e-03	2.499e-02	1.186e-02
			σ	2.789e-02	4.749e-03	8.012e-03	3.968e-02	2.571e-02
	10	2000	μ	**8.134e-04**	1.101e-03	1.020e-03	1.392e-03	1.322e-03
			σ	1.752e-03	1.294e-04	1.401e-04	1.817e-04	2.917e-04
	15	3000	μ	2.367e-02	1.485e-02	**3.746e-03♦**	1.801e-02	2.396e-02
			σ	3.151e-02	3.086e-02	1.543e-02	3.258e-02	2.453e-02

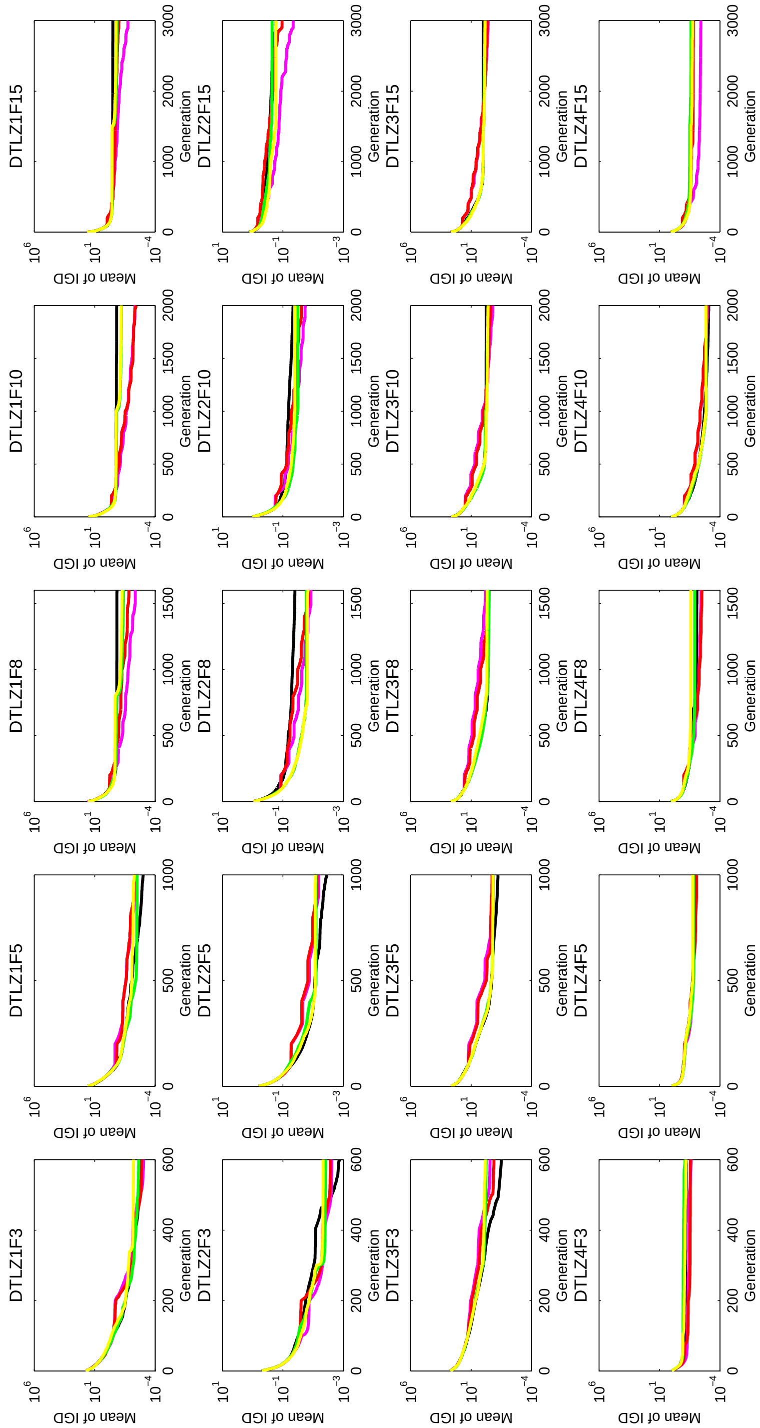


Figure 5.13: Variation of Mean IGD Value Over Time (Number of Generations) for DTLZ Problems with 3, 5, 8, 10 and 15 Objectives, NSGA3 Result in Black, HAU in Red, HAB in Cyan, HHU in Yello and HAB in Green

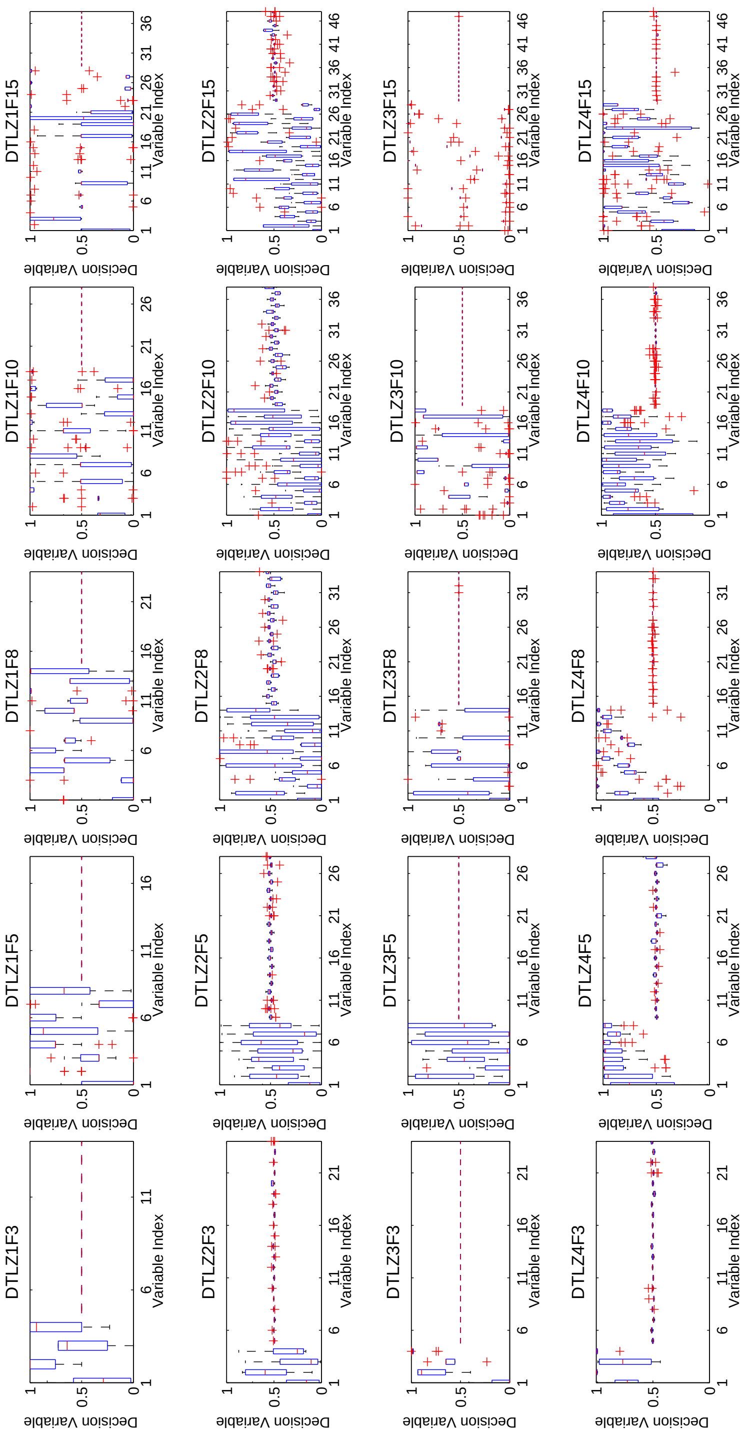


Figure 5.14: The Rule Population of HAU of First Run for DTLZ Problems with 3, 5, 8, 10 and 15 Objectives.

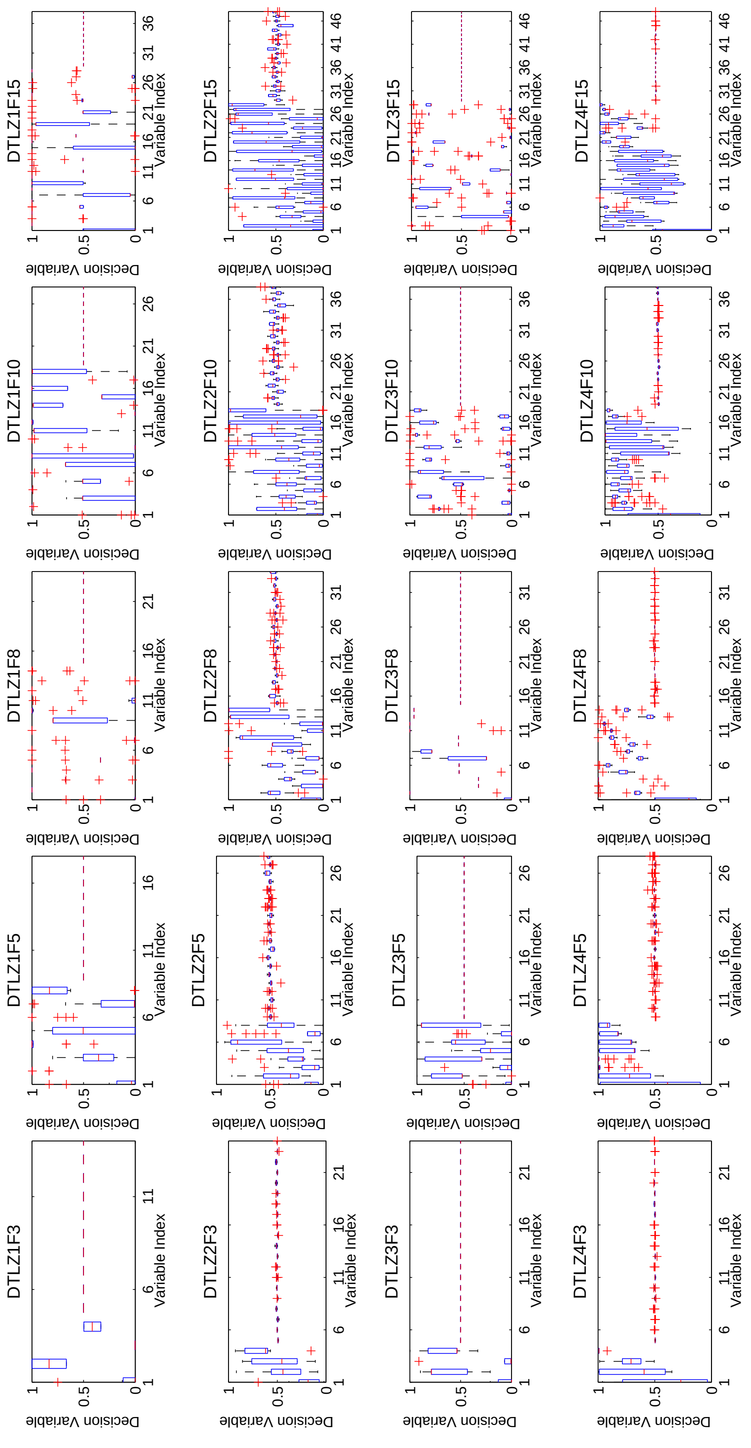


Figure 5.15: The Rule Population of HAB of First Run for DTLZ Problems with 3, 5, 8, 10 and 15 Objectives.

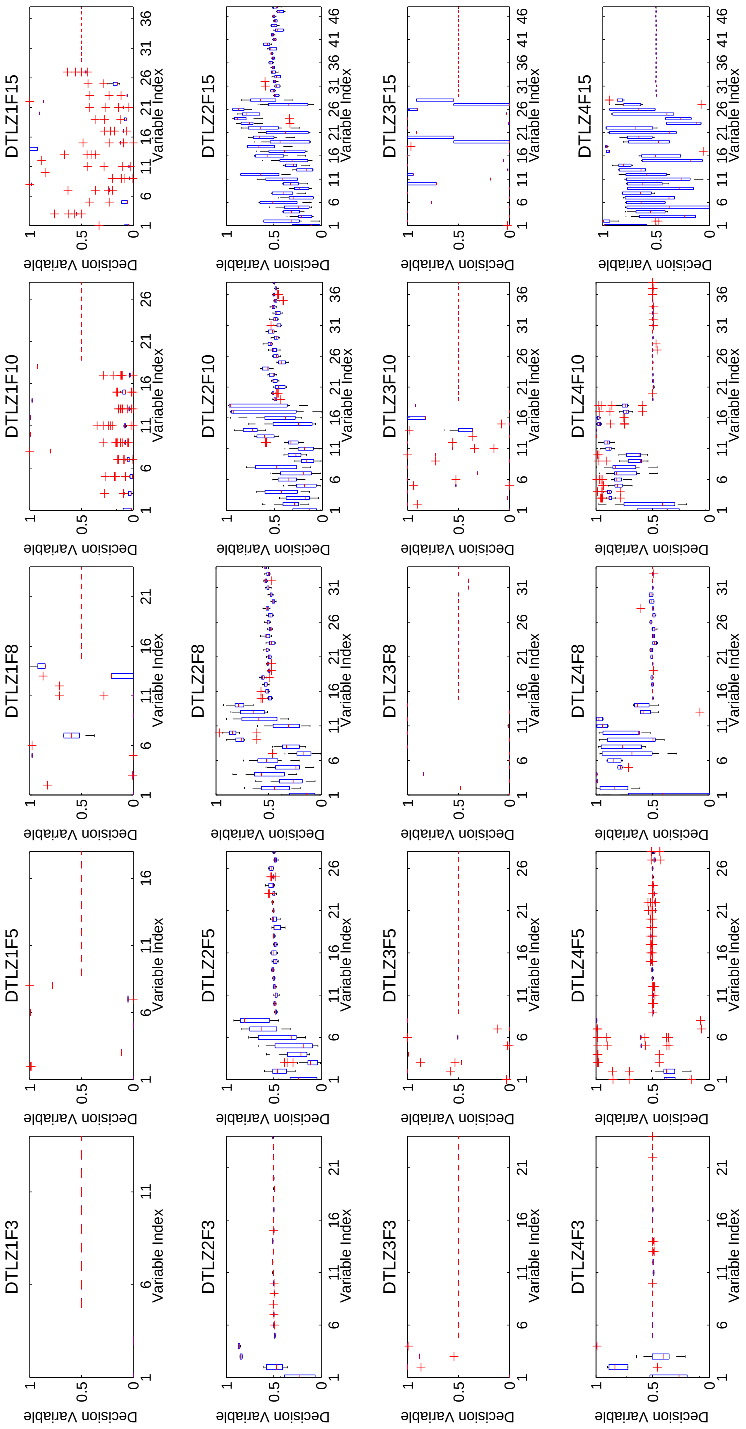


Figure 5.16: The Rule Population of HHU of First Run for DTLZ Problems with 3, 5, 8, 10 and 15 Objectives.

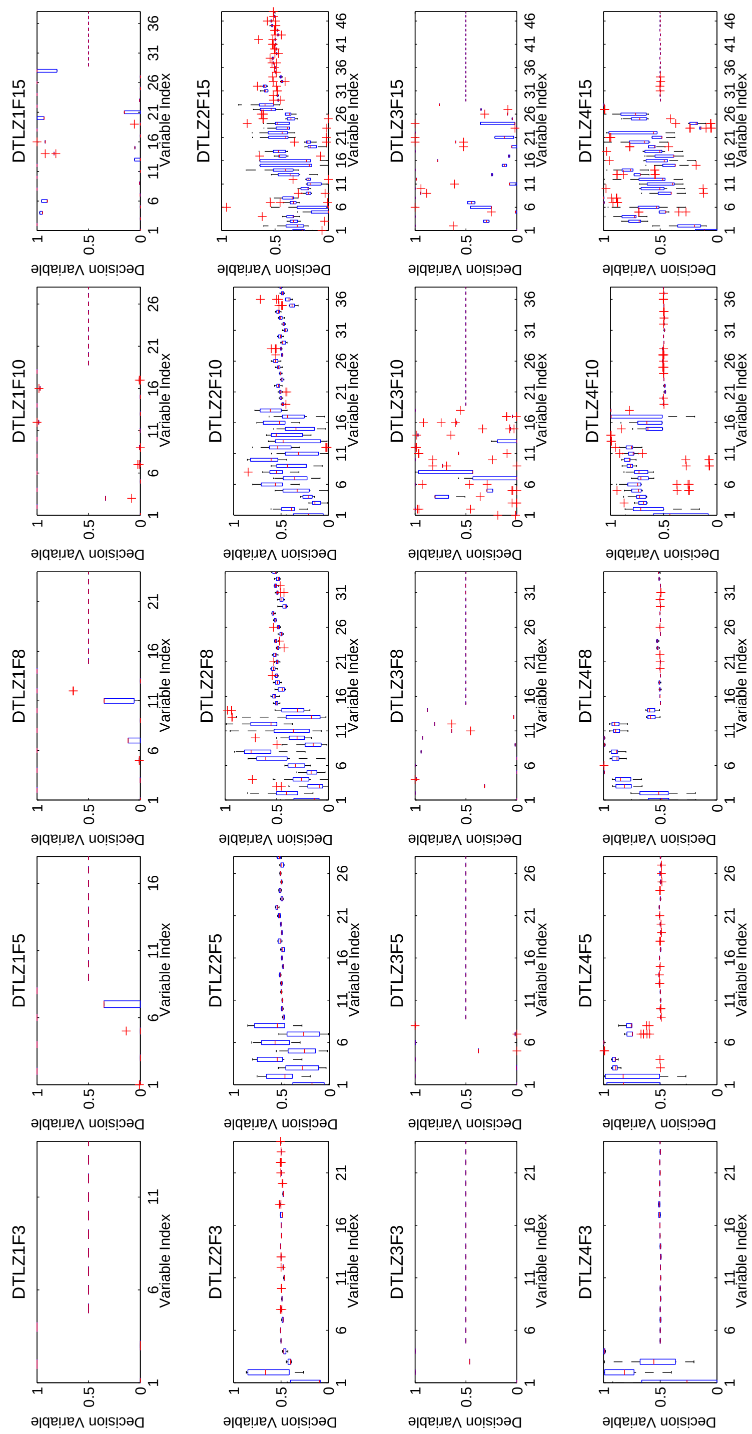


Figure 5.17: The Rule Population of HHB of First Run for DTLZ Problems with 3, 5, 8, 10 and 15 Objectives.

Table 5.4: Solution Average PSV and Standard Deviation at Last Generation with ⧫ Showing Statistical Significance at 0.95 Level Using Wilcoxon Test

Prob	m	Gen		NSGA3	HAU	HAB	HHU	HHB
DTLZ1	3	600	μ	9.563e+01	9.583e+01	9.564e+01	**9.612e+01**⧫	9.600e+01
			σ	4.372e-01	3.968e+01	4.554e+01	4.509e+01	4.120e+01
	5	1000	μ	6.190e+01	6.443e+01	6.471e+01	6.589e+01	**6.642e+01**⧫
			σ	3.699e+00	1.130e+01	1.843e+01	1.647e+01	1.241e+01
	8	1600	μ	3.309e+00	**1.040e+01**⧫	1.027e+01	5.812e+00	7.346e+00
			σ	4.310e+00	3.641e+00	3.837e+00	3.714e+00	2.790e+00
	10	2000	μ	6.074e+01	9.402e+01	9.448e+01	**9.615e+01**⧫	9.590e+01
			σ	2.837e+01	2.308e+01	2.521e+01	2.978e+01	3.170e+01
	15	3000	μ	2.785e+01	8.483e+01	**9.198e+01**⧫	8.958e+01	9.168e+01
			σ	1.641e+01	2.258e+01	2.092e+01	2.149e+01	1.630e+01
DTLZ2	3	600	μ	9.711e+01	9.710e+01	9.713e+01	9.724e+01	**9.729e+01**⧫
			σ	1.568e-02	3.130e+00	3.226e+00	3.628e+00	2.811e+00
	5	1000	μ	7.324e+01	7.264e+01	7.444e+01	**7.457e+01**⧫	7.445e+01
			σ	2.577e+00	7.467e+00	1.223e+01	1.347e+01	8.432e+00
	8	1600	μ	1.292e+01	1.228e+01	1.310e+01	1.282e+01	**1.401e+01**⧫
			σ	4.880e+00	2.964e+00	3.026e+00	2.429e+00	2.671e+00
	10	2000	μ	**9.507e+01**	9.473e+01	9.488e+01	9.355e+01	9.415e+01
			σ	4.739e-01	1.399e+01	1.967e+01	1.537e+01	1.070e+01
	15	3000	μ	8.363e+01	8.771e+01	**8.851e+01**⧫	8.398e+01	8.814e+01
			σ	1.414e+01	2.132e+01	1.633e+01	2.011e+01	2.202e+01
DTLZ3	3	600	μ	**9.592e+01**	9.424e+01	9.193e+01	8.346e+01	9.474e+01
			σ	1.290e+00	3.917e+01	4.304e+01	4.743e+01	4.490e+01
	5	1000	μ	**7.081e+01**	5.171e+01	6.137e+01	5.922e+01	6.142e+01
			σ	4.686e+00	2.326e+01	2.530e+01	2.571e+01	2.203e+01
	8	1600	μ	**9.523e+00**	4.484e+00	5.290e+00	9.492e+00	7.872e+00
			σ	7.049e+00	3.477e+00	3.534e+00	5.489e+00	4.329e+00
	10	2000	μ	4.916e+01	9.412e+01	**9.571e+01**⧫	9.221e+01	9.535e+01
			σ	2.809e+01	3.610e+01	3.227e+01	2.703e+01	2.403e+01
	15	3000	μ	1.146e+01	8.477e+01	8.076e+01	**8.489e+01**⧫	7.890e+01
			σ	5.631e+00	2.663e+01	2.860e+01	1.439e+01	1.513e+01
DTLZ4	3	600	μ	1.295e+01	**3.568e+01**⧫	3.012e+01	3.156e+01	2.800e+01
			σ	9.745e+00	2.393e+01	2.086e+01	2.339e+01	2.102e+01
	5	1000	μ	1.290e+01	2.176e+01	**2.716e+01**⧫	1.717e+01	1.501e+01
			σ	9.469e+00	1.201e+01	1.286e+01	1.044e+01	8.456e+00
	8	1600	μ	1.693e+00	**3.582e+00**⧫	2.434e+00	3.478e+00	3.187e+00
			σ	1.552e+00	3.481e+00	1.479e+00	3.641e+00	2.162e+00
	10	2000	μ	6.940e+01	**7.974e+01**⧫	7.926e+01	7.464e+01	7.782e+01
			σ	1.091e+01	7.896e+00	2.167e+01	1.453e+01	1.693e+01
	15	3000	μ	6.009e+01	6.942e+01	**7.547e+01**⧫	7.298e+01	7.291e+01
			σ	3.731e+00	9.337e+00	8.896e+00	1.035e+01	1.211e+01

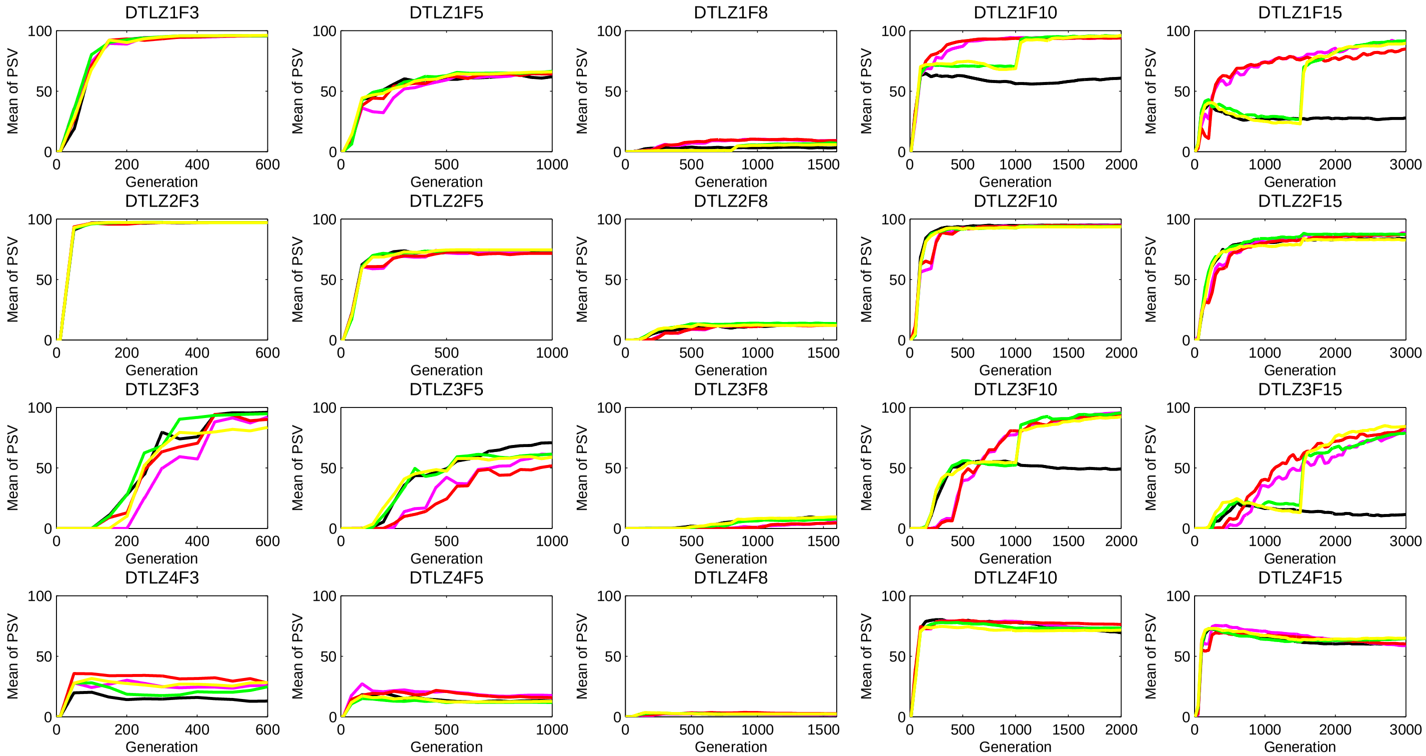


Figure 5.18: Variation of Mean PSV Value Over Time (Number of Generations) for DTLZ Problems with 3, 5, 8, 10 and 15 Objectives, NSGA3 Result in Black, HAU in Red, HAB in Cyan, HHU in Yello and HAB in Green

5.6.2 Decision Space

The hybrid design of knowledge based many-objective optimization evolves rules in decision space in some periods of the optimization process expecting to identify interesting distribution patterns of Pareto sets. Here, we first represent the rules that the hybrid approach returned in Section 5.6.2.1 and then evaluate the performance using PSV metric in Section 5.6.2.2.

5.6.2.1 Rules

When doing the knowledge based evolution in the hybrid setups, the algorithm maintains an archive of high quality non-dominated solutions as well as a population of rules evolving to cover the optimal patterns in decision space. The box plots of the rule population of the first run are shown in Figure 5.14, Figure 5.15, Figure 5.16 and Figure 5.17 with respect to different problem specification.

It is observed that DTLZ2 problems are most difficult for the rule to converge. For the two groups of decision variables of DTLZ test suite, the first $m-1$ variables are used to determine the shape of the tradeoff surface and the rest $n-m+1$ variables are used to determine the location of the Pareto front. In DTLZ2 problems, the second group of variables are close to the optimal value but fluctuate around $x_i = 0.5, i = m, m+1, \dots, n$. For the rest DTLZ problems, most rules successfully converge to the optimal area. And in some subproblems, especially for those with 3 or 5 objectives, the full Pareto set or most of the optimal area can be identified with rules.

Among the four different setups, the two halfway-switching approaches (checking the rule quality of HHU and HHB in Figure 5.16 and Figure 5.17) are comparatively better than the alternating setups (HAU and HAB in Figure 5.14 and Figure 5.15). HHU and HHB setups allow more generations for the rules to evolve consistently whereas the other two setups will hinder the evolution of rules since it switches every 100 generations.

Overall, all the hybrid approaches can successfully provide a set of high quality

Table 5.5: Rule Average PSV and Standard Deviation at Last Generation with ♦ Showing Statistical Significance at 0.95 Level Using Wilcoxon Test

Prob	m	Gen		NSGA3	HAU	HAB	HHU	HHB
DTLZ1	3	600	μ	9.563e+01	8.111e+01	8.044e+01	8.627e+01	**9.851e+01**♦
			σ	4.372e-01	1.483e+01	1.907e+01	3.001e+01	2.291e+00
	5	1000	μ	6.190e+01	5.563e+01	8.296e+01	8.506e+01	**9.503e+01**♦
			σ	3.699e+00	2.231e+01	6.684e+00	1.483e+01	7.363e+00
	8	1600	μ	3.309e+00	2.555e+01	5.634e+01	5.978e+01	**7.438e+01**♦
			σ	4.310e+00	1.388e+01	1.495e+01	2.671e+01	1.941e+01
	10	2000	μ	6.074e+01	9.117e+01	9.257e+01	9.628e+01	**9.825e+01**♦
			σ	2.837e+01	3.925e+00	4.886e+00	3.402e+00	1.507e+00
	15	3000	μ	2.785e+01	7.312e+01	8.463e+01	8.351e+01	**8.591e+01**♦
			σ	1.641e+01	1.864e+01	7.186e+00	1.049e+01	9.564e+00
DTLZ2	3	600	μ	**9.711e+01**	5.371e+01	5.135e+01	1.529e+01	1.845e+01
			σ	1.568e-02	9.756e+00	8.095e+00	1.264e+01	1.120e+01
	5	1000	μ	**7.324e+01**	3.054e+01	3.827e+01	6.754e+00	1.452e+01
			σ	2.577e+00	1.182e+01	8.739e+00	3.487e+00	1.207e+01
	8	1600	μ	1.292e+01	5.676e+00	**1.309e+01**♦	4.097e-01	2.829e+00
			σ	4.880e+00	3.793e+00	1.202e+01	4.696e-01	2.107e+00
	10	2000	μ	**9.507e+01**	7.030e+01	6.311e+01	1.259e+01	3.308e+01
			σ	4.739e-01	9.614e+00	1.370e+01	1.432e+01	1.880e+01
	15	3000	μ	**8.363e+01**	5.602e+01	5.482e+01	1.326e+01	2.138e+01
			σ	1.414e+01	1.505e+01	1.640e+01	1.560e+01	1.744e+01
DTLZ3	3	600	μ	**9.592e+01**	6.030e+01	5.700e+01	1.477e+01	5.409e+01
			σ	1.290e+00	2.587e+01	3.271e+01	1.990e+01	3.705e+01
	5	1000	μ	7.081e+01	4.778e+01	7.189e+01	7.089e+01	**8.287e+01**♦
			σ	4.686e+00	1.388e+01	1.398e+01	1.632e+01	1.678e+01
	8	1600	μ	9.523e+00	1.169e+00	1.228e+01	3.560e+01	**3.939e+01**♦
			σ	7.049e+00	2.272e+00	2.559e+01	2.534e+01	3.132e+01
	10	2000	μ	4.916e+01	8.545e+01	**9.301e+01**♦	8.080e+01	8.568e+01
			σ	2.809e+01	1.031e+01	3.746e+00	1.279e+01	7.766e+00
	15	3000	μ	1.146e+01	**7.311e+01**♦	6.844e+01	6.910e+01	6.837e+01
			σ	5.631e+00	1.819e+01	1.898e+01	2.005e+01	1.481e+01
DTLZ4	3	600	μ	1.295e+01	3.581e+01	3.779e+01	5.465e+01	**5.969e+01**♦
			σ	9.745e+00	1.067e+01	1.125e+01	7.072e+00	1.117e+01
	5	1000	μ	1.290e+01	1.859e+01	1.542e+01	**3.545e+01**♦	3.407e+01
			σ	9.469e+00	6.152e+00	3.830e+00	4.271e+00	6.908e+00
	8	1600	μ	1.693e+00	2.038e+00	1.898e+00	7.905e+00	**8.602e+00**♦
			σ	1.552e+00	8.312e-01	6.885e-01	1.911e+00	1.616e+00
	10	2000	μ	6.940e+01	5.774e+01	5.486e+01	7.094e+01	**7.145e+01**♦
			σ	1.091e+01	4.384e+00	4.178e+00	3.236e+00	2.750e+00
	15	3000	μ	6.009e+01	5.438e+01	4.732e+01	**7.053e+01**♦	6.472e+01
			σ	3.731e+00	9.157e+00	7.331e+00	5.722e+00	4.191e+00

rules for decision making from 3-objective to many-objective optimization.

5.6.2.2 PSV

The PSV evaluates the percentage of the area of convex hull determined by solutions over the overall area of optimal patterns. Higher value is preferred. The quickhull algorithm is utilized for its calculation. However, when estimating the PSV values in many-objective environments, one problem is the determination of high dimensional convex hull. For DTLZ problems with 10 and 15 objectives, we have to identify the convex hull in 9 and 14 dimensional decision spaces respectively while the quickhull algorithm can only handle up to 8 dimensions. In this case, we just pick any 2 dimensions and calculate the PSV value with 2-dimensional convex hull and then take the average as the final metric value. For example, when dealing with DTLZ1 with 10 objectives, there are $\binom{9}{2} = 36$ combinations if we choose two dimensions from the corresponding decision space. We treat these combinations individually and use the average of them as the final PSV value for the problem.

The PSV values of original NSGA3 and four hybrid setups over DTLZ problems with different number of objectives are shown in Table 5.4. The PSV values determined by the vertices of corresponding rules are listed in Table 5.5. The value in bold means the best in this row and a ⧫ means it's significantly better than the original NSGA3 using a Wilcoxon signed-rank test at the default 95% significance level. The change of PSV over time (number of generations) is shown in Figure 5.18.

When evaluating the performance in terms of this metric in Table 5.4, the hybrid approaches generate the best results in most cases, except in 4 subproblems. The hybrid design utilizes NSGA3 operation for convergence and help the original NSGA3 with improved diversity in decision space significantly.

When comparing the result of solution archives and the rules in Table 5.4 and 5.5, the rules can outperform the solution archive in most subproblems of DTLZ1, DTLZ3 and DTLZ4. The HHB hybrid setup provides the best performance comparatively. However, in DTLZ2, the rules suffered from the convergence as explained in Section 5.6.2.1 and the result is not as good as the solution archive.

Overall, the hybrid design of knowledge based many-objective optimization can improve the original NSGA3 in performance regarding decision space in terms of PSV metric.

5.7 Summary

NSGA3 expands the application scope of Pareto dominance based evolutionary algorithm to many objective space. It tries to solve the convergence and diversity with the support of a set of reference points spread evenly in the objective space, although the representation of the whole Pareto surface is highly constrained simply by the distribution of these reference points.

In order to introduce knowledge based optimization into a many-objective environment, a hybrid design is adopted. This approach utilizes the NSGA3 operation to promise the convergence and combined rule based evolution to generate optimal patterns to describe the Pareto set. Four different setups are described in this chapter and their performance are analyzed with problems having up to 15 objectives.

Generally, the hybrid approaches can benefit many-objective optimization problems with comparable solution quality in objective space as well as improved performance in decision space. The hybrid design is able to return a set of high quality rules to describe the distribution of the Pareto set and provides flexibility for decision making compared to a limited number of isolated optimal solutions.

Chapter 6

Conclusion and Future Work

Online Knowledge-based Evolutionary Multi-objective Optimization is a challenging task. It transforms the traditional solution based evolution to knowledge based optimization to improve the performance of algorithms in both decision space and objective space. It requires effective algorithmic designs for the evolution in multi-objective optimization context, including many-objective problems and an appropriate evaluation methodology for judging on performance. In this thesis, we attempted to address the evaluation benchmarks and metrics first and then focused on algorithmic design of knowledge-based evolutionary multi-objective optimization. A summary of contributions in this thesis is provided below.

6.1 Summary of Contributions

The contributions of this thesis focus on online knowledge-based evolutionary multi-objective optimization design and the development of evaluation benchmarks in the decision space covering the following points:

- This thesis presents a novel knowledge-based multi-objective optimization framework which automatically searches for optimal patterns in the decision space by evolving directly a population of rules. The rules correspond to the bounding areas in the design space and are evaluated based on the quality of sampled solutions from their bounded area. The framework allows using the existing MOEA design for the evaluation of rule quality. This facilitates the online discovery of knowledge during the optimization process in an interpretable form.

- An implementation of the framework using a hyperrectangular rule representation and evaluating rules based on NSGA2 for optimization with multiple objectives are presented.

- An implementation of the framework using a hybrid design combined with NSGA3 for many objectives optimization is presented.

- A mechanism is proposed to generate test problems where those solutions belonging to the efficient or Pareto frontier are mapped to defined hyperrectangular patterns in the decision space. The modular design allows the generation of a number of test problem instances (HPS) with varying degrees of complexity in terms of number of decision variables, number of optimal patterns and other interesting features.

- A new metric is proposed to evaluate the performance of MOEAs in the decision space. The metric (PSV) relies on computing the ratio of volume covered by the solution set obtained by an MOEA to the total volume occupied by the defined Pareto sets in the decision space.

The effectiveness of the resulting algorithms is established through comprehensive experimentation and analysis. The algorithms are tested on some standard benchmark problems as well as newly proposed test functions. The performance was analysed using existing metrics and newly designed metrics. The experimental results demonstrate the optimization capabilities of the proposed algorithms in comparison to the state-of-the-art multi-objective evolutionary optimization algorithms, such as NSGA2, NSGA3 and MOEA/D. The main findings of this thesis are summarized below as follows:

- When compared to state-of-the-art MOEAs, the knowledge-based algorithms can generate comparable performance regarding the solution quality in the objective space, such as convergence and diversity issues. In most cases, the knowledge-based computational methods are able to perform equally or better than the mainstream algorithms, although no obvious statistical differences are found.

- The proposed algorithms demonstrate considerable improvement in performance in the decision space, including rules of high quality to support decision making and better solution diversity and coverage over the whole Pareto set. Firstly, the knowledge-based algorithm is able to capture the high fitness areas in decision space with minimum number of rules. The rules can be represented in natural and standard if-then forms which can further facilitate understanding and decision making in multi-objective context. Secondly, the diversity and coverage of solutions over the whole Pareto set in decision space is significantly improved. As shown in Chapter 4, the knowledge-based algorithm outperforms competing algorithms for all HPS problems in terms of PSV metric, especially for those with more than 4 patterns. The rise in performance can only be matched when the order of magnitude of the population size is increased for competing algorithms.

- The new test problems and metric proposed in this thesis can be used to benchmark the performance of MOEAs from the decision space's perspective, contributing to the scant research in performance evaluation of MOEAs in the decision space.

The thesis reveals that performance in the decision space is not necessarily reflecting performance in the objective space. While a MOEA performs comparatively good in the objective space, its performance in the decision space is not guaranteed. In fact, the diversity or distribution patterns of optimal solutions in the decision space enable us to realize and utilize the causative side of the mapping from decisive optimal parameter settings to the resulting optimal front. When solving engineering design problems, it is the design parameter space to be tuned for the generation of satisfying performance in the objective space. From this perspective, the investigation over decision space is of important value to facilitate a comprehensive understanding of the original optimization problem.

Overall, we believe that the knowledge-based MOEA framework presented in this thesis has important implications for many domains including engineering design and decision support systems. It helps in understanding the optimization process better by revealing the relationships between the decision variables and the objective functions based on the rules or patterns that lead to optimal designs, hence for the design of robust optimizers. It is able to deal with a wide range of real world MOPs, presenting the Pareto set in the form of interpretable rules with promising diversity of solutions in the decision space and improvement in the implementation of solutions.

6.2 Limitations

There are also a number of limitations in our work.

Scalability of proposed methodology - First, the knowledge-based evolutionary algorithms for multi-objective optimization suffers from the cardinality and the dimensionality of optimal patterns in decision space. The proposed algorithms can generate dominating results compared to other methods, but the performance deteriorates when either the number or dimensionality of optimal patterns increases. Second, the calculation of convex hull based decision space performance metric, PSV, get more expensive as we increase the dimensionality of optimal patterns.

Comparison with other Knowledge Extraction Algorithms for Multi-objective optimization - The proposed knowledge-based evolution is compared against leading multi-objective evolutionary algorithms regarding performance in both objective space and decision space, without other knowledge extraction methods involved. However, it is difficult to do such analysis directly since the knowledge representation varies over different applications and there is no standard benchmarks serving this purpose, as shown in Chapter 2.

Analysis of Learnt Knowledge - The online knowledge-based evolutionary multi-objective optimization has demonstrated the effectiveness of the identification of optimal patterns in the form of rules, however, the question how these rules can be utilized to support decision making has not been researched. This is thesis is about the establishment of the knowledge-based optimization. We will address the application of rules in the future research.

Real World Applications - Online knowledge-based evolutionary multi-objective optimization is evaluated based on classic benchmarks and proposed test problems. This work needs to extend to real-world applications. In particular, the nature of the problem is more suitable for optimizing rule-based multi-agent systems, where the discovery of rules during the optimization process is a natural way to generalize the optimization process of the multi-agent system.

6.3 Future Work

Overall, this thesis is a first attempt to analyse and set up online knowledge-based evolutionary multi-objective optimization. There is a broad scope for extensions to the current research. Based on the limitations and discussions above, there are a number of directions stemming from this work for future research.

Scaling to bigger problems - Regarding limitations in the scalability of the method, the knowledge-based optimization requires new improvements to tackle more complex problems with many and high-dimensional optimal patterns more effectively. Dynamic resource allocation is probably needed to deal with the increase in dimensionality of Pareto set to apply more computations on inefficient dimensions.

Generalizing the implementation to other MOEAs - Regarding the instantiations of the framework, the establishment of its generality require extensions using a number of other leading MOEAs, not only the Pareto dominance based algorithms and extensions using a number of other rule representations including hyper-ellipsoids and fuzzy representations.

Extending the application scope of knowledge-based optimization - First, knowledge-based evolutionary multi-objective optimization requires real world problems to demonstrate its practical use. Second, knowledge-based evolution can be successful and needs to be testing in dynamic environments. The online design enjoys the capability to provide knowledge about the problem at any time during the evolution for dynamic optimization. Last, the hypervolume rule-represented knowledge can be transferred to be used in robust optimization without too much effort. Robust optimization problems can also be considered in the future to extend the scope of application.

To sum up, the topic of knowledge discovery for optimization problems is relatively new and this thesis is about the design of online knowledge based multi-objective optimization. This thesis does not solve all problems in this domain, but instead concentrates on the construction, implementation and establishment of a framework to provide an alternative approach compared to the existing methods. Improvement of the proposed framework will provide interesting insights for future research in the field of evolutionary multi-objective optimization.

Bibliography

[1] Hussein Abbass, Ruhul Sarker, Charles Newton, et al. PDE: a pareto-frontier differential evolution approach for multi-objective optimization problems. In *Proceedings of the 2001 Congress on Evolutionary Computation.*, volume 2, pages 971–978. IEEE, 2001.

[2] Hussein A Abbass. An inexpensive cognitive approach for bi-objective optimization using bliss points and interaction. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 712–721. Springer, 2004.

[3] Hussein A Abbass. An economical cognitive approach for bi-objective optimization using bliss points, visualization, and interaction. *Soft Computing*, 10(8):687–698, 2006.

[4] Hussein A Abbass. *Computational Red Teaming: Risk Analytics of Big-Data-to-Decisions Intelligent Systems*. Springer, 1st edition, 2014.

[5] Salem F Adra and Peter J Fleming. Diversity management in evolutionary many-objective optimization. *IEEE Transactions on Evolutionary Computation*, 15(2):183–195, 2011.

[6] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zitzler. Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. In *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms*, pages 87–102. ACM, 2009.

[7] Johannes Bader and Eckart Zitzler. Robustness in hypervolume-based multi-objective search. Technical report, Tech. Rep. TIK 317, Computer Engineering and Networks Laboratory, ETH Zurich, 2010.

[8] Johannes Bader and Eckart Zitzler. HypE: An algorithm for fast hypervolume-based many-objective optimization. *Evolutionary computation*, 19(1):45–76, 2011.

[9] Sunith Bandaru and Kalyanmoy Deb. Automated discovery of vital knowledge from pareto-optimal solutions: First results from engineering design. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[10] Sunith Bandaru and Kalyanmoy Deb. Automated innovization for simultaneous discovery of multiple rules in bi-objective problems. In *Evolutionary Multi-Criterion Optimization*, pages 1–15. Springer, 2011.

[11] Sunith Bandaru and Kalyanmoy Deb. Towards automating the discovery of certain innovative design principles through a clustering-based optimization technique. *Engineering optimization*, 43(9):911–941, 2011.

[12] Sunith Bandaru and Kalyanmoy Deb. Higher and lower-level knowledge discovery from pareto-optimal sets. *Journal of Global Optimization*, 57(2):281–298, 2013.

[13] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, 22(4):469–483, 1996.

[14] Matthieu Basseur. Design of cooperative algorithms for multi-objective optimization: application to the flow-shop scheduling problem. *4OR*, 4(3):255–258, 2006.

[15] Matthieu Basseur and Eckart Zitzler. Handling uncertainty in indicator-based multiobjective optimization. *International Journal of Computational Intelligence Research*, 2(3):255–272, 2006.

[16] To Thanh Binh and Ulrich Korn. An evolution strategy for the multiobjective optimization. *Institute for Automation, Magdeburg*, 1996.

[17] Jürgen Branke and Kalyanmoy Deb. Integrating user preferences into evolutionary multi-objective optimization. In *Knowledge incorporation in evolutionary computation*, pages 461–477. Springer, 2005.

[18] Jürgen Branke, Thomas Kaußler, and Harmut Schmeck. Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, 32(6):499–507, 2001.

[19] Dimo Brockhoff and Eckart Zitzler. Are all objectives necessary? on dimensionality reduction in evolutionary multiobjective optimization. In *Parallel Problem Solving from Nature-PPSN IX*, pages 533–542. Springer, 2006.

[20] Dimo Brockhoff and Eckart Zitzler. Improving hypervolume-based multiobjective evolutionary algorithms by using objective reduction methods. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 2086–2093. IEEE, 2007.

[21] Lam T Bui, Jing Liu, Axel Bender, Michael Barlow, Slawomir Wesolkowski, and Hussein A Abbass. DMEA: a direction-based multiobjective evolutionary algorithm. *Memetic Computing*, 3(4):271–285, 2011.

[22] Andrea Caponio, Giuseppe Leonardo Cascella, Ferrante Neri, Nadia Salvatore, and Mark Sumner. A fast adaptive memetic algorithm for online and offline control design of pmsm drives. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(1):28–41, 2007.

[23] Pei Chann Chang, Shih Hsin Chen, Qingfu Zhang, and Jun Lin Lin. Moea/d for flowshop scheduling problems. In *Evolutionary Computation, 2008. CEC*

2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on, pages 1433–1438. IEEE, 2008.

[24] Xianshun Chen, Yew-Soon Ong, Meng-Hiot Lim, and Kay Chen Tan. A multi-facet survey on memetic computation. *IEEE Transactions on Evolutionary Computation*, 15(5):591–607, 2011.

[25] Kazuhisa Chiba, Akira Oyama, Shigeru Obayashi, Kazuhiro Nakahashi, and Hiroyuki Morino. Multidisciplinary design optimization and data mining for transonic regional-jet wing. *Journal of Aircraft*, 44(4):1100–1112, 2007.

[26] Karim J Chichakly and Margaret J Eppstein. Discovering design principles from dominated solutions. *Access, IEEE*, 1:275–289, 2013.

[27] Carlos A Coello Coello. A comprehensive survey of evolutionary-based multiobjective optimization techniques. *Knowledge and Information systems*, 1(3):269–308, 1999.

[28] Carlos A Coello Coello. Evolutionary multi-objective optimization: a historical view of the field. *Computational Intelligence Magazine, IEEE*, 1(1):28–36, 2006.

[29] Carlos A Coello Coello and Gary B Lamont. *Applications of multi-objective evolutionary algorithms*, volume 1. World Scientific, 2004.

[30] Carlos A Coello Coello and Maximino Salazar Lechuga. Mopso: A proposal for multiple objective particle swarm optimization. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 2, pages 1051–1056. IEEE, 2002.

[31] Carlos A Coello Coello and Margarita Reyes Sierra. A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm. In *MICAI 2004: Advances in Artificial Intelligence*, pages 688–697. Springer, 2004.

[32] David W Corne, Nick R Jerram, Joshua D Knowles, Martin J Oates, et al. PESA-II: Region-based selection in evolutionary multiobjective optimization.

In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO2001*. Citeseer, 2001.

[33] David W Corne and Joshua D Knowles. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 773–780. ACM, 2007.

[34] David W Corne and Joshua D Knowles. Techniques for highly multiobjective optimisation: some nondominated points are better than others. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 773–780. ACM, 2007.

[35] David W Corne, Joshua D Knowles, and Martin J Oates. The pareto envelope-based selection algorithm for multiobjective optimization. In *Parallel Problem Solving from Nature PPSN VI*, pages 839–848. Springer, 2000.

[36] Indraneel Das and John E Dennis. Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems. *SIAM Journal on Optimization*, 8(3):631–657, 1998.

[37] Swagatam Das, Ajith Abraham, and Amit Konar. Automatic clustering using an improved differential evolution algorithm. *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*, 38(1):218–237, 2008.

[38] Kalyanmoy Deb. Unveiling innovative design principles by means of multiple conflicting objectives. *Engineering Optimization*, 35(5):445–470, 2003.

[39] Kalyanmoy Deb. Advances in evolutionary multi-objective optimization. In *Search Based Software Engineering*, pages 1–26. Springer, 2012.

[40] Kalyanmoy Deb and Ram B Agrawal. Simulated binary crossover for continuous search space. *Complex Systems*, 9(3):1–15, 1994.

[41] Kalyanmoy Deb, Samir Agrawal, Amrit Pratap, and Tanaka Meyarivan. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II. *Lecture notes in computer science*, 1917:849–858, 2000.

[42] Kalyanmoy Deb and Himanshu Jain. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part I: Solving problems with box constraints. 2013.

[43] Kalyanmoy Deb and Abhishek Kumar. Interactive evolutionary multi-objective optimization and decision-making using reference direction method. In *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, pages 781–788. ACM, 2007.

[44] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and TAMT Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 6(2):182–197, 2002.

[45] Kalyanmoy Deb and D Saxena. Searching for pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)*, pages 3352–3360, 2006.

[46] Kalyanmoy Deb and Dhish Kumar Saxena. On finding pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems. *Kangal report*, 2005011, 2005.

[47] Kalyanmoy Deb, Ankur Sinha, Pekka J Korhonen, and Jyrki Wallenius. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *Evolutionary Computation, IEEE Transactions on*, 14(5):723–739, 2010.

[48] Kalyanmoy Deb, Ankur Sinha, and Saku Kukkonen. Multi-objective test problems, linkages, and evolutionary methodologies. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1141–1148. ACM, 2006.

[49] Kalyanmoy Deb and Aravind Srinivasan. Innovization: Innovating design principles through optimization. In *Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pages 1629–1636. ACM, 2006.

[50] Kalyanmoy Deb, Lothar Thiele, Marco Laumanns, and Eckart Zitzler. S-calable multi-objective optimization test problems. In *Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA)*, pages 825–830. Proceedings of the Congress on Evolutionary Computation (CEC-2002),(Honolulu, USA), 2002.

[51] Stéphane Doncieux and Mohamed Hamdaoui. Evolutionary algorithms to analyse and design a controller for a flapping wings aircraft. In *New Horizons in Evolutionary Robotics*, pages 67–83. Springer, 2011.

[52] Nicole Drechsler, Rolf Drechsler, and Bernd Becker. Multi-objective optimisation based on relation favour. In *Evolutionary multi-criterion optimization*, pages 154–166. Springer, 2001.

[53] Habiba Drias. Genetic algorithm versus memetic algorithm for association rules mining. In *Nature and Biologically Inspired Computing (NaBIC), 2014 Sixth World Congress on*, pages 208–213. IEEE, 2014.

[54] Francis Ysidro Edgeworth. *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. Number 10. CK Paul, 1881.

[55] M Farina. A neural network based generalized response surface multiobjective evolutionary algorithm. In *Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on*, volume 1, pages 956–961. IEEE, 2002.

[56] Alberto Fernández, Salvador García, Julián Luengo, Ester Bernadó-Mansilla, and Francisco Herrera. Genetics-based machine learning for rule induction: state of the art, taxonomy, and comparative study. *Evolutionary Computation, IEEE Transactions on*, 14(6):913–941, 2010.

[57] David B Fogel. The advantages of evolutionary computation. In *BCEC*, pages 1–11. Citeseer, 1997.

[58] Carlos M Fonseca and Peter J Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary computation*, 3(1):1–16, 1995.

[59] Carlos M Fonseca, Peter J Fleming, et al. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *ICGA*, volume 93, pages 416–423. Citeseer, 1993.

[60] Carlos M Fonseca, Luís Paquete, and Manuel López-Ibánez. An improved dimension-sweep algorithm for the hypervolume indicator. In *Evolutionary Computation, 2006. CEC 2006. IEEE Congress on*, pages 1157–1163. IEEE, 2006.

[61] Mario Garza-Fabre, Gregorio Toscano Pulido, and Carlos A Coello Coello. Ranking methods for many-objective optimization. In *MICAI 2009: Advances in Artificial Intelligence*, pages 633–645. Springer, 2009.

[62] Abhinav Gaur and Kalyanmoy Deb. Towards an automated innovization method for handling discrete search spaces. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 2899–2906, May 2015.

[63] Chi-Keong Goh and K Chen Tan. A competitive-cooperative coevolutionary paradigm for dynamic multiobjective optimization. *Evolutionary Computation, IEEE Transactions on*, 13(1):103–127, 2009.

[64] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1989.

[65] Prabhat Hajela and C-Y Lin. Genetic search strategies in multicriterion optimal design. *Structural optimization*, 4(2):99–107, 1992.

[66] Jesús García Herrero, Antonio Berlanga, and José M Molina Lopez. Effective evolutionary algorithms for many-specifications attainment: application to air traffic control tracking filters. *Evolutionary Computation, IEEE Transactions on*, 13(1):151–168, 2009.

[67] Jeffrey Horn, Nicholas Nafpliotis, and David E Goldberg. A niched pareto genetic algorithm for multiobjective optimization. In *Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on*, pages 82–87. Ieee, 1994.

[68] Ko-Wei Huang, Jui-Le Chen, Chu-Sing Yang, and Chun-Wei Tsai. A memetic gravitation search algorithm for solving clustering problems. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 751–757. IEEE, 2015.

[69] Simon Huband, Philip Hingston, Luigi Barone, and Lyndon While. A review of multiobjective test problems and a scalable test problem toolkit. *Evolutionary Computation, IEEE Transactions on*, 10(5):477–506, 2006.

[70] Ken-ichi Ikeda, Hajime Kita, and Shigenobu Kobayashi. Failure of pareto-based MOEAs: does non-dominated really mean near to optimal? In *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on*, volume 2, pages 957–962. IEEE, 2001.

[71] Ronald L Iman. *Latin hypercube sampling*. Wiley Online Library, 2008.

[72] Hisao Ishibuchi, Naoya Akedo, and Yusuke Nojima. A many-objective test problem for visually examining diversity maintenance behavior in a decision space. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 649–656. ACM, 2011.

[73] 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*, pages 91–100. Springer, 2010.

[74] Hisao Ishibuchi, Hiroji Masuda, Yuki Tanigaki, and Yusuke Nojima. Review of coevolutionary developments of evolutionary multi-objective and many-objective algorithms and test problems. In *Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 2014 IEEE Symposium on*, pages 178–184. IEEE, 2014.

[75] Hisao Ishibuchi, Hiroyuki Masuda, Yuki Tanigaki, and Yusuke Nojima. Difficulties in specifying reference points to calculate the inverted generational distance for many-objective optimization problems. In *Computational Intelligence in Multi-Criteria Decision-Making (MCDM), 2014 IEEE Symposium on*, pages 170–177. IEEE, 2014.

[76] Hisao Ishibuchi and Tadahiko Murata. A multi-objective genetic local search algorithm and its application to flowshop scheduling. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 28(3):392–403, 1998.

[77] Hisao Ishibuchi, Yuji Sakane, Noritaka Tsukamoto, and Yusuke Nojima. Simultaneous use of different scalarizing functions in MOEA/D. In *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, pages 519–526. ACM, 2010.

[78] Hisao Ishibuchi, Noritaka Tsukamoto, and Yusuke Nojima. Evolutionary many-objective optimization: A short review. In *IEEE Congress on Evolutionary Computation*, pages 2419–2426, 2008.

[79] Hisao Ishibuchi, Masakazu Yamane, Naoya Akedo, and Yusuke Nojima. Two-objective solution set optimization to maximize hypervolume and decision space diversity in multiobjective optimization. In *Soft Computing and Intelligent Systems (SCIS) and 13th International Symposium on Advanced Intelligent Systems (ISIS), 2012 Joint 6th International Conference on*, pages 1871–1876. IEEE, 2012.

[80] Andrzej Jaszkiewicz, Hisao Ishibuchi, and Qingfu Zhang. Multiobjective memetic algorithms. In Ferrante Neri, Carlos Cotta, and Pablo Moscato, editors, *Handbook of Memetic Algorithms*, volume 379 of *Studies in Computational Intelligence*, pages 201–217. Springer Berlin Heidelberg, 2012.

[81] Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. Dynamic weighted aggregation for evolutionary multi-objective optimization: Why does it work and how? In *Proceedings of the Genetic and Evolutionary Computation Conference*, pages 1042–1049, 2001.

[82] Yaochu Jin and Bernhard Sendhoff. Pareto-based multiobjective machine learning: An overview and case studies. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 38(3):397–415, 2008.

[83] Hillol Kargupta and David E Goldberg. Search, blackbox optimization, and sample complexity. In *FOGA*, pages 291–324. Citeseer, 1996.

[84] Joshua D Knowles and David W Corne. Approximating the nondominated front using the pareto archived evolution strategy. *Evolutionary computation*, 8(2):149–172, 2000.

[85] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[86] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[87] Andreas Konstantinidis, Christoforos Charalambous, Aimin Zhou, and Qingfu Zhang. Multi-objective mobile agent-based sensor network routing using moea/d. In *Evolutionary Computation (CEC), 2010 IEEE Congress on*, pages 1–8. IEEE, 2010.

[88] Mario Köppen and Kaori Yoshida. Substitute distance assignments in NSGA-II for handling many-objective optimization problems. In *Evolutionary Multi-Criterion Optimization*, pages 727–741. Springer, 2007.

[89] Fumiya Kudo and Tomohiro Yoshikawa. Knowledge extraction in multi-objective optimization problem based on visualization of pareto solutions. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–6. IEEE, 2012.

[90] Saku Kukkonen and Kalyanmoy Deb. Improved pruning of non-dominated solutions based on crowding distance for bi-objective optimization problems. In *Proceedings of the World Congress on Computational Intelligence (WCCI-2006)(IEEE Press). Vancouver, Canada*, pages 1179–1186. Proceedings of the World Congress on Computational Intelligence (WCCI-2006)(IEEE Press). Vancouver, Canada, 2006.

[91] Saku Kukkonen and Jouni Lampinen. Ranking-dominance and many-objective optimization. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 3983–3990. IEEE, 2007.

[92] Frank Kursawe. A variant of evolution strategies for vector optimization. In *Parallel Problem Solving from Nature*, pages 193–197. Springer, 1991.

[93] Marco Laumanns, Lothar Thiele, Kalyanmoy Deb, and Eckart Zitzler. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary computation*, 10(3):263–282, 2002.

[94] Bingdong Li, Jinlong Li, Ke Tang, and Xin Yao. Many-objective evolutionary algorithms: A survey. *ACM Computing Surveys*, V(N), 2015.

[95] Hui Li and Qingfu Zhang. Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II. *Evolutionary Computation, IEEE Transactions on*, 13(2):284–302, 2009.

[96] Ke Li, Kalyanmoy Deb, Qingfu Zhang, and Sam Kwong. An evolutionary many-objective optimization algorithm based on dominance and decomposition. *Evolutionary Computation, IEEE Transactions on*, 19(5):694–716, 2015.

[97] Yadong Li, Jing Liu, and Chenlong Liu. A comparative analysis of evolutionary and memetic algorithms for community detection from signed social networks. *Soft Computing*, 18(2):329–348, 2014.

[98] Chenlong Liu, Jing Liu, and Zhongzhou Jiang. A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks. *Cybernetics, IEEE Transactions on*, 44(12):2274–2287, 2014.

[99] Lawrence Mandow and José Luis Pérez De La Cruz. Multiobjective A* search with consistent heuristics. *Journal of the ACM (JACM)*, 57(5):27, 2010.

[100] Achille Messac, Amir Ismail-Yahaya, and Christopher A Mattson. The normalized normal constraint method for generating the pareto frontier. *Structural and multidisciplinary optimization*, 25(2):86–98, 2003.

[101] Ryan Meuth, Meng-Hiot Lim, Yew-Soon Ong, and Donald C Wunsch II. A proposition on memes and meta-memes in computing for higher-order learning. *Memetic Computing*, 1(2):85–100, 2009.

[102] Yoan Miche, Meng-Hiot Lim, Amaury Lendasse, and Yew-Soon Ong. Meme representations for game agents. *World Wide Web*, 18(2):215–234, 2015.

[103] Nestor F Michelena and Alice M Agogino. Multiobjective hydraulic cylinder design. *Journal of Mechanical Design*, 110(1):81–87, 1988.

[104] Kaisa Miettinen. *Nonlinear Multiobjective Optimization.* Volume 12 of International Series in Operations Research & Management Science. Springer US, 1999.

[105] Mohamed Wiem Mkaouer, Marouane Kessentini, Slim Bechikh, Kalyanmoy Deb, and Mel Ó Cinnéide. High dimensional search-based software engineering: Finding tradeoffs among 15 objectives for automating software refactoring

using NSGA-III. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 1263–1270. ACM, 2014.

[106] Antonio J Nebro and Juan J Durillo. A study of the parallelization of the multi-objective metaheuristic MOEA/D. In *Learning and Intelligent Optimization*, pages 303–317. Springer, 2010.

[107] Ferrante Neri and Carlos Cotta. Memetic algorithms and memetic computing optimization: A literature review. *Swarm and Evolutionary Computation*, 2:1–14, 2012.

[108] Ferrante Neri, Jari Toivanen, Giuseppe Leonardo Cascella, and Yew-Soon Ong. An adaptive multimeme algorithm for designing hiv multidrug therapies. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 4(2):264–278, 2007.

[109] Long Nguyen, Lam T Bui, and Hussein A Abbass. DMEA-II: the direction-based multi-objective evolutionary algorithm-II. *Soft Computing*, 18(11):2119–2134, 2014.

[110] Shigeru Obayashi and Daisuke Sasaki. Visualization and data mining of pareto solutions using self-organizing map. In *Evolutionary multi-criterion optimization*, pages 796–809. Springer, 2003.

[111] Tatsuya Okabe, Yaochu Jin, Markus Olhofer, and Bernhard Sendhoff. On test functions for evolutionary multi-objective optimization. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 792–802. Springer, 2004.

[112] Tatsuya Okabe, Yaochu Jin, and Bernhard Sendhoff. A critical survey of performance indices for multi-objective optimisation. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 2, pages 878–885. IEEE, 2003.

[113] Yew-Soon Ong, Meng Hiot Lim, and Xianshun Chen. Research frontier-memetic computation - past, present & future. *IEEE Computational Intelligence Magazine*, 5(2):24, 2010.

[114] Panos Y Papalambros and Douglass J Wilde. *Principles of optimal design: modeling and computation*. Cambridge university press, 2000.

[115] Vilfredo Pareto. *Cours d'économie politique*. Rouge, Lausanne, Switzerland, 1896.

[116] Somwrita Sarkar, Andy Dong, and John S Gero. A learning and inference mechanism for design optimization problem (re)-formulation using singular value decomposition. In *ASME 2008 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pages 97–106. American Society of Mechanical Engineers, 2008.

[117] Ruhul Sarker and Hussein A Abbass. Differential evolution for solving multiobjective optimization problems. *Asia-Pacific Journal of Operational Research*, 21(02):225–240, 2004.

[118] Hiroyuki Sato, Hernán E Aguirre, and Kiyoshi Tanaka. Controlling dominance area of solutions and its impact on the performance of MOEAs. In *Evolutionary Multi-Criterion Optimization*, pages 5–20. Springer, 2007.

[119] J David Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In *Proceedings of the 1st International Conference on Genetic Algorithms, Pittsburgh, PA, USA, July 1985*, pages 93–100, 1985.

[120] Jason R Schott. Fault tolerant design using single and multicriteria genetic algorithm optimization. Technical report, DTIC Document, 1995.

[121] Paolo Serafini. Simulated annealing for multi objective optimization problems. In *Multiple criteria decision making*, pages 283–292. Springer, 1994.

[122] Bahareh Shaabani and Hedieh Sajedi. Mmdt: Multi-objective memetic rule learning from decision tree. *Journal of Computer & Robotics*, 5(1):37–46, 2014.

[123] Kamran Shafi, Axel Bender, and Hussein A Abbass. Multi objective learning classifier systems based hyperheuristics for modularised fleet mix problem. In *Simulated Evolution and Learning*, pages 381–390. Springer, 2012.

[124] Jim E Smith. Coevolving memetic algorithms: a review and progress report. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 37(1):6–17, 2007.

[125] Moshe Sniedovich. *Dynamic programming: foundations and principles*. CRC press, 2010.

[126] Andrew R Solow and Stephen Polasky. Measuring biological diversity. *Environmental and Ecological Statistics*, 1(2):95–103, 1994.

[127] Francis Sourd and Olivier Spanjaard. A multiobjective branch-and-bound framework: Application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, 20(3):472–484, 2008.

[128] Nidamarthi Srinivas and Kalyanmoy Deb. Muiltiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221–248, 1994.

[129] Sujatha Srinivasan and Sivakumar Ramakrishnan. Evolutionary multi objective optimization for rule mining: a review. *Artificial Intelligence Review*, 36(3):205–248, 2011.

[130] Wolfram Stadler. A survey of multicriteria optimization or the vector maximum problem, part I: 1776–1960. *Journal of Optimization Theory and Applications*, 29(1):1–52, 1979.

[131] Bradley S Stewart and Chelsea C White III. Multiobjective A*. *Journal of the ACM (JACM)*, 38(4):775–814, 1991.

[132] Kazuyuki Sugimura, Shinkyu Jeong, Shigeru Obayashi, and Takeshi Kimura. Kriging-model-based multi-objective robust optimization and trade-off-

rule mining using association rule with aspiration vector. In *Evolutionary Computation, 2009. CEC'09. IEEE Congress on*, pages 522–529. IEEE, 2009.

[133] Kazuyuki Sugimura, Shigeru Obayashi, and Shinkyu Jeong. Multi-objective design exploration of a centrifugal impeller accompanied with a vaned diffuser. In *ASME/JSME 2007 5th Joint Fluids Engineering Conference*, pages 939–946. American Society of Mechanical Engineers, 2007.

[134] André Sülflow, Nicole Drechsler, and Rolf Drechsler. Robust multi-objective optimization in high dimensional spaces. In *Evolutionary multi-criterion optimization*, pages 715–726. Springer, 2007.

[135] El-Ghazali Talbi. *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons, 2009.

[136] Kay Chen Tan. Advances in evolutionary multi-objective optimization. In *Soft Computing Applications*, pages 7–8. Springer, 2013.

[137] Kay Chen Tan, Eik Fun Khor, Tong Heng Lee, and YJ Yang. A tabu-based exploratory evolutionary algorithm for multiobjective optimization. *Artificial Intelligence Review*, 19(3):231–260, 2003.

[138] Kay Chen Tan, Tong Heng Lee, and Eik Fun Khor. Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. *Artificial intelligence review*, 17(4):251–290, 2002.

[139] Masahiro Tanaka and Tetsuzo Tanino. Global optimization by the genetic algorithm in a multiobjective decision support system. In *Proceedings of the Tenth International Conference on Multiple Criteria Decision Making*, volume 2, pages 261–270, 1992.

[140] Tomoaki Tatsukawa, Taku Nonomura, Akira Oyama, and Kozo Fujii. A new multiobjective genetic programming for extraction of design information from non-dominated solutions. In *Evolutionary Multi-Criterion Optimization*, pages 528–542. Springer, 2013.

[141] Lothar Thiele, Kaisa Miettinen, Pekka J Korhonen, and Julian Molina. A preference-based evolutionary algorithm for multi-objective optimization. *Evolutionary Computation*, 17(3):411–436, 2009.

[142] Tamara Ulrich. Pareto-set analysis: Biobjective clustering in decision and objective spaces. *Journal of Multi-Criteria Decision Analysis*, 20(5-6):217–234, 2013.

[143] Tamara Ulrich, Johannes Bader, and Lothar Thiele. Defining and optimizing indicator-based diversity measures in multiobjective search. In *Parallel Problem Solving from Nature, PPSN XI*, pages 707–717. Springer, 2010.

[144] Tamara Ulrich, Dimo Brockhoff, and Eckart Zitzler. Pattern identification in pareto-set approximations. In *Proceedings of the 10th annual conference on Genetic and evolutionary computation*, pages 737–744. ACM, 2008.

[145] Tamara Ulrich and Lothar Thiele. Maximizing population diversity in single-objective optimization. In *Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 641–648. ACM, 2011.

[146] A Van Veldhuizen and G Lamont. Multi-objective evolutionary algorithms: Analyzing the state-of-the art. *Evolutionary Computation*, 8(2):125–47, 1970.

[147] David A Van Veldhuizen. Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Technical report, DTIC Document, 1999.

[148] M Visée, J Teghem, M Pirlot, and EL Ulungu. Two-phases method and branch and bound procedures to solve the bi–objective knapsack problem. *Journal of Global Optimization*, 12(2):139–155, 1998.

[149] Lyndon While, Lucas Bradstreet, and Luigi Barone. A fast way of calculating exact hypervolumes. *Evolutionary Computation, IEEE Transactions on*, 16(1):86–95, 2012.

[150] Lyndon While, Phil Hingston, Luigi Barone, and Simon Huband. A faster algorithm for calculating hypervolume. *Evolutionary Computation, IEEE Transactions on*, 10(1):29–38, 2006.

[151] Brian C Williams and Jonathan Cagan. Activity analysis: Simplifying optimal design problems through qualitative partitioning. *Engineering Optimization*, 27(2):109–137, 1996.

[152] J. Xiong, R. Leus, Z. Yang, and Abbass H.A. Evolutionary multi-objective resource allocation and scheduling in the Chinese navigation satellite system project. *European Journal of Operational Research*, Accepted Nov-2015.

[153] Jian Xiong, Yingwu Chen, Kewei Yang, and Jing Liu. A decision support model for multi-attribute group decision making using a multi-objective optimization approach. *International Journal of Computational Intelligence Systems*, 6(2):337–353, 2013.

[154] Jian Xiong, Yingwu Chen, and Zhongbao Zhou. Resilience analysis for project scheduling with renewable resource constraint and uncertain activity durations. *Management*, 12(2):719–737, 2016.

[155] Jian Xiong, Jing Liu, Yingwu Chen, and Hussein A Abbass. A knowledge-based evolutionary multi-objective approach for stochastic extended resource investment project scheduling problems. *Evolutionary Computation, IEEE Transactions on*, 18(5):742–763, 2013.

[156] Jian Xiong, Li-ning Xing, and Ying-wu Chen. Robust scheduling for multi-objective flexible job-shop problems with random machine breakdowns. *International Journal of Production Economics*, 141(1):112–126, 2013.

[157] Jian Xiong, Ke-wei Yang, Jing Liu, Qing-song Zhao, and Ying-wu Chen. A two-stage preference-based evolutionary multi-objective approach for capability planning problems. *Knowledge-Based Systems*, 31:128–139, 2012.

[158] Jingyu Yan, Chongguo Li, Zhi Wang, Lei Deng, and Demin Sun. Diversity metrics in multi-objective optimization: review and perspective. In *Integration Technology, 2007. ICIT'07. IEEE International Conference on*, pages 553–557. IEEE, 2007.

[159] G.G. Yen and Zhenan He. Performance metric ensemble for multiobjective evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on*, 18(1):131–144, Feb 2014.

[160] Yuan Yuan, Hua Xu, and Bo Wang. An improved NSGA-III procedure for evolutionary many-objective optimization. In *Proceedings of the 2014 conference on Genetic and evolutionary computation*, pages 661–668. ACM, 2014.

[161] Bin Zhang, Kamran Shafi, and Hussein Abbass. Online knowledge-based evolutionary multi-objective optimization. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 2222–2229. IEEE, 2014.

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

[163] Qingfu Zhang, Wudong Liu, and Hui Li. The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances. In *IEEE Congress on Evolutionary Computation*, volume 1, pages 203–208, 2009.

[164] Qingfu Zhang, Aimin Zhou, Shizheng Zhao, Ponnuthurai Nagaratnam Suganthan, Wudong Liu, and Santosh Tiwari. Multiobjective optimization test instances for the cec 2009 special session and competition. *University of Essex, Colchester, UK and Nanyang Technological University, Singapore, Special Session on Performance Assessment of Multi-Objective Optimization Algorithms, Technical Report*, pages 1–30, 2008.

[165] Weicai Zhong, Jing Liu, Mingzhi Xue, and Licheng Jiao. A multiagent genetic algorithm for global numerical optimization. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 34(2):1128–1141, 2004.

[166] Aimin Zhou, Bo-Yang Qu, Hui Li, Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, and Qingfu Zhang. Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, 1(1):32–49, 2011.

[167] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In K.C. Giannakoglou et al., editors, *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pages 95–100. International Center for Numerical Methods in Engineering (CIMNE), 2002.

[168] Eckart Zitzler. *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. PhD thesis, ETH Zurich, Switzerland, 1999.

[169] Eckart Zitzler, Kalyanmoy Deb, and Lothar Thiele. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary computation*, 8(2):173–195, 2000.

[170] Eckart Zitzler and Simon Künzli. Indicator-based selection in multiobjective search. In *Parallel Problem Solving from Nature-PPSN VIII*, pages 832–842. Springer, 2004.

[171] Eckart Zitzler and Lothar Thiele. Multiobjective optimization using evolutionary algorithms - a comparative case study. In *Parallel problem solving from naturePPSN V*, pages 292–301. Springer, 1998.

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