AN ADVANCED *p*-METRIC BASED MANY-OBJECTIVE EVOLUTIONARY ALGORITHM

By

CHAITANYA KUMAR RAVIPALLI

Bachelor of Technology in Electronics & Instrumentation

Gitam University

Visakhapatnam, Andhra Pradesh, India

2015

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May 2018

AN ADVANCED *p*-METRIC BASED

MANY-OBJECTIVE EVOLUTIONARY ALGORITHM

Thesis Approved:

Dr. Gary Yen

Thesis Adviser

Dr Ramakumar R. G

Dr. Weili Zhang

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor and academic mentor, Dr. Gary G. Yen for his continuous support, patience and motivation. During my study of master's and research, the knowledge he shared will stay with me all my life. I am blessed to have an advisor like him. Without Dr. Yen, this research work and thesis write-up would have never been possible.

I would also like to thank Dr. Ramakumar R. G and Dr Weili Zhang for being my graduate committee members and I would also like to thank them for attending my defense amidst their busy schedule.

I would also like to extend my sincere thanks to the department of Electrical and Computer Engineering at Oklahoma State University for giving me the opportunity to pursue my Masters degree.

I am forever grateful to my parents, Koteswara Rao Ravipalli and Sridevi Ravipalli, who supported me throughout my career. I would like to thank them for their love, support and patience right through my research work.

Name: CHAITANYA KUMAR RAVIPALLI

Date of Degree: MAY, 2018

Title of Study: AN ADVANCED *p*-METRIC BASED MANY-OBJECTIVE EVOLUTIONARY ALGORITHM

Major Field: ELECTRICAL ENGINEERING

Abstract: Evolutionary many objective based optimization has been gaining a lot of attention from the evolutionary computation researchers and computational intelligence community. Many of the state-of-the-art multi-objective and many-objective optimization problems (MOPs, MaOPs) are inefficient in maintaining the convergence and diversity performances as the number of objectives increases in the modern-day real-world applications. This phenomenon is obvious indeed as Pareto-dominance based EAs employ non-dominated sorting which fails considerably in providing enough convergent pressure towards the Pareto front (PF). Researchers invested much more time and effort in addressing this issue by improving the scalability in MaOPs and they have come up with non-Pareto-dominance-based EAs such as decomposition-based, indicator-based and reference-based approaches. In addition to that, the algorithm has to account for the additional computational budget. This thesis proposes an advanced polar-metric (*p*-metric) based Many-objective EA (in short APMOEA) for tackling both MOPs and MaOPs. *p*-metric, a recently proposed performance based visualization metric, employs an array of uniformly, distributed direction vectors. In APMOEA, a two-phase selection scheme is employed which combines both non-dominated sorting and *p*-metric. Moreover, this thesis also proposes a modified P-metric methodology in order to adjust the direction vectors dynamically. In the experiments, we compare APMOEA with four state-of-the-art Many-objective EAs under, three performance indicators. According to the empirical results, APMOEA shows much improved performances on most of the test problems, involving both MOPs and MaOPs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

INTRODUCTION

Optimization is extensively involved in many real-world problems. More often than not in physical world solving complex problems involves the simultaneous optimization of multiple conflicting objectives. Typically, these problems occur in various fields including engineering, chemistry, finance, physics and manufacturing. Some real-world scenarios include engineers aiming for the best performance of their designs; manufacturing representatives expect efficacy in their production systems; bank lenders try to minimize the risk of investment while maximizing the returns. Generally, the optimization process involves a number of design challenges in the form of optimizing some objectives and corresponding constraints. There exist many classical mathematical methods to solve multi-objective optimization problems. However, they at best offer adequate performance for a maximum of three objectives and fail to perform well in the environment of more than three conflicting objectives.

Most of the multi-objective evolutionary algorithms (MOEAs) utilize Pareto-dominance relationship, they perform quite well with three objectives, but, they were proven to be ineffective when the number of objectives is more than three. The reason for this scenario is that, most of the existing MOEAs utilize pair-wise comparison (i.e, tournament selection) and they lose their efficiency with more number of objectives. There by, dramatically increasing the number of non-dominated solutions with the increase in the number of objectives and to create some selection

pressure in this case is almost unattainable. Hence, MOEAs have very less to negligible success in tackling Many-objective optimization problems (MaOPs).

1.1 Problem Definition

MaOPs contain *m* (i.e., *m*>3) conflicting objectives to be solved concurrently. Generally, a MaOP can be defined by the following equation.

$$f(x) = (f_1(x), \cdots f_m(x)) \qquad (1)$$

$$\text{s.t} \quad x \subset \phi$$

Here, $\Phi \subset \mathbb{R}_n$ is the search space, $f: \Phi \rightarrow \Omega \subset \mathbb{R}_m$, and Ω is the objective space . Here , m are the number of decision variables and number of objectives respectively. For the sake of convenient discussions, $f_1(x)$ is assumed as a minimization problem, where, $f_1(x), \cdots, f_m(x)$ are a set of minimization problems. Usually, there is more than one solution for a specific minimization problem: multiple trade-off solutions also called Pareto-optimal solutions, which in turn forms a Pareto set (PS) in the decision space and is mapped as Pareto front (PF) in the objective space. So, an algorithm's goal is to solve a minimization problem $(f(x))$ to obtain a PF full of uniformly distributed solutions on it. To accomplish the same goal, numerous Multi-objective evolutionary algorithms were proposed in the last few decades.

Many researchers and practitioners are obliged to follow one between the two paths. The first path is to adjust the number of objectives they are dealing with to an algorithm or module which performs. Typically, this process is done by combining several objectives (can be conflicting) to one. This approach is not ideal because unlike treating objectives one at a time and optimizing them separately, optimizing a combination of objectives is bound to lose useful trade-off solutions. On top of it, combining two objectives without much knowledge of their objectives is very hard, if not impossible.

Generally, we think to develop strategies in order to reduce the number of objectives while retaining as much information of as many objectives as possible. For example, Brockhoff and Zitzler [1] initially identified conflict and nonconflictual relationships between each pair of objectives and then combined non-conflicting objectives into one objective. To find the correct lower dimensional interactions of each objective by iteratively starting from the interior of the search space heading for the Pareto-optimal region, Deb and Saxena [2] proposed a Principle Component Analysis method. Singh *et al.* [3] generated an approximate non-dominated front and came to conclusion whether an objective is redundant or not, just by having a look at the approximate front.

While objective reduction works in some special conditions, there are a lot of real-world problems whose objectives cannot be reduced any further. For those problems, the algorithms will only stick with the relatively important objectives [4]. Moreover, eliminating few objectives will not solve most of the MaOPs to produce desired results. Even after the number of objectives are reduced to a maximal extent, it is ambiguous as to how the derived Pareto front in a reduced low-dimensional space can mimic the true Pareto front in the original higher dimensional space.

The second path is to make use of a number of algorithms, one for each dimension. This obviously is an inconvenient and a cumbersome approach, although it is just an alternative followed to avoid the problems caused in the first step.

In researchers' perspective, the never-ending trade-off amongst convergence and diversity becomes much more complicated with the increase of the objectives. Moreover, the convergence and diversity dilemma are usually conflicting. Researchers' often find it difficult to find the right balance between convergence and diversity as early emphasis on diversity will either delay the convergence or the solutions get stuck in a local optimum. Furthermore, relying on convergence alone will not produce all the trade-off solutions in most cases. Besides, it is almost inconceivable without a landscape of the solution set to adjudge which of the two (convergence and diversity) is the optimizer's primary concern. On the grounds of that, most of the currently existing algorithms maintain relatively the same balance between the both. Without much ambiguity, one can say that

the final goal of this line of research is to design an algorithm that can dynamically adapt the delicate balance between convergence and diversity at any stage of the problem searching process.

1.2 Motivation

Solving the first problem involves careful construction an algorithm that can adapt the number of objectives at any stage of the problem scenario without any complicated mapping techniques. This step is usually important because every problem is categorized into either three or four-dimensional categories [5, 6, 7]. The actual range of the problem is further categorized into finer grains. Strictly speaking, categorized into one many-objective category does not mean that an eight objectives problem, for example, a 15-objectives problem should be treated in the same manner. In addition to that, the algorithms are usually inconsistent to either scale up (to solve dimensional problems higher than initially projected) or down (to solve dimensional problems lower than initially predicted), these inconsistencies makes it more difficult to rely on either one of them to develop the desired algorithm. Furthermore, to the algorithmic motivations, there are other practical motivations for the initial part of the study. We will discuss four of them here. First, in order to solve any optimization problem, it (the problem) must first be implemented (coded or expressed symbolically) within the optimizer (either a computer code or a commercial software). Often, this implementation process involves linking the optimizer to a third-party evaluation software such as a finite element or a computational fluid dynamics software or a network flow simulator etc. Secondly, in order to obtain better performance of the algorithm, customizing the optimizer itself for the problem at hand is recommended [8, 9] as it can be done either by introducing new operators or by modifying existing genetic operators utilizing the “heuristics”. For example, a heuristically biased initial population is preferred over generating a random initial population. These customized initializations and algorithmic modifications involve careful analysis, and are certainly time-consuming. Thirdly, we need to address each objective individually for most multi and many-

objective optimization methods in order to obtain ideal and Nadir points prior to solving the actual multi or many-objective version of the problem. Ideal points are obtained by individually optimizing each objective over the search space. Where, Nadir points are obtained from construction of worst objective function values of Pareto optimal solutions, it gets tougher when the objective functions increase. Often, we come across methods where, several lower-dimensional runs are implemented, executed to examine either the performance of the algorithm or to be certain in obtaining better high dimensional front [10]. Fourthly, in design exploration problems, objectives, constraints and decision variables are altered to get an even better idea of the possible range of optimal solutions [11]. Taking these four challenges into consideration, let us presume that a distinct optimizer is needed for each and every dimensional version of the actual optimization problem. In that case, for each and every optimizer the following changes has to be made: implementation of the problem, slight modifications in the algorithm and customized initializations. Thereby making the overall process slow, tiresome and prone to errors. Solving a single objective version of a multi-objective optimization problem will be so much more complicated as it requires a particular optimizer.

Finally, we need to design an optimizer which suits its dimensionality of every version of a design exploration problem, which in turn relies on the distinct combination of number of constraints decision variables and objectives. Alternatively, if an unified optimization algorithm competent of managing one-to-many objectives efficiently is available, modification of the algorithm based on heuristics, problem implementation and integrating with external evaluation soft-ware can only be done once which would be more convenient for solving different dimensional versions of the original problem. This provides flexibility for users to move back and forth amid different objective -dimensions of the same original problem and it saves a great deal of time, effort and most predominantly reduces if not free from process errors.

Developing a unified algorithm that can adapt to the dimension dynamically is only half the story. We also need to make sure that when it comes to more than one objective, the state-of-the-art algorithm should be able understand the scenario and emphasize on either convergence, diversity or both. Shifting the emphasis of the algorithm from one to another might help in discovering solutions that are difficult to attain otherwise and might as well reveal some interesting aspects about the optimization problem itself. It is very helpful for a practitioner with little or no knowledge about the problem to try and put emphasis only on convergence at one time and on diversity at another. This will give the direction to work with. After all, it is difficult to tell ahead of time, whether the problem need to emphasize on convergence or diversity. For the very reason it will be extremely helpful if the algorithm itself can solve the problem by dynamically adapting the need for more convergence or diversity.

Considering the above trends, we employ *p*-Metric based technique to design our MOEA in addition to non-dominated sorting. Combining both the methodologies gives a right balance between convergence and diversity for the MOEA. Later, we explain how we emphasize on either convergence or diversity at a specific stage by updating the direction vectors which addresses a tricky problem. By employing *p*-Metric based technique, we are equipped with *p*-Metric based visualization [91] tool. As mentioned before maintaining a right balance between convergence and diversity becomes next to impossible, with the increase in the number of objectives. This visualization tool helps the algorithm to monitor convergence performance and diversity performance at any stage of the algorithm, which makes our MOEA very accurate.

1.3 Thesis Statement

This thesis develops an APMOEA (Advanced *p*-Metric based Many-Objective Evolutionary Algorithm), where a two-phase selection scheme is employed which combines both non-dominated sorting and *p*-metric selection. Moreover, this thesis also proposes a modified *p*-metric

methodology in order to adjust the direction vectors dynamically. In the experiments, we compare APMOEA with four state-of-the-art Many-Objective EAs under, three performance indicators, including *p*-metric.

1.4 Thesis Organization

Chapter Two provides the literature review for each type of MOEAs for MaOPs. It presents the necessary background with references to each type of methods and analyzes pros and cons of them. It also lists widely used performance metrics and most popular benchmark functions and test suites for different scales of optimization tasks.

Chapter Three elaborates the proposed method of fitness evaluation based on *p*-metric. It also discloses a unique way to maintain the convergence and diversity by utilizing non-dominated sorting and both the fitness evaluations in the form of Radial Distance calculation (RD) and Angular Distance (AD). This chapter also provides algorithms and related framework of APMOEA and the two-phase selection strategy.

Chapter Four compares the performance of APMOEA with four other state-of-the-art Many-Objective EAs and we tabulate the results based on three performance metrics. The last one being the visualization based on *p*-metric method. We detail the experimental results for seventeen selected benchmark problems from both DTLZ, and MAF test suites. These problems offer various problem characteristics that present numerous degrees of complications for the underlying state-of-the-art Many-Objective EAs.

In **Chapter Five,** we concludes the study. We also provide recommendations for future work. Additionally, we probe on how one can extend APMOEA into a constrained many objective evolutionary algorithm.

CHAPTER II

LITERATURE REVIEWS

In this chapter, we review and analyze five classes of MOEAs in terms of their convergence and diversity methods used. For each class of MOEAs, we present an algorithm in detail to gain more insight.

2.1 MOEAs Based on Pareto-Dominance Modification

Firstly, there are a few designs which incorporate modified Pareto dominance concepts to adapt it to a higher dimensional space include Pareto α-Dominance [12], Pareto ε-Dominance [12], and Pareto cone ε-Dominance [12]. For all the above methods, parameters are heuristically integrated. Each modified Pareto dominance design is a relaxed form of the Pareto dominance in that it makes one individual dominates others easier in a high-dimensional space. Based on a similar idea, proposed a ε - Domination Based Multi-Objective Evolutionary Algorithm (ε-MOEA) [13] was proposed and has been performing well for MaOPs [14].

This class of MOEAs replace Pareto-dominance and provides a new fitness assignment measure to select individuals in the evolution process in order to push the whole population towards the true Pareto front. Accordingly, the convergence power mainly comes from the modified dominance relation, which is the domain criterion in the evolution process. Different modification methods can be considered as the adjustment of dominance degree, the degree level varies from one dominance relation to another. The hardest dominance level being the Pareto

Dominance where one individual is better than the other in one dimension if and only if its objective value is strictly smaller (for minimization problems) than the other. On the contrary, in ε-Dominance, one individual is better than the other in one dimension means that, it is not worse than the other in the same dimension, it might vary with some other dimension.

This design achieves diversity by using the parameter of degree, e.g., ε in ε-Dominance and α in α-Dominance. In addition to controlling the dominance degree, these parameters also determines the size of hyperboxes. Where in each of the hyperbox can contain no more than one individual in order to maintain the diversity.

2.1.1 ε*-MOEA* [10]

Input: a MOP and stopping criteria; (the size of population)*;*

Step1: Randomly initialize a population *P*(0). The non-dominated solutions of *P*(0) are copied to an archive population *A*(0). Set the iteration counter *t* = 0.

Step2: One solution *p* is chosen from the population *P*(*t*) using the pop selection procedure.
Two population members from *P*(*t*) are picked up at random, if one dominates the other, the former is chosen; otherwise, randomly choose one of non-dominated solutions

Step3: One solution *e* is randomly chosen from the archive population *A*(*t*)

Step4: One offspring solutions *c* is created using *p* and *e*.

Step5: Solution *c* is included in *P*(*t*) using a pop acceptance procedure.
1) If the offspring dominates one or more population members, then the offspring replaces one of them (chosen at random).
2) If any population member dominates the offspring, it is not accepted.
3) When both the above tests fail, the offspring replaces a randomly chosen population member.

Step6: Solution *c* is included in *A*(*t*) using an archive acceptance procedure based on ε-dominance

Step7: If termination criterion is not satisfied, set *t* = *t* + 1 and go to Step 2, else report *A*(*t*).

Output: archive population (t)

Figure 2.1 General framework of **ε**-MOEA

Based on the ε-dominance relation ε-MOEA is a steady-state algorithm. It breaks down the objective space into hyperboxes of size ε. Each hyperbox can contain at most a single solution on the basis of ε -dominance. From [13], ε -MOEA provides a tradeoff amongst convergence, diversity, and also computational time. Additionally, we can make it interactive with a decision-maker which refers that, ε can be chosen by a decision-maker according to user's preference.

ε-MOEA applies ε-dominance to direct the search towards the true Pareto front and ε-dominance plays a similar role like Pareto dominance in aiding the convergence of the population in low-dimensional space. Nonetheless, an improper choice of ε value will result in a poor performance of the algorithm. Furthermore, in the evolution process, ε-dominance can only be applied during the selection stage and still there is no help to handle the ineffectiveness of recombination operators caused by the large search space.

The diversity is kept by restricting each hyperbox with at most a single solution. Therefore, solutions are bound to have a minimum distance ε between them. By doing this we can only solve the distribution problem among solutions; as the spread of population cannot be improved. Moreover, a good distribution still requires a perfect ε value. If ε is way too small, solutions will be crowded with others. Whereas, a larger ε will eliminate more solutions in the beginning of the evolution process which is not ideal. Figure 2.1 shows the general framework of ε-MOEA.

2.2 Decomposition Based MOEAs

The second class is decomposition based designs, like Multiple Single Objective Pareto Sampling (MSOPS), and Multiobjective Evolutionary Algorithm based on Decomposition (MOEA/D) [15]. This type of designs decomposes a multi-objective optimization problem into a number of single optimization problems and predefines a group of search directions corresponding to these single optimization problems in order to optimize them concurrently. During the evolution process, Tchebycheff [15] and Achievement Scalarizing Function [5] can be applied as a fitness

assignment. The fitness values are used in selecting individuals instead of using Pareto-Dominance. For the very reason, this method can be adaptable to solve MaOPs.

In comparison with Pareto dominance, a group of weight vectors (search directions) are defined in advance and aggregate all objective values to push the population towards the true Pareto front; simultaneously, a solution can be recombined with another only if they are neighbors, which curtails a lot of difficulties from the large search space. However, the performance of algorithm is relies primarily on the selected aggregation method. When optimizing different problems, different aggregation methods must be chosen for each problem. For example, as stated in [28], weighted sum method is more efficient for convex problems while Tchebycheff method is advisable in nonconvex problems. Also, the number of weight vectors is equally important in respect to the performance. If this number of weight vectors is too small, each solution is very much different from solutions in its own neighborhood. As mentioned earlier, when recombining (crossover) two distant parents, chances of obtaining a good offspring are very low and the difficulty from large search space still exists. On the other hand, if the number of weight vectors is too large, it takes a lot of computational budget. In addition to that, in the high-dimensional search space, no one really knows in advance as to, how many weight vectors are sufficient for the evolution process, as we don't know what a suitable population size is for a high-dimensional search space. Moreover, from a large search space if the neighborhood size is too large it will make so much harder for the algorithm to perform well.

The diversity of the population is maintained by a group of well distributed weight vectors or reference points. In the evolution process, each weight vector directs the respective individuals towards their reference point in the true Pareto front. However, well distributed weight vectors and sub problems cannot ensure that, their corresponding optimal solutions are well distributed too. In a high-dimensional search space, there could be one single solution, which is an optimal solution for multiple sub problems, which cause a severe damage to the population diversity [19].

Also, the size of the neighborhood still affects the diversity performance. This indicates that a smaller neighborhood size cannot ensure a good population diversity.

In summary, both convergence and diversity power are mainly dependent on well distributed weight vectors and the corresponding neighborhood. However, the difficulties of setting the number of weight vectors, neighborhood size, and the choice of aggregation method, make the algorithm harder to obtain good convergence or diversity performance.

Input: An MOP and a stopping criterion; (the number of subproblems);
$\lambda^1, \lambda^2, \dots, \lambda^N$(a uniform spread of N weight vectors); T(the number of weight vectors in neighborhood of each weight vector)

Step1: Initialization
1) Set$EP = \emptyset$.
2) Compute the Euclidean distances between any two weight vectors and then work out the T closest weight vectors to each one. $\forall\, i = 1, \dots, N$, set $(i) = \{i_1, \dots, i_T\}$, where $\lambda^{i1}, \dots, \lambda^{iT}$ are the T closest weight vectors to λ^i
3) Generate an initial population $x^1, \dots, x^N$ randomly. Set $FV^i = F(x^i)$
4) 4) Initialize $z^* = (z_i^*, \dots, z_m^*)^T$ by a problem-specific method.

Step2: Update For $i = 1, \dots,$
N, do
1) Reproduction: Randomly select two indexes , l from $B(i)$, and then generate a new solution y from x^k and x^l by using genetic operators.
2) Improvement: Apply a problem-specific repair/improvement heuristic on y to produce y'
3) Update of z: for each $j = 1, \dots, m$, if $z_j < f_i(y')$, then set $z_j = f_i(y')$, .
4) Update of Neighboring Solutions: for each index $j \in B(i)$,if $a^{te}(y'\lambda^j, z) < (x^j\lambda^j, z)$, set $x^j = y'$ and $FV^j = (y')$
5) Update of *EP*: Remove *EP* from all the vectors dominated by (y') and add (y') to *EP* if no vectors in *EP* dominate $F(y')$.

Step3: Stopping Criteria
1) If stopping criteria is satisfied, then stop and output *EP*. Otherwise, go to Step2.

Output: External Population (EP)

Figure 2.2 General Framework of MOEA/D

2.3 Grid Based MOEAs

The third class is the grid-based method. In [16], a grid reflects the status of the convergence and diversity at the same time. Grid-Based Evolutionary Algorithm (GrEA) [16] focuses on the potential of the grid-based approach to increase the selection pressure towards the optimal direction, while maintaining exclusive uniformly distributed solutions. Territory Defining Multi-objective Evolutionary Algorithm (TDEA) [17] marks a territory around each individual to prevent the crowding problem. However, decision maker does not decide the hyperbox of TDEA; it is related to the individuals.

Grid based method employs grid coordinates in the process of evolution. This approach increases the selection pressure towards the global Pareto front simultaneously conserving an extensive and uniform distribution among solutions. As outlined in [16], when compared with Pareto dominance, a grid-based criterion can not only compare solutions qualitatively but also gives the quantitative comparison of objective values of the solutions. This characteristic provides main convergence power. However, the choice of the size of hyperbox and grid parameter can be a challenge. Meanwhile, the selection pressure towards the true Pareto front is hard to increase, as direct use of grid coordinates is not possible. On the contrary, it still requires aggregation methods or some fitness assignment techniques to handle these grid coordinates.

Like NSGA-III, Grid based method employs the idea of fitness sharing to maintain diversity. Here, based on its objective values population is divided into different boxes. In addition, we degrade each individual's fitness if its hyperbox contains multiple individuals. Different methodologies implement different strategies; ε-MOEA minimizes one individual for each hyperbox, while GrEA only punishes individuals in the crowded hyperbox.

2.3.1 GrEA [16]

Input: (the size of population)*;* a MOP and stopping criteria

Step 1: Initialization

N Individuals are randomly generated to form an initial population P

Step 2: Grid setting (Grid environment for current population is setting)

For population P, $\min(P)$ is the minimum value and $\max_k(P)$ is the maximum value of the kth objective, div is the number of the divisions of objective space in each dimension, then

Lower Bound: $lb_k = \min(P) - \max_k(P) - \min_k(P)/(2 \times div)$

Upper Bound: $ub_k = \max(P) + \max_k(P) - \min_k(P)/(2 \times div)$ Grid width:

$d_k = (ub_k - lb_k)/div$

Step 3: Fitness assignment

1) Convergence Estimator (GR)

M

$(x) = (x)_{k=1}$

The grid coordinate of x in the kth objective is: $(x) = \lfloor (f_k(x) - lb_k)/d_k \rfloor$, where $f_k(x)$ is the actual objective value of an individual x in the k th objective,

2) Density Estimation (GCD)

$GCD(x) = \sum_{y \in N(x)} M - GD(x, y)$

(x) is the neighbor of a solution x. A solution y is regarded as a neighbor of a solution x if $(x, y) <$. Grid difference $(x, y) = \sum_{k=1}^{M} |(x) - G_k(y)|$

3) Normalized Euclidean Distance ($GCPD$)

$(x) = \sum_{k=1}^{M} (x) - (lb_k + (x) \times d_k)/d_k^2$

$GCPD$ is the Euclidean distance between an individual and the utopia point in its hyperbox

Step 4: Mating selection (picks up promising solutions for variation)

1) Check Grid dominance and Pareto dominance Grid dominance:

Let $x, y \in P$, give M objectives, x grid-dominates y:

$x \prec_{grid} y \Leftrightarrow \forall i \in (1,2, \dots, M)$: $G_i(x) \leq G_i(y)$ and $\exists j \in (1,2, \dots, M)$: $G_j(x) \leq G_j(y)$

2) Prefer lower Density Estimation Value (GCD)

Step 5: Environmental selection

Choose the best N individuals into archive and generate (t)

Output (t) if stopping criteria is satisfied; otherwise, go back to step 2.

Output: archive population (t)

Figure 2.3 General Framework of GrEA

2.4 Performance Indicator Based MOEAs

The fourth class is based on the idea of developing the algorithms according to the quality indicators. These quality indicators aim at unfolding advantages and shortcomings of the state-of-the-art MOEAs and they determine the best performance associated to the specific problem characteristics by assigning a specific fitness measure for every individual. For instance, Volume Dominance (VD) [20] assigns fitness value which is equivalent to the volume dominated in the objective space by that particular individual. Contraction/Expansion of Dominated Area (CE) [21] regulates the selection process by altering the size of the respective individuals' dominance area as well as the distance to the best known solution. Whereas, GB [22] evaluates the best reference point's value that dominates the entire population size. Hypervolume Estimation Algorithm for Multiobjective Optimization (HypE) [23], is probably the most successful implementation of this class in that it has been shown to be more effective than other MOEAs for MaOPs. Also, there are a few other designs in a similar fashion like Indicator-Based Evolutionary Algorithm (IBEA) [24] and *S* Metric Selection Evolutionary Multiobjective Optimization Algorithm (SMS-EMOA) [25].

This type of designs implements any of the modified version of performance indicators to directly assign each individual a fitness value. The assigned fitness value should reflect both convergence and diversity performance of the individuals concurrently. In literature, these performance/quality indicators look at answering three main objectives [26]: minimizing the distance between the obtained non-dominated set to the true Pareto-Front, obtaining a uniformly distributed solution set in the objective space, and maximizing the obtained non-dominated front. Generally, all those indicators assign a value for the whole approximation front which reflect their performance. Here, these performance indicators are adjusted in such a way that, every individual of the approximate front is assigned a value based on its sole contribution to the overall

performance. Hence, the fitness value is directly proportional to the optimization, and instead of Pareto dominance it can be employed for selection of individuals and environmental mating (crossover/mutation) in the evolution process.

Nonetheless, many performance indicators alone cannot faithfully measure MOEA performance [27], the assigned fitness value by most of the indicators can only provide one perspective of the performance, but it is inconsistent in other perspectives of the performance. Which, results in using different quality indicators under some specific conditions. Furthermore, few indicators oppose Pareto-dominance by assigning the dominated individual a better score over the non-dominated one. Therefore, the selection of indicator is very important for the algorithm. Besides, even if the indicator is chosen soundly, this method only addresses a way for fitness assignment and it cannot limit the problems related to large search space.

2.4.1 HypE [15]

Input: a MOP and stopping criteria; (the size of population); (reference set);

Step1: Initialization
Initial population P by selecting N solutions uniformly at random

Step2: Mating selection
1) Hypervolume-based fitness value Estimating b y Monte Carlo Simulation
2) Choose parents P' according to hypervolume-based fitness

Step3: Variation
Generate offspring P'' from chosen parents P'

Step4: Environmental selection
1) Combine P'' and P and select best N individuals into archive population $A(t)$ based on Pareto-dominance and hypervolume-based fitness values
2) Output $A(t)$ if stopping criteria is satisfied; otherwise, go back to step 2

Output: archive population (t)

Figure 2.4 General Framework of HypE

HypE is a hypervolume-based many-objective evolutionary optimization algorithm. It employs Monte Carlo simulation to evaluate the exact hypervolume value, and assigns ranks to the solutions induced by the hypervolume indicator. These ranks of the solutions are utilized in calculating fitness values, mating selection, and environmental selection. On a whole, it balances the accuracy of the estimates and the computational budget involved in the Hypervolume calculation. The modified hypervolume indicator can assign each individual a fitness value which reflects both convergence and diversity performance of that particular individual. Figure 2.4 shows the general framework of HypE.

2.5 Diversity-Emphasis Method

In this type of designs, diversity becomes the main criteria in the evolution process rather than convergence. Convergence is maintained by either the Pareto dominance or other methods which help in pushing the non-converged individuals into more crowded area [29].

When the Pareto domination resistance occurs in the higher dimensional space: all the individuals are non-dominated among each other, this sort of designs transforms the work directly from optimizing both convergence and diversity to improve only the diversity. So it is called the diversity-emphasis method.

2.5.1 NSGA-III [5]

NSGA-III is a hybrid algorithm, which has a framework similar to the original NSGA-II except it includes a refined diversity preservation technique especially for handling many-objective optimization problems instead of using crowding distance method. This technique works in this manner: at each generation, an ideal point is determined by finding the minimum value in each objective function among all current solutions; after that, objective values of the solutions are

rendered by deducting the original objective value from the ideal point. Later on, we identify the extreme solutions by Achievement Scalarizing Function (ASF); the identified extreme solutions and ideal point contain a hyper-plane. Next, many well-distributed reference points are generated on the hyper-plane prior to the evolution process. Every individual is associated with one of the reference points. Niche count of each reference point is evaluated, by determining the number of its respective population member associated. Finally, the population members associated with the reference point that has the smallest niche count value at the last accepted non-dominated front are selected.

Input: An MOP and a stopping criterion; (population size); (reference points)

Step1: Initialize population

A random parent population P_0 is created. The population is sorted based on the fuzzy Pareto dominance. Each solution is assigned a rank equal to its Pareto domination level where 1 is the best level. Binary tournament selection, SBX recombination, and polynomial mutation operators are used to create a child population Q_0 of the same size N.Set $t = 0$

Step2: $R_t = P_t \cup Q_t$

Combine parent and children population. The population R_t will have size $2N$.

Step3: *F*=Non-nondominated-sort (R_t) until $|P_{t+1}| < N$.

Population R_t is sorted based on the non-domination sorting. The new parent population P_{t+1} is formed by adding solutions from the first front to the next best front (F_l)before the size exceeds N.

Step4: Choose $N - |P_{t+1}|$ solutions from F_l

1) Construct the hyperplane and normalize objectives of each solution
2) Associate each member with a reference point of Z^r
3) Compute niche count of each reference point
4) Sort reference points in descending order with the number of associated population Members. Identify the first $N - |P_{t+1}|$ reference points and choose their associated population members to construct new population
5) If stopping criterion is not satisfied, go back to Step2; otherwise, output P_{t+1}

Output: P_{t+1}

Figure 2.5 General Framework of NSGA-III

Similar to NSGA-II, NSGA-III also applies Pareto dominance to sort the individuals and assign rank to every individual. Instead of relying on crowding distance at the final level, NSGA-III picks up the non-dominated point associated with the reference point with less niche count. This approach is efficient in maintaining diversity in the solutions and avoids the condition that happens with most of the aggregation methods: which means one single solution can be the optimal solution of many single objective optimization problems. In addition to that, this idea is very much like the concept of fitness sharing, which is a classic niching technique [30] and is used in punishing those solutions in the crowded area. That is, the population is sectioned into distinct groups of populations based on the similarity of the individuals in the population, in such a way that the population member is associated with the respective reference point in NSGA-III. Then, every individual's fitness is decreased if more than one individual shares the same subpopulation with the reference point. Which means, in NSGA-III, individuals related to the reference point whose niche count is smaller have more chance to be chosen. Figure 2.5 shows the generic framework of NSGA-III.

2.6 Summary

Usually, MOEAs have five major steps: population initialization, fitness evaluation, selection, recombination (crossover and mutation), and environmental selection. Recently proposed algorithms for solving MaOPs mainly focus on a modified technique for fitness assignment. For instance, ε-MOEA and algorithms of this type generally modifies the Pareto-dominance, HypE however, uses the hypervolume indicator, MOEA/D employs weight vectors, NSGA-III emphasizes on diversity performance, and GrEA incorporates a grid. Some of them also addresses few other aspects. For example, MOEA/D limits the selection, mutation and crossover to be done only in the neighborhood of the particular individuals. Meanwhile, NSGA-III choose large distribution index in recombination step. Table 2.1 is the summary of each type of MOEAs.

Table 2.1 Summary of Each Type of MOEAs

Type	MOEAs	Characteristics	Effort
Dominance Modification	ε-MOEA [10]	New dominance relations	Fitness assignment
Indicator	HypE [15]	Quality indicator	Fitness assignment
Decompose	MOEA/D [18]	Weight vectors	Fitness assignment
		Neighborhood	Selection & combination
Diversity Emphasis	NSGA-III [5]	Diversity as main criteria	Fitness assignment
Grid-based	GrEA [19]	Grid-based criteria	Fitness assignment

2.7 Performance Assessment

Researchers and Practitioners contributed to the Evolutionary Multi/Many-Objective Optimization literature in the design of performance criteria and methodologies for assessing performance of Evolutionary Multi/Many-Objective Optimization algorithms. It is necessary to select a relevant and effective method of performance assessment for either evaluating or comparing Evolutionary Multi/Many-Objective Optimizatx`ion algorithms. We use these methods of performance assessment to measure an algorithm's performance with respect to the convergence, diversity, and pertinence of the final approximation set. Most amongst the many methods of performance assessment rely on reference points or any equivalent to reference points. This dependency is not achievable in real-world problem scenarios, even more unlikely in the case where the problem is new and has not been subjected to optimization.

In this section, we explain three widely used performance metrics. Subsection 2.7.1 explains the Hypervolume Indicator metric, Subsection 2.7.2 outlines the Generational Distance metric, and Subsection 2.7.3 elaborates the Inverted Generational Distance metric.

2.7.1 The Hypervolume Indicator

The hypervolume indicator also known as “s-metric” is a performance metric introduced by [31] where it is described as the “size of the space covered or size of dominated space” for indicating the quality of a non-dominated approximation set,. Defined as below [32]:

$$s_{f^{ref}}(\mathrm{x}) = \Lambda\left(\cup_{\mathrm{x}_n \subset \mathrm{x}} \left[f_1(\mathrm{x}_n), f_1^{ref}\right] \times \cdots \times \left[f_m(\mathrm{x}_n), f_m^{ref}\right)\right) \quad (1)$$

Here, $s_{f^{ref}}(\mathrm{x})$ resolves an issue associated with the size of the space covered by an approximation set x, $s_{f^{ref}}(\mathrm{x}) \subset \mathbb{R}$ refers to a particular reference point chosen beforehand and $\Lambda(.)$ reflects the Lebesgue measure. This has been illustrated in Figure 2.6 in two-dimensional objective space (to allow for an easy visualization) with a population of 3 solutions.

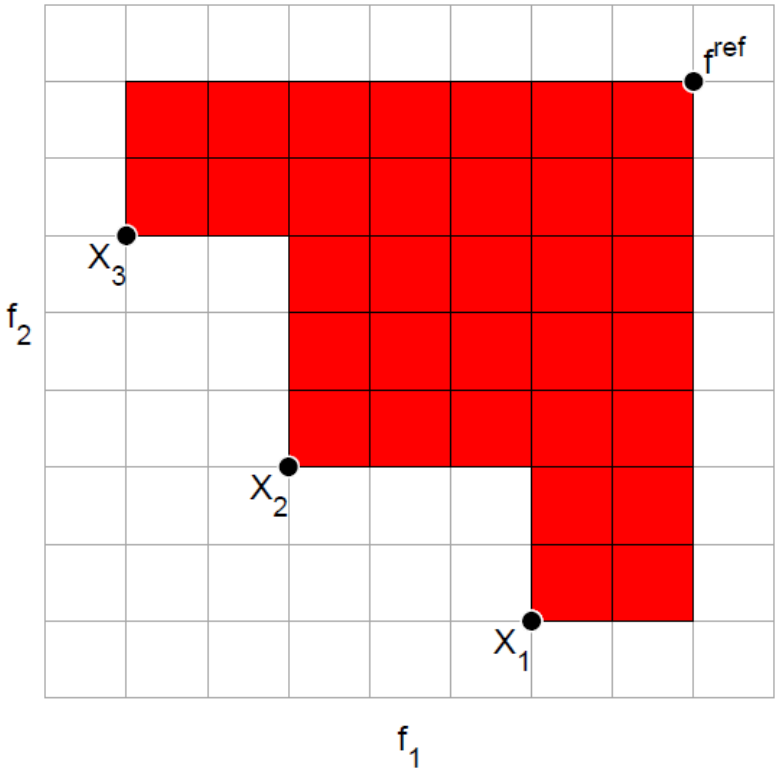


Figure 2.6 an example of the hypervolume indicator in two-dimensional objective space

The hypervolume indicator is appealing because it is scaling independent and requires no prior knowledge of the true Pareto-optimal front. This is important when working with real-world problems which have not yet been solved. The hypervolume indicator is currently used in the field of multi-objective optimization as both a proximity and diversity performance metric, and in the decision making process as well [33, 34].

A vector of reference points is necessary to evaluate the hypervolume indicator value. When on comparison with a couple or multiple algorithms at the same time, reference vector employed must be the same; otherwise, the final hypervolume indicator values are not valid. In order to include the entire objective values in any approximation set to have a place in reference vector, we approximate the reference vector with all large values of each objective. A much better and accurate choice of a reference vector is to select the worst objective values from the union of approximation sets obtained on a particular test problem for different state-of-the-art evolutionary algorithms. Numerous versions of implementations of the use of hypervolume indicator have been presented in [35, 36, 37, 38, 39], all with the aim to aid its evaluation and computational time. The hypervolume indicator has been employed in the performance assessment of algorithms in much of the multi-objective optimization and evolutionary computation literature (e.g. [40, 41, 42, 43, 44]).

2.7.2 The Generational Distance

The Generational Distance (GD) introduced in [45, 46] measures the proximity of the approximation set to the true Pareto-optimal front in the objective space. We can express GD as given below:

$$GD = \frac{\sqrt{\sum_{i=1}^{n^*} d_i^2}}{n^*} \qquad (2)$$

where n^* is the number of solutions in the approximation set, and d_i represents the Euclidean distance (in objective space) amongst every solution in the approximation set to the nearest member on the true Pareto-Front. A zero GD value represents that every individual present in the approximation set are on the true Pareto-optimal front, and any value above zero illustrates the magnitude by which the approximation set is deviated from the true Pareto-optimal front. Calculation of GD is straightforward, at the same time the concept is very intuitive. However, prior knowledge concerning the true Pareto-optimal front is necessary for the reference vector. The

selection of solutions for the reference set will have an impact on the results obtained from the GD, and therefore the reference set must be diverse. In addition, the calculation of the GD can be computational expensive when working with a large populations or a high number of problem objectives. The GD measure has been widely employed to assess the performance of algorithms in most of the multi-objective optimization and in evolutionary computation literature (e.g. [47, 48, 49, 50, 51, 52]).

2.7.3 Inverted Generational Distance

Following with another performance metric, introduced an Inverted Generational Distance (IGD) which correct a known deficiency in the GD measure, measuring the proximity of the approximation set to the true Pareto-optimal front in the objective space. Below is the expression of IGD:

$$IGD = \frac{\sqrt{\sum_{i=1}^{n^*} d_i^2}}{n^*} \tag{3}$$

Here, n^* represents the number of solutions in the reference set, and d_i represents the Euclidean distance (in objective space) amongst every solution in the approximation set to the nearest member on the true Pareto-Front. A zero IGD value represents that every individual present in the approximation set are on the true Pareto-optimal front, and any value above zero illustrates the magnitude by which the approximation set is deviated from the true Pareto-optimal front. This version of IGD implementation addresses an issue associated with its predecessor, which is that it will not rate an entire approximation set based on a single solution on the reference set as better than an approximation set which has more non-dominated (better) solutions that are much closer to the reference set. Yet, like the GD measure, prior knowledge concerning the true Pareto-optimal front is necessary in order to form a reference set. Moreover, solutions selected for the reference

set will have a huge impact on the results obtained from the IGD, and accordingly the reference set considered must be diverse. The IGD calculation can be computational expensive when working with either large reference sets or when high number of objectives are considered for the particular problem. The IGD measure has been exploited to assess the performance of the algorithms in most of the multi-objective optimization and evolutionary computation literature (e.g. [53, 54, 55, 56, 57]).

2.8 Benchmark Test Problems

In order to assess the performance of several nature inspired optimization algorithms, researchers and practitioners proposed numerous benchmark test problems. This section analyzes a few widely known and used benchmark test problems for large-scale optimization, multi- and many-objective optimization.

2.8.1 Benchmark Test Problems for Large-Scale Optimization

Tang *et al* proposed the first benchmark suite for large-scale optimization. in the CEC'2008 special session and competition on large-scale global optimization [62], known as the CEC'2008 suite, which consists of seven test problems. This is the first attempt where characteristics like separability and non-separability are explicitly included for large-scale optimization test problems. However, they roughly designed the problems to be either separable or non-separable; which is a major limitation of this test suite as this represents only two extreme cases of the most existing real-world problems.

On the verge of improving the CEC'2008 test suite, Tang *et al.* later proposed the CEC'2010 test suite [63]. The major advantage of CEC'2010 test suite over the previous one is that, CEC'2010 test suite employs modularity design principle, which in turn divides the decision variables into several subcomponents, and the separability of every subcomponent is independent to each another.

In the same way, with different combinations of separable and non-separable subcomponents, the test problems can be either fully partially separable, separable or fully non-separable. With the proposal of the CEC'2010 test suite, it has successfully motivated the development of techniques such as random grouping [64], cooperative co-evolution [60], and differential grouping [65].

As a further improved version of the CEC'2010 test suite, they included some new characteristics [66] and proposed the CEC'2013 test suite. Firstly, the subcomponents of the decision variables in the CEC'2010 test suite are completely independent, while in the CEC'2013 test suite, some subcomponents are overlapped. Secondly, all the subcomponents in the CEC'2010 test suite have a fixed size, while in the CEC'2013 test suite; the subcomponents are of different sizes, such that they have non-uniform contributions to the objective function.

2.8.2 Benchmark Test Problems for Multi- and Many-objective Optimization

Many researchers designed test suites for empirical studies to evaluate the performance of various MOEAs. Among many others test suites, the ZDT test suite [67], DTLZ test suite [68,69] and the WFG test suite [70, 71] are the most widely known and used ones. The ZDT test suite is one of the most popular if not the most popular test suites in the multi-objective optimization literature [67]. Deb based on the generic design principles in [72], proposed the test problems in the ZDT suite. By introducing three basic functions, including a distribution function $f1$, a distance function g and a shape function h are constructed. Where function $f1$ represents the ability to test an MOEA to maintain the diversity along the PF. While, function g is meant for testing the ability of an MOEA to converge to the PF and function h for defining the shape of the PF. On a whole, ZDT test suite consists of six test problems, five of them (ZDT1 to ZDT4, ZDT6) are real-coded and one (ZDT5) is binary-coded. Characteristics of ZDT test problems differ from one problem to another. Typically, ZDT3 has a disconnected Pareto-Front: which is partially convex and partially concave; ZDT4 consists of a large number of local PFs; ZDT6 has a fitness landscape, which is non-uniform in nature, which eventually cause a biased distribution of the Pareto optimal solutions along the PF.

Although, the ZDT test suite has gained immense popularity over the years, it still has a significant limitation, i.e., all test problems are bi-objective (two objectives).

To overcome the shortcomings of the ZDT test suite (as well as many other bi-objective test problems), Deb *et al.* proposed a new test suite, i.e., the DTLZ test suite [68, 69], which contains test problems which are scalable to any desired number of objectives. DTLZ test suite contains nine test problems; in constructing every one of them, utilized design principle is the same. Where, the first ($M-1$) decision variables describe the Pareto Optimal Front, while the remaining decision variables define the convergence property. The DTLZ test suite also have many exclusive characteristics. For example, the majority of the fitness landscape of DTLZ1 and DTLZ3 consists of a large number of local PFs; the distribution of the Pareto optimal solutions of DTLZ4 is usually non-uniform; the Pareto-Fronts of both DTLZ5 and DTLZ6 are a degenerate curve; DTLZ7 has a broken (not continuous) Pareto-Front; and DTLZ8 and DTLZ9 are constrained problems. A significant contribution of the DTLZ test suite would be the proposal of a generic design principle for constructing test problems that are scalable to have any number of objectives, as well as decision variables.

Many researchers have developed additional alternatives of the DTLZ test suite. To assess the performance of MOEAs on highly scaled problems [61], Deb *et al.* proposed a method to scale the value of each objective function to a different range. [73] suggests, some constrained DTLZ problems to verify the MOEAs constraints handling capacity. Nonetheless, the fact that DTLZ test suite is extensively used in the computational intelligence community it still has a few drawbacks. For instance, DTLZ test suite does not take some important characteristics commonly seen in the real-world problems into account, such as variable linkage and variable separability. In this context, separability means the correlation relationship amongst the decision variables in the entire decision space, while to characterize the relationship between the decision variables of Pareto optimal solutions they use variable linkage.

To avoid the deficiencies of the DTLZ test suite, Huband *et al.* proposed a whole new test suite, i.e., the WFG test suite [70, 71]. The WFG test suite employs an array of important characteristics that generally exist in real-world problems. In order to design a test problem, it only requires specification of a shape function, which in turn finds the PF, and a transformation function, which determines the fitness landscape. In the WFG test suite, test problems WFG1, WFG7 and WFG9 have partial PFs; WFG5 and WFG9 have misleading fitness landscapes; WFG2, WFG3, WFG6, WFG8 and WFG9 have fitness landscapes, which are non-separable. Since these test problems are also compatible with scalability concerning number of objectives, the WFG test suite becomes another extensive benchmark for many-objective optimization, apart from the DTLZ test suite.

In addition to the above three general-purpose test suites, Researchers constructed other test problems to include some specific characteristics. In [74], Okabe *et al.* proposed quite a few design principles to construct test problems with an arbitrarily complex Pareto-Set, generalized from [58]. In recent years, Saxena *et al.* have extended the same test problems with complicated Pareto-Sets for many-objective optimization problems [75]. Few other modification versions of ZDT and DTLZ test problems can be found in [59] [76], where linear or nonlinear variables are introduced into decision variables. Although, the test problems detailed above are static, some Researchers proposed dynamic multi-objective optimization test problems in [77], where the PFs and/or PSs keep changing with respect to time. In [78, 79] some variants of these test problems can also be found.

CHAPTER III

PROPOSED ADVANCED *p*-METRIC BASED MANY-OBJECTIVE EVOLUTIONARY ALGORITHM

In this section, we propose an advanced *p*-metric based Many-Objective EA (APMOEA) for tackling both MOPs as well as MaOPs. We employed a two-phase selection strategy, which unites both non-dominated sorting and *p*-metric as well. During the first phase, we use non-dominated sorting to remove solutions with poor convergence. Despite the fact that non-dominated sorting often fails in many-objective cases, it can help eliminate the far-off individuals and assure the stability of the convergence. In the second phase, *p*-metric comes into picture to find the final set of reserved solutions, while we also proposed a modified method to maintain a proper set of the direction vectors of *p*-metric. We summarize the contributions as follows:

1) On comparison with a few other indicator-based EAs, APMOEA does not include additional parameters. In addition to that, due to the aggregate effect of non-dominated sorting and p-metric, APMOEA is very efficient in tackling both MOPs and MaOPs.
2) In APMOEA, we propose a modified performance-contribution evaluation method of *p*-metric, in order to assist the enhancement of the diversity maintenance. That very adjustment makes *p*-metric possible to adapt to the situations for evolving the population.

3) During the evolution process, in the *p*-metric based selection the direction vectors set keeps changing with respect to time, in accordance with the current population distribution in the objective space.

4) In the experiment setup, we test on seventeen problems, among three-, five- and ten-objective cases, with respect to the three performance indicators. Four of the problems included are regular test problems and are widely-used, while the rest of the 13 problems, specifically designed for the CEC 2017 competition, have either irregular, degenerate, disconnected, badly-scaled, mixed, (or) many other complex PFs. On comparison with four other most widely-used state-of-the-art Many-Objective EAs, the empirical results identifies that APMOEA has promising adaptability in tackling both MOPs and MaOPs with different types of PFs.

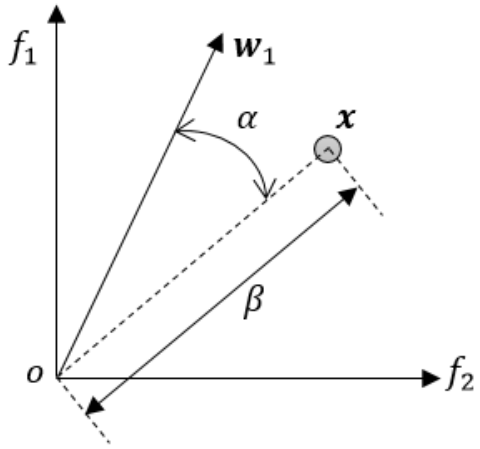


Fig. 3.1 Example of the angular and radial distances.

3.1 *p*-metric Measurement Method

In 2016, He and Yen proposed Polar-Metric (*p*-metric) to measure the performances of the state-of-the art multi-objective and many-objective EAs. A preset set of direction vectors in the objective space plays a major role in the p-metric measurement. First, we uniformly sample a set of points on the PF and then we calculate a global ideal point by getting the minimum value of each objective

vector. Later, a vector between each generated sample point and the calculated global ideal point forms a direction vector. Utilizing direction vectors, we calculate the angular distance (AD) of a solution $(\boldsymbol{x})$ by the following formula:

$$AD(\boldsymbol{x}) = \min_{w_1 \cdots w_k} (1 - \cos(\boldsymbol{F}(\boldsymbol{x}), (\boldsymbol{w_1} \cdots \boldsymbol{w_k}))) \quad (1)$$

Here, $\boldsymbol{F}(\boldsymbol{x})$ represents the objective vector of $\boldsymbol{x}$, while $\boldsymbol{w_1} \cdots \boldsymbol{w_k}$ denotes $\boldsymbol{k}$ direction vectors, and cos represents cosine value of the acute angle between the two vectors in brackets. Then, $AD(\boldsymbol{x})$ evaluates how close $\boldsymbol{x}$ is to all the direction vectors $(\boldsymbol{w_1} \cdots \boldsymbol{w_k})$ and gives the minimum value associated with the closest direction vector ($\boldsymbol{w_1}$ for $\boldsymbol{x}$) in the objective space. With the help of angular distances, every solution in the objective space is associated with the closest direction vector. For each directional vector, we find the associated solution with the smallest radial distance possible. The radial distance $RD(\boldsymbol{x})$ is calculated as follows:

$$RD(\boldsymbol{x}) = ||(\boldsymbol{F}(\boldsymbol{x})||, \quad (2)$$

Where, $||.||$ expresses the Euclidean distance of the vector in brackets. Then, we find the inverse of the smallest radial distance (i.e., $1/RD$) which reflects the performance contribution of the direction vector associated. Finally, the aggregation of all the performance contributions accounts for the *p*-metric score.

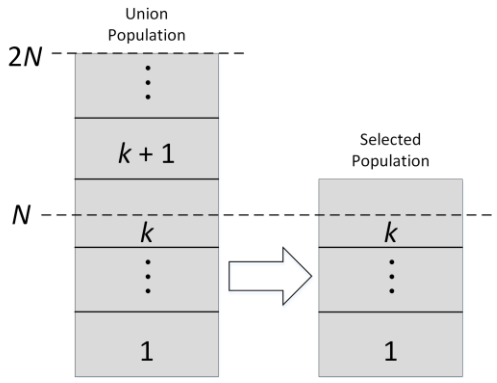


Fig. 3.2 Example of the first selection phase.

Fig. 3.1 explains both the angular distance and radial distance in a bi-objective space, where $\boldsymbol{w_1}$ being a direction vector and $\boldsymbol{x}$ is one of the solution. According to (1) and (2), an acute angle α represents the angular distance between $\boldsymbol{w_1}$ and $\boldsymbol{x}$ in Fig 3.1, while the radial distance of $\boldsymbol{x}$ is equal to the distance β. In actuality, there are three different types of radial distance calculations employed in p-metric, corresponding to three different Pareto-Front shapes. However, when we apply p-metric to evolution, learning the shape of the PF beforehand is highly unlikely. Therefore, we only employ Euclidean distance type of radial distance measurement, universally used for the proposed algorithm.

3.2 Proposed Algorithm: APMOEA

In this section, we first emphasize two of the selection phases, the first selection phase based on non-dominated sorting and the second selection phase based on p-metric, respectively. Then, we outline the general framework of APMOEA.

3.2.1 First Selection Phase

In the first selection phase, the idea of using non-dominated sorting is to remove the remote solutions, in order to enhance the convergence stability. [82] Explains that we can sort a population into a series of non-dominated fronts, while solutions in the same front are non-dominated to each other and those in the lower-rank front dominate solutions in the higher-rank front. Assume that there exists $2N$ solutions in the population, and N solutions are all we need for the next generation, shown is an example for selecting N solutions in Fig. 3.2.

In Fig. 2, we sort a total of $2N$ solutions into several non- dominated fronts, while the front number and front rank exhibits an inversely proportional relationship, (i.e., the greater the front number, the lower is the rank). To reserve a minimum of N solutions, we choose the first k fronts. The number of solutions present within k fronts should range between N and $2N$. Typically, while

tackling MaOPs, it is highly likely that all the solutions are stuck in one non-dominated front, making the first selection phase less effective. Nonetheless, for tackling MOPs the first selection phase is still valid, and can assist in getting rid of remote points to some degree.

3.2.2 Second Selection Phase

In the second selection phase, we use a set of uniformly distributed direction vectors, preset by the Das and Dennis's method [50], to select the final reserved solutions.

Algorithm 1 *p-Metric Select*(Pop, N, D)

Input: population, Pop; required selection number, N; direction vector set, D.
Output: selected solution set, S.

1) Initialize an empty set S to store all the selected solutions.
2) Associate each solution in Pop with the closest direction vector in D, according to the angular distance.
3) **for** each direction vector $\boldsymbol{w}_1$ in D **do**
4) Find the solution $\boldsymbol{x}^*$, associated with $\boldsymbol{w}_1$, with the minimum fitness value calculated by eqn. (3).
5) Add $\boldsymbol{x}^*$ into S.
6) **end for**
7) **if** $Length(S) < N$ **then**
8) Store all the unselected solutions of Pop into U.
9) Initialize an empty set D^* to store all the new direction vectors constructed below.
10) Copy S and U to new sets S^* and U^*, respectively.
11) **while** $Length(D^*) < N - Length(S)$ **do**
12) **for** each solution $\boldsymbol{x}$ in U^* **do**
13) Find the minimum angular distance of $\boldsymbol{x}$ to all the solutions in S^*.
14) **end for**
15) Find the solution $\boldsymbol{x}^*$ in U^*, with the minimum angular distance found above.
16) Delete $\boldsymbol{x}^*$ from U^*, and add $\boldsymbol{x}^*$ into S^*.
17) Delete any invalid direction vector in D,
18) And add $\boldsymbol{F}(\boldsymbol{x}^*)$ into D^*.
19) **end while**
20) Associate each solution in U with the closest direction vector in D^*, according to the angular distance.
21) **for** each direction vector $\boldsymbol{w}_1$ in D^* **do**
22) Find the solution $\boldsymbol{x}^*$, associated with $\boldsymbol{w}_1$, with the minimum fitness value calculated by (3).
23) Add $\boldsymbol{x}^*$ into S.
24) **end for**
25) **end if**

Fig 3.3 Algorithm 1

Algorithm 1 presents the pseudo-code of the second selection phase. For Algorithm 1, we feed a population *Pop* reserved from the first selection phase as input and an output of (final) selected solution set *S* of size *N* needs to be worked out. First, as shown in lines 1 to 6 in Algorithm 1, we select a maximum of *N* solutions by the original direction vectors according to the fitness value. We calculate the fitness (*Fit*) of each solution ($\boldsymbol{x}$) associated with the corresponding direction vector ($\boldsymbol{w}_1$) by the following expression:

$$Fit = \left(\frac{1}{RD(x)}\right) * \left(1 - \frac{AD(x)}{T}\right) \quad (3)$$

Here, radial distance of $\boldsymbol{x}$ is expressed as *RD*, while *AD* is the angular distance between $\boldsymbol{x}$ and $\boldsymbol{w}_1$, and *T* is the smallest angular distance of $\boldsymbol{w}_1$ to all the other direction vectors. Compared with the performance contribution of *p*-metric $\left(i.e., \frac{1}{RD(x)}\right)$ in fitness evaluation (Fit) we employ *AD* as an angular penalty, in order to provide some search pressure towards $\boldsymbol{w}_1$. In this way, one optimal solution will converge towards the intersection point between $\boldsymbol{w}_1$ and the PF, in the objective space, which results in balancing convergence and diversity efficiently. However, to select solutions, not all direction vectors have solutions associated. This problem arises because some direction vectors do not have nearby solutions in the search space. During evolution, it is highly likely to happen when either the PF shape is disconnected or irregular this can also happen when a direction vector is just within the infeasible area of the objective space.

Accordingly, in Algorithm 1 from lines 7 to 24 in order to get enough number (i.e., *N*) of solutions reserved for the next generation, we substitute those invalid direction vectors with a same number of newly constructed direction vectors, and utilize the latter ones to select more solutions. Lines 8 to 18 in Algorithm 1 shows the construction process of the new direction vectors, based on the spread of the solutions in the objective space. In addition to that, lines 19 to 23 in

Algorithm 1, which is similar to lines 2 to 6 shows the later stage of the selection by those newly, constructed direction vectors.

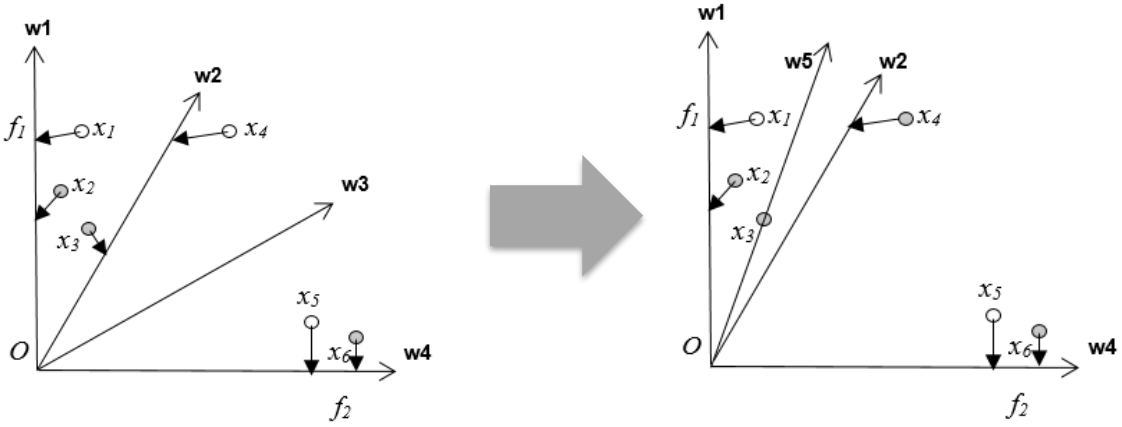


Fig 3.4 Example of the second selection phase (case-I).

Fig 3.4 shows an example of the second selection phase case-I based on *p*- metric, wherein we utilize four direction vectors in selecting four reserved solutions. Firstly, in the bi-objective space as shown on the left side of Fig 3.4 four original direction vectors are uniformly distributed, and six solutions are associated with the corresponding direction vectors, corresponding to the angular distance represented by the arrows. Then, the algorithm selects $\boldsymbol{x}2$, $\boldsymbol{x}3$ and $\boldsymbol{x}6$ by $\boldsymbol{w}1$, $\boldsymbol{w}2$ and $\boldsymbol{w}4$, respectively, according to (Fig 3.4), depicted by the bold circles. Yet, we still have to select one more solution because $\boldsymbol{w}3$ has no associated solution. Later we delete the invalid direction vector $\boldsymbol{w}3$, as shown on the right side of the Fig 3.4 and construct a new direction vector $\boldsymbol{w}5$, which passes through $\boldsymbol{x}3$. The reason behind not choosing $\boldsymbol{x}4$ to construct $\boldsymbol{w}5$ is that we consider convergence; also, $\boldsymbol{x}4$ contributes very less to the algorithm as it is a distant solution and we do not have any other associated solutions. Therefore, in order to maintain convergence, we select x3, x4 is associated to the w2 by angular distance calculation and is selected by w2 to aid

the sparse area of the feasible objective space. Thus, we select $\boldsymbol{x}4$ by $\boldsymbol{w}5$ instead of the other solutions, based on the extensive consideration of both diversity and convergence. This results in, $\boldsymbol{x}2$, $\boldsymbol{x}3$, $\boldsymbol{x}4$ and $\boldsymbol{x}6$ as the final reserved solutions.

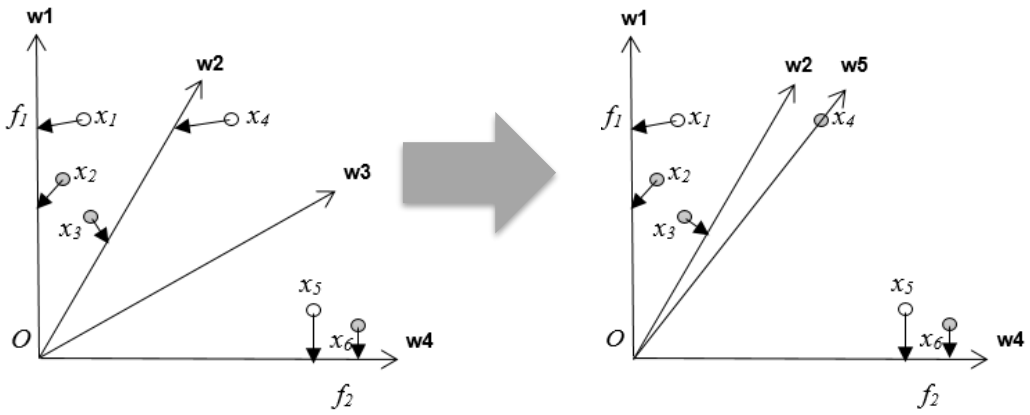


Fig 3.5 Example of the second selection phase (case-II).

Fig 3.5 shows an example of the second selection phase case-II based on *p*-metric, wherein we utilize four direction vectors in selecting four reserved solutions. Firstly, in the bi-objective space as shown on the left side of Fig 3.5 four original direction vectors are uniformly distributed, and six solutions are associated with the corresponding direction vectors, corresponding to the angular distance, represented by the arrows. Then, the algorithm selects $\boldsymbol{x}2$, $\boldsymbol{x}3$ and $\boldsymbol{x}6$ by $\boldsymbol{w}1$, $\boldsymbol{w}2$ and $\boldsymbol{w}4$ respectively, according to (Fig 3.5), depicted by the bold circles. Yet, we still have to select one more solution because $\boldsymbol{w}3$ has no associated solution. Later we delete the invalid direction vector $\boldsymbol{w}3$, as shown on the right side of the Fig 3.5 and construct a new direction vector $\boldsymbol{w}5$, which passes through $\boldsymbol{x}4$. The reason behind choosing $\boldsymbol{x}4$ to construct $\boldsymbol{w}5$ in this case is that we consider diversity at this stage of the evolution, $\boldsymbol{x}4$ accesses the feasible search space even though it contributes very little to the algorithm and we do not have any other associated

solutions. So, in order to maintain diversity we select $\boldsymbol{x}4$ by $\boldsymbol{w}5$ instead of the other solutions. Based on the extensive consideration of both diversity and convergence, which results in, $\boldsymbol{x}2$, $\boldsymbol{x}3$, $\boldsymbol{x}4$ and $\boldsymbol{x}6$ as the final reserved solutions.

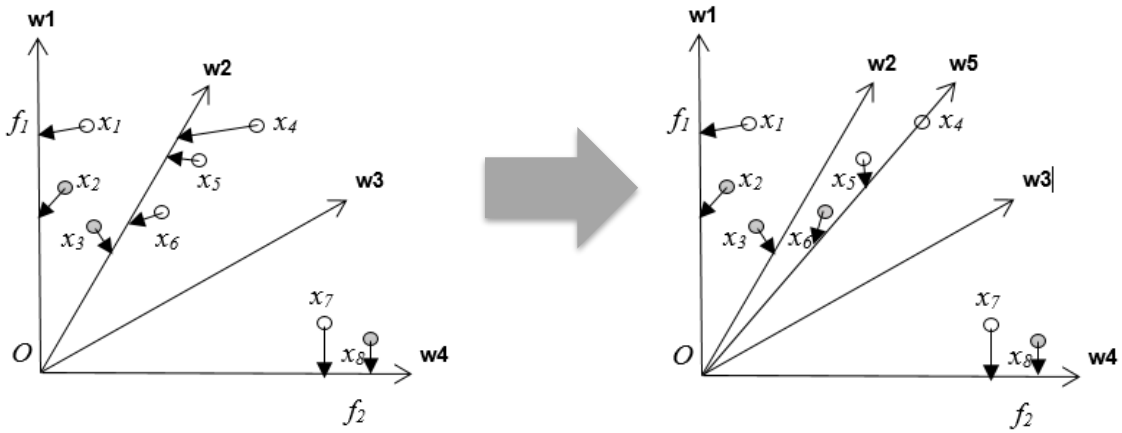


Fig 3.6 Example of the second selection phase (case- III).

Fig 3.6 shows an example of the second selection phase case-II based on *p*-metric, wherein we utilize four direction vectors in selecting four reserved solutions. Firstly, in the bi-objective space as shown on the left side of Fig 3.6 four original direction vectors are uniformly distributed, and eight solutions are associated with the corresponding direction vectors in this special case, corresponding to the angular distance, represented by the arrows. Then, the algorithm selects $\boldsymbol{x}2$, $\boldsymbol{x}3$ and $\boldsymbol{x}8$ by $\boldsymbol{w}1$, $\boldsymbol{w}2$ and $\boldsymbol{w}4$, respectively, according to (Fig 3.6), depicted by the bold circles. Yet, we still have to select one more solution because $\boldsymbol{w}3$ has no associated solution. Later we delete the invalid direction vector $\boldsymbol{w}3$, as shown on the right side of the Fig 3.6 and construct a new direction vector $\boldsymbol{w}5$, which passes through $\boldsymbol{x}3$. The reason behind choosing $\boldsymbol{x}4$ to construct $\boldsymbol{w}5$ in this case is that we consider diversity at this stage of the evolution and although we have other associated solutions unlike the above cases the angular distance is maximum compared to all the solution. Even after choosing $\boldsymbol{x}4$ to construct a direction vector we can choose between $\boldsymbol{x}4$ and $\boldsymbol{x}6$,

as we still need to maintain convergence; we select $\boldsymbol{x}6$ by $\boldsymbol{w}5$ based on the extensive consideration of both diversity and convergence.

Algorithm 2 *Parent Selection for Mating* (Pop, D^*)

Input: Pop (population), N (Required number of Parent Population)

Output: Pop' (Set of Parent Population)

```
1  for i = 1 to M do
2      /*M denotes the number of objectives*/
3      Calculate the fitness of each solution by (3);
4  Pop' ← Ø;
5  while the size of Pop' is less than N do
6      for i = 1 to |Pop| do
7          Randomly select p and q from Pop;
8          if fitness_p > fitness_q then
9              Pop' ← Pop' ∪ {p};
10         else
11             Pop' ← Pop' ∪ {q};
12     end
13 return Pop'
```

Fig 3.7 Algorithm 2

In Algorithm 2 we apply the binary tournament selection [86] approach to select individuals from the current population to fill up the parent population. Here, we select two individuals from the Pop; p and q are the randomly chosen individuals in this case. Then, we calculate the fitness values

(line 3). Next, we select an individual with a better fitness value to copy the individual to the Pop' (lines 8-11). When the Pop' is filled with N solutions, two parent solutions are randomly selected from Pop' for generating an offspring, and then these selected parent solutions are removed from the Pop' until the Pop' is empty. Note here that, we employ the SBX [87] and the polynomial mutation [88] operators for the corresponding crossover and mutation operations in the proposed algorithm. Researches and many practitioners suggest that two solutions selected in a large search space is not necessary to generate a promising offspring [80], [89]. In general, research suggests two ways to solve this problem of generating a promising offspring. One method is the mating restriction to which limits the chances of offspring generated by the neighbor solutions [90]. The alternative approach is to use SBX with a large distribution index [81]. In the proposed algorithm, we employ the latter one due to its simplicity. Later, we initialize OS (offspring solution set) and store all the generated offspring solutions obtained by both SBX [87] and the polynomial mutation [88] operators when applied to the parent solution. After that, we obtain a total of N offspring solutions in OS at the end of the mating and reproduction section.

3.2.3 General Framework of APMOEA

Algorithm 3 presents the general framework of the proposed algorithm, APMOEA, for which the number of objectives and the population size act as inputs. Algorithm 3 typically consists of four steps: initialization, mating and reproduction, the first selection phase (non-dominated sorting), and the second selection phase (*p*-metric). In Step 3-2, we employ simulated binary crossover (SBX) method [84] and the polynomial mutation (PM) method [85] to reproduce *N* offspring solutions. Step 4-2, presents the normalization method used for any solution:

$$\boldsymbol{F}(\boldsymbol{x})_i = \frac{(\boldsymbol{F}(\boldsymbol{x})_i - \boldsymbol{z}^*)_i}{(\boldsymbol{F}_i^* - \boldsymbol{z}^*)_i}, i = 1,2,\cdots,M \quad (4)$$

Algorithm 3 $APMOEA(M,\ N)$

Input: objective number ; population size, N.

Step 1: Initialization

1) Generate population Pop of N solutions, randomly sampled from the decision space.
2) Generate a global ideal point $\boldsymbol{z}^* = (z_1^* \cdots z_M^*)$, where z_i^* is the best value for the ith objective, so far found in each solution of Pop.
3) Generate a set of N uniformly distributed direction vectors: D, by the Das and Dennis's method.

Step 2: Mating and Reproduction

1) Randomly select N solutions from Pop as parents.
2) Generate a set of N offspring: OS, by the simulated binary crossover and the polynomial mutation.

Step 3: First Selection Based on Non-dominated Sorting

1) Update $\boldsymbol{z}^*$ with each solution in OS.
2) Generate a union population: $Pop = Pop \cup OS$.
3) Use the method introduced in Section 3.1 to select at least N solutions reserved in Pop.

Step 4: Second Selection Based on P-Metric

1) If $Size(Pop) > N$, go on. Otherwise, go to Step 4-4.
2) Normalize the objective values of all the solutions in Pop, by (4) introduced in Section 3.2.3.
3) Use Algorithm 1 introduced in Section 3.2.2 to select the final N solutions reserved in Pop.
4) If the termination criterion is fulfilled, then stop Algorithm 2. Otherwise, return to Step 2-1.

Output: population, Pop.

Fig 3.8 Algorithm 3

Here, $\boldsymbol{F}(\boldsymbol{x})$ is the objective vector of $\boldsymbol{x}$, $\boldsymbol{z}^*$ represents the objective vector of the global ideal point, $\boldsymbol{F}^*$ indicates the objective vector of the current nadir point, and M is the number of objectives. For $\boldsymbol{F}^*$, $\boldsymbol{F}_i^*$ is the largest value of all the objective vectors in the ith objective direction.

CHAPTER IV

EXPERIMENTAL RESULTS AND DISCUSSIONS

In this chapter, we first introduce the experimental setups, including competing algorithms, test functions, performance indicators, and parameter settings. Then we analyze all the experiments according to the statistical results and the population distributions.

4.1 Experimental Setups

There are four other algorithms used for comparison in this paper, i.e., NSGA-III [5], IBEA [26], MOMBI-II [27] and RVEA [93]. NSGA-III utilizes decomposition technique; NSGA-III is efficient on tackling both MOPs and MaOPs. In order to solve scaled problems, NSGA-III normalizes the objective values of solutions. Whereas, RVEA adapts the direction vectors to achieve a similar effect. Both IBEA and MOMBI-II belongs to the family of indicator-based EAs, which utilizes additive E and $R2$ indicators, respectively. Whereas, for IBEA the fitness scaling factor κ is set to 0.05. For MOMBI-II, the threshold of variance α, the tolerance threshold E and the *record* size of nadir vectors are set to 0.5, 0.001, and 5, respectively. For NSGA-III and APMOEA, however, there is no additional parameters assigned.

The test problems utilized in the experiments include DTLZ1- DTLZ5 [88], and MaF1-MaF12 [89], in terms of three, five and ten objectives. The DTLZ test suite is one of the most widely used test suite to quantify the performance of an algorithm on both MOPs and MaOPs. The MaF test suite, specifically constructed for the CEC 2017 com-petition on evolutionary many-objective

optimization, which challenges the ability of an algorithm to tackle irregular, degenerate, disconnected, badly scaled, mixed, and many other complex PFs. MaF12-MaF15 are not included in the experiments because they are utilized for large-scale decision space tests. We also exclude DTLZ6 and DTLZ7, because MaF7 derives from DTLZ7, while DTLZ6 is quite similar to DTLZ5. Based on [86], [88], [89], Table 4.1 provides the characteristics of each test problem considered. Furthermore, Table 4.2 shows the number of decision variables for each problem, where M is the number of objectives.

TABLE 4.1

MAIN PROPERTIES OF 17 TEST PROBLEMS

Problem	**Features (Notes)**
DTLZ1	Linear, Multi-modal, Pareto many-to-one
DTLZ2	Concave, Uni-modal, Pareto many-to-one
DTLZ3	Concave, Multi-modal, Pareto many-to-one
DTLZ4	Concave, Uni-modal, Pareto many-to-one
DTLZ5	Concave, Degenerate
MaF1	Linear (Modified inverted DTLZ1)
MaF2	Concave (DTLZ2BZ)
MaF3	Convex, Multi-modal (Convex DTLZ3)
MaF4	Concave, Multi-modal (Inverted badly-scaled
MaF5	Convex, Biased (Convex badly-scaled DTLZ4)
MaF6	Concave, Degenerate (DTLZ5(I,M))
MaF7	Mixed, Disconnected, Multi-modal (DTLZ7)
MaF8	Linear, Degenerate (Multi-point distance)
MaF9	Linear, Degenerate (Multi-line distance)
MaF10	Mixed, Biased (WFG1)
MaF11	Convex, Disconnected, Nonseparable (WFG2
MaF12	Concave, Nonseparable, Biased, Deceptive

TABLE 4.2

NUMBER OF DECISION VARIABLES

Test Problem	**Decision**
DTLZ1	$M + 4$
DTLZ2-DTLZ4, MaF1-MaF6, MaF10-MaF12	$M + 9$
DTLZ5	$M + 10$
MaF7	$M + 19$
MaF8, MaF9	2

There are three performance indicators used in this paper, i.e., HV [31], IGD [90] and *p*-metric based visualization [91]. All of them are able to measure both convergence and diversity performances, to some degree. For HV, we normalize all the objective values by the global ideal points and the nadir points of the PF before actually measuring them. Thus, we are actually utilizing the normalized HV (NHV). The reference point is set to the M -dimensional vector, where M is the number of objectives. For IGD, using Das and Dennis's method we sample around 10,000 uniformly distributed points on the PF.

Generally, we implemented all experiments in MATLAB, based on the open-source platform PlatEMO [92]. We run each algorithm for 50 times on each problem, while the population sizes are set to 105, 210, and 275 for three-objective, five-objective, and ten-objective problems, respectively. We activate normalization method in APMOEA for only scaled problems. For reproduction, both SBX and PM operators are adopted, while the probabilities of crossover and mutation are set to 1.0 and $1/D$ (with D denoting the number of decision variables), respectively. The distribution indexes of both SBX and PM are set to 20. Moreover, the stopping criterion for each problem, based on the number of function evaluations (NFE), is shown in Table 4.3, in three-, five- and ten-objective cases.

TABLE 4.3

STOPPING CRITERIA FOR EACH PROBLEM

Problem	**NFE(3)**	**NFE(5)**	**NFE(10)**
DTLZ1	20,000	20,000	20,000
DTLZ2	20,000	20,000	20,000
DTLZ3	20,000	20,000	20,000
DTLZ4	20,000	20,000	20,000
DTLZ5	20,000	20,000	20,000
MaF1	20,000	60,000	100,000
MaF2	20,000	40,000	70,000
MaF3	60,000	80,000	120,000
MaF4	60,000	80,000	100,000
MaF5	10,000	20,000	20,000
MaF6	20,000	40,000	40,000
MaF7	30,000	40,000	70,000
MaF8	60,000	80,000	80,000
MaF9	60,000	60,000	60,000
MaF10	160,000	200,000	260,000
MaF11	80,000	100,000	200,000
MaF12	40,000	40,000	100,000

4.2 Performance Analysis

We compare the statistical performances of APMOEA and four chosen algorithms in Tables 4.4 and 4.5 in terms of IGD, HV and visualized by p-metric visualization framework for test problems DTLZ1-DTLZ7 for both 5-D and 10-D. Moreover, we also detail the number of performance rankings for APMOEA against all the other algorithms in Table 4.6, where rank-1st means the best and rank-5th means the worst. According to Tables 4.4, 4.5, 4.6, the performance of APMOEA is the best on most of the test problems and is very promising on all the problems, in terms of three indicators, under extensive considerations of three, five, and ten objectives. Nevertheless, on some problems, such as MaF10 and MaF11 in five- and ten-objective cases, the HV performances of APMOEA are not as good as we hoped for, although the corresponding IGD results are pretty decent. The reason behind the shortcomings only in the HV metric is that it usually gives more

preference for the knees and borders of the approximate PF, whereas APMOEA assigns equal preference to the search space around each direction vector.

For a clear observation, the distribution of non-dominated solutions in three-objective cases, and the parallel coordinates of the objective values in ten-objective cases, for each algorithm with the median IGD values tested on some problems, are shown in Figs 4.1 and 4.2 respectively. Fig 4.1 involves three-objective DTLZ1, MaF1, MaF6 and MaF7, while Fig 4.2 involves ten-objective MaF1, MaF6, MaF8 and MaF13. In Fig 4.1, APMOEA maintains its population distributions uniformly, not just with problem DTLZ1 that has a regular PF shape, but also on the other three problems, which have irregular, degenerate and disconnected PF shapes. In Fig 4.2, APMOEA performs the best on all the four problems, including MaF8, which is a multi-point distance minimization problem, in terms of both convergence and diversity. In general, APMOEA shows great versatility for tackling both MOPs and MaOPs with various PF shapes.

TABLE 4.4
IGD VALUES OBTAINED BY APMOEA, NSGA-III, IBEA, MOMBI-II and RVEA ON DTLZ1–DTLZ5, MaF1-MaF12 WITH 3, 5 and 10 OBJECTIVES. WE HIGHLIGHT THE BEST RESULT IN EACH ROW.

Problem	M	APMOEA	NSGA-III	IBEA	MOMBI-II	RVEA
DTLZ1	3	2.1198e-2 (1.88e-3)	†2.3600e-2 (1.01e-2)	1.7104e-1 (2.13e-2)	3.4124e-2 (6.19e-2)	3.8069e-2 (2.66e-2)
	5	1.2019e-1 (7.56e-2)	1.9110e-1 (1.31e-1)	2.0979e-1 (2.80e-2)	1.3377e-1 (1.08e-1)	2.3575e-1 (1.83e-1)
	10	2.2313e-1 (1.26e-1)	1.5523e+0 (4.80e-1)	†2.2375e-1 (4.25e-2)	3.2365e-1 (1.53e-1)	2.4292e-1 (1.38e-1)
DTLZ2	3	5.4975e-2 (1.32e-4)	5.4980e-2 (2.45e-4)	7.2648e-2 (2.22e-3)	1.0365e+02(1.13e+00)	6.5676e-2 (1.74e-3)
	5	1.6636e-1 (4.15e-4)	1.6832e-1 (7.61e-4)	1.9229e-1 (2.12e-3)	1.8113e-1 (3.12e-3)	1.6787e-1 (5.50e-4)
	10	4.2647e-1 (2.91e-3)	4.9105e-1 (7.23e-2)	†4.2787e-1 (2.73e-3)	4.7574e-1 (6.95e-3)	4.5337e-1 (5.90e-3)
DTLZ3	3	5.7031e+0 (3.04e+0)	1.0891e+1 (4.69e+0)	6.7873e+0 (4.31e+0)	8.3224e+0 (3.81e+0)	8.4824e+0 (4.12e+0)
	5	5.8905e+0 (3.28e+0)	2.3404e+1 (8.49e+0)	†1.1319e+1 (3.46e+0)	†7.5490e+0 (4.17e+0)	1.7114e+1 (6.49e+0)
	10	†1.5495e+1 (8.53e+0)	5.9115e+1 (2.31e+1)	8.5871e+0 (4.36e+0)	†1.0140e+1 (6.51e+0)	9.8658e+1 (2.87e+1)
DTLZ4	3	7.9805e-2 (3.28e-3)	1.7934e-1 (2.67e-1)	1.8462e-1 (2.19e-1)	1.7249e-1 (2.07e-1)	1.4554e-1 (1.81e-1)
	5	2.3412e-1 (5.12e-2)	2.9580e-1 (1.11e-1)	2.9690e-1 (9.73e-2)	3.5860e-1 (1.26e-1)	2.5822e-1 (8.54e-2)
	10	†5.2506e-1 (3.71e-3)	6.3147e-1 (6.13e-2)	5.3798e-1 (3.08e-2)	7.0670e-1 (8.56e-2)	5.0839e-1 (2.39e-2)
DTLZ5	3	5.7677e-3 (1.67e-4)	†1.2643e-2 (1.70e-3)	1.6652e-2 (1.32e-3)	2.4954e-2 (4.03e-4)	2.0776e-2 (1.35e-3)
	5	7.1020e-2 (8.26e-3)	1.7362e-1 (7.36e-2)	†1.1034e-1 (2.14e-2)	1.2239e-1 (2.38e-2)	1.3359e-1 (2.34e-2)
	10	1.2418e-1 (2.76e-2)	3.5017e-1 (8.68e-2)	†2.2508e-1 (5.25e-2)	6.2985e-1 (9.40e-2)	3.5232e-1 (6.61e-2)
MaF1	3	4.1054e-2 (1.28e-3)	6.1540e-2 (1.83e-3)	4.5408e-2 (6.78e-4)	4.0919e+01(3.34e-01)	†4.4128e-2 (5.32e-4)
	5	1.4087e-1 (1.46e-3)	2.3635e-1 (2.50e-2)	1.5163e-1 (4.17e-3)	1.7178e+01(7.52e-01)	1.5525e-1 (2.50e-3)
	10	2.6759e-1 (1.31e-2)	3.2167e-1 (1.07e-2)	3.2662e-1 (1.17e-2)	2.8553e-1 (5.12e-3)	3.2167e-1 (1.07e-2)
MaF2	3	3.1778e-2 (5.74e-4)	3.6746e-2 (1.00e-3)	3.2742e-2 (5.36e-4)	3.9384e-2 (6.97e-4)	3.3035e-2 (9.35e-4)
	5	1.1576e-1 (2.34e-3)	1.4468e-1 (5.43e-3)	1.2133e-1 (1.81e-3)	1.7003e-1 (3.02e-3)	1.1865e-1 (1.85e-3)
	10	†2.4865e-1 (1.41e-2)	3.0727e-1 (7.13e-2)	2.2123e-1 (1.44e-2)	7.6777e-1 (1.28e-2)	5.9463e-1 (2.43e-2)
MaF3	3	6.4752e+1 (6.28e+1)	2.4437e+2 (2.60e+2)	3.4753e+3 (5.16e+3)	1.1557e+2 (1.04e+2)	1.1887e+2 (9.66e+1)
	5	8.3387e+1 (9.37e+1)	8.7128e+2 (9.93e+2)	5.4951e+3 (6.91e+3)	7.5476e+2 (5.77e+2)	4.5934e+3 (9.54e+3)
	10	1.8352e+2 (2.78e+2)	4.1352e+6 (1.73e+7)	9.0205e+3 (1.46e+4)	†7.0651e+2 (8.96e+2)	1.0799e+6 (1.06e+6)
MaF4	3	1.5751e+1 (8.44e+0)	3.4171e+1 (1.06e+1)	†2.0514e+1 (9.95e+0)	2.4542e+1 (1.01e+1)	3.0433e+1 (1.52e+1)
	5	†1.0414e+2 (5.71e+1)	1.7461e+2 (7.20e+1)	9.6364e+1 (3.96e+1)	1.6339e+2 (8.57e+1)	1.5145e+2 (6.54e+1)
	10	2.7609e+3 (1.47e+3)	6.6189e+3 (3.83e+3)	3.4308e+3 (1.59e+3)	3.7591e+3 (2.83e+3)	4.2162e+3 (2.35e+3)
MaF5	3	3.5893e-1 (1.83e-2)	†5.9919e-1 (9.42e-1)	7.9586e-1 (7.81e-1)	†5.1156e-1 (6.04e-1)	6.6788e-1 (6.97e-1)
	5	2.4983e+0 (3.89e-1)	†2.6524e+0 (8.71e-1)	†2.8019e+0 (9.12e-1)	3.4824e+0 (1.29e+0)	2.8837e+0 (1.25e+0)
	10	1.5292e+2 (1.14e+1)	1.2006e+2 (3.15e+1)	9.8900e+1 (1.93e+1)	1.4429e+2 (2.00e+1)	7.3746e+1 (1.76e+1)
MaF6	3	7.0897e-3 (3.43e-3)	†1.3409e-2 (3.27e-3)	6.6833e-2 (1.60e-2)	2.4419e-2 (1.55e-3)	1.6403e-2 (1.27e-3)
	5	6.7567e-3 (9.83e-4)	†2.5173e-2 (1.20e-2)	6.1118e-2 (1.13e-2)	1.3542e-1 (2.58e-2)	2.6758e-2 (2.38e-3)
	10	5.8515e-1 (7.47e-1)	†2.6774e+0 (2.99e+0)	†1.4543e+0 (1.62e+0)	†7.0169e-1 (8.66e-2)	7.2294e+0 (3.85e+0
MaF7	3	9.0549e-2 (5.83e-2)	1.1379e-1 (6.10e-2)	1.0312e-1 (9.56e-2)	1.4743e-1 (1.00e-1)	1.8850e-1 (1.40e-1)
	5	3.2289e-1 (4.47e-2)	6.2580e-1 (2.10e-1)	†4.1632e-1 (1.77e-1)	7.9075e-1 (3.29e-1)	3.9373e-1 (2.63e-2)
	10	†2.2130e+0 (4.15e-1)	5.1786e+0 (1.23e+0)	1.8624e+0 (6.46e-1)	6.5077e+0 (8.55e-1)	4.9220e+0 (9.64e-1)
MaF8	3	3.7262e-1 (2.68e-1)	†6.1977e-1 (3.44e-1)	7.6816e-1 (2.66e-1)	5.5607e-1 (4.01e-1)	5.5061e-1 (5.00e-1)
	5	5.0554e-1 (2.70e-1)	7.1838e-1 (3.85e-1)	8.4877e-1 (2.62e-1)	5.0642e-1 (2.64e-1)	†5.3878e-1 (2.66e-1)
	10	1.0106e+0 (5.55e-1)	8.6578e-1 (3.79e-1)	1.0988e+0 (3.28e-1)	2.2747e+0 (9.85e-1)	7.4474e-1 (6.32e-1)
MaF9	3	2.6675e-1 (1.73e-1)	†2.7648e-1 (1.91e-1)	5.5378e-1 (2.25e-1)	3.1394e-1 (1.77e-1)	4.7392e-1 (2.00e-1)
	5	†9.2493e-1 (6.43e-1)	1.2209e+0 (7.92e-1)	1.1568e+0 (6.32e-1)	5.4160e-1 (3.40e-1)	1.0583e+0 (4.70e-1)
	10	1.5577e+0 (1.87e+0)	4.7482e+0 (6.77e+0)	†2.5509e+0 (1.77e+0)	5.4219e+0 (3.29e+0)	2.0726e+0 (2.04e+0)
MaF10	3	6.6827e-1 (7.80e-2)	1.1150e+0 (1.15e-1)	1.0493e+0 (9.05e-2)	†8.0024e-1 (8.86e-2)	9.1609e-1 (8.74e-2)
	5	1.7992e+0 (8.67e-2)	1.7851e+0 (1.03e-1)	1.2107e+0 (2.28e-1)	†1.4199e+0 (2.81e-1)	1.4591e+0 (1.30e-1)
	10	2.1383e+0 (3.12e-1)	2.7969e+0 (2.87e-1)	2.7921e+0 (1.11e-1)	3.3828e+0 (7.07e-1)	2.5473e+0 (2.14e-1)
MaF11	3	1.9497e-1 (2.34e-2)	1.9551e-1 (1.25e-2)	†2.7726e-1 (4.83e-2)	3.4319e-1 (1.64e-2)	2.5308e-1 (3.56e-2)
	5	†8.9418e-1 (2.61e-1)	8.0953e-1 (1.28e-1)	1.7517e+0 (4.76e-1)	1.0712e+0 (8.96e-2)	1.1603e+0 (3.18e-1)
	10	3.3980e+0 (4.76e-1)	5.8326e+0 (1.34e+0)	8.6023e+0 (2.02e+0)	5.9623e+0 (1.15e+0)	†3.7527e+0 (8.21e-1)
MaF12	3	2.3536e-1 (9.27e-3)	†2.5166e-1 (3.45e-2)	2.9363e-1 (8.87e-3)	2.6066e-1 (2.46e-2)	2.4287e-1 (5.09e-3)
	5	1.1577e+0 (1.14e-2)	1.2339e+0 (2.19e-2)	1.2206e+0 (1.12e-2)	1.8058e+0 (2.99e-2)	1.2210e+0 (1.31e-2)
	10	4.8059e+0 (3.85e-2)	†5.5871e+0 (1.53e-1)	5.6924e+0 (1.19e-1)	1.3821e+1 (3.37e+0)	5.9548e+0 (4.06e-1)

TABLE 4.5
HV VALUES OBTAINED BY APMOEA, NSGA-III, IBEA, MOMBI-II and GREA ON DTLZ1–DTLZ5, MaF1-MaF12 WITH 3, 5 and 10 OBJECTIVES. WE HIGHLIGHT THE BEST RESULT IN EACH ROW.

Problem	M	APMOEA	NSGA-III	IBEA	MOMBI-II	RVEA
DTLZ1	3	1.3885e-1 (1.06e-3)	†1.3782e-1 (4.10e-3)	7.7961e-2 (8.81e-3)	1.3404e-1 (2.46e-2)	1.3262e-1 (9.74e-3)
	5	†3.5367e-2 (4.50e-3)	3.4063e-2 (1.63e-2)	4.0259e-2 (1.33e-2)	4.3999e-2 (9.43e-3)	3.0784e-2 (1.91e-2)
	10	2.3159e-3 (3.80e-4)	3.7239e-8 (1.67e-7)	†1.9038e-3 (6.98e-4)	1.7745e-3 (7.73e-4)	1.7986e-3 (7.61e-4)
DTLZ2	3	7.4233e-1 (1.17e-3)	†7.3927e-1 (1.07e-3)	7.1794e-1 (3.77e-3)	5.6200e-01(2.25e-04)	7.4020e-1 (9.13e-4)
	5	1.3019e+0 (1.41e-3)	1.2776e+0 (4.48e-3)	1.2829e+0 (3.13e-3)	1.2964e+0 (1.09e-3)	1.2898e+0 (2.63e-3)
	10	2.4556e+0 (4.24e-3)	1.9538e+0 (2.61e-1)	2.3713e+0 (1.67e-2)	9.4428e-01(1.09e-02)	2.4103e+0 (4.23e-2)
DTLZ3	3	5.5614e-1(3.93e-3)	5.4616e-1(1.04e-2)	2.4518e-1(3.18e-3)	5.5494e-1 (4.79e-3)	4.9268e-1 (1.69e-1)
	5	†7.9626e-01(6.96e-03)	7.7140e-01(2.21e-02)	3.7351e-01(4.65e-03)	8.0278e-01(3.47e-03)	7.4055e-01(1.44e-01)
	10	9.6335e-01(3.71e-03)	2.3231e-01(3.51e-01)	6.1432e-01(3.14e-03)	8.0504e-01(1.23e-01)	9.5208e-01(2.11e-02)
DTLZ4	3	7.4182e-1 (1.55e-3)	6.5952e-1 (1.77e-1)	6.6593e-1 (1.26e-1)	†6.7494e-1 (1.24e-1)	6.9215e-1 (1.14e-1)
	5	1.2261e+0 (6.17e-2)	1.1485e+0 (9.85e-2)	†1.2204e+0 (6.01e-2)	1.1339e+0 (1.17e-1)	1.2113e+0 (3.79e-2)
	10	2.4193e+0 (9.49e-3)	2.1541e+0 (1.84e-1)	2.4350e+0 (2.96e-2)	2.0960e+0 (2.40e-1)	2.4417e+0 (2.81e-2)
DTLZ5	3	1.3196e-1 (1.61e-4)	†1.2871e-1 (8.88e-4)	1.3208e-1 (2.00e-4)	1.2631e-1 (3.28e-5)	1.2542e-1 (4.71e-4)
	5	7.8459e-3 (2.51e-4)	5.1449e-3 (1.94e-3)	†7.5503e-3 (2.86e-4)	5.5069e-3 (1.13e-3)	4.7089e-3 (1.16e-3)
	10	5.7397e-8 (1.01e-9)	6.5403e-9 (1.09e-8)	3.0275e-8 (1.04e-8)	†5.3111e-8 (1.80e-9)	5.6428e-10 (1.71e-9)
MaF1	3	†2.0355e-01(1.13e-04)	2.6899e-1 (2.29e-3)	2.9078e-1 (7.69e-4)	2.9106e-1 (1.04e-3)	2.8735e-1 (1.13e-3)
	5	1.4636e-2 (2.60e-4)	5.8283e-3 (9.59e-4)	†1.4584e-2 (3.05e-4)	5.3498e-03(9.17e-05)	1.2169e-2 (4.46e-4)
	10	6.5513e-7 (5.83e-8)	†3.6843e-7 (6.85e-8)	9.2068e-8 (4.72e-7)	1.5077e-7 (4.89e-8)	5.5413e-7 (7.31e-7)
MaF2	3	†2.1084e-1 (9.81e-4)	2.0904e-1 (1.60e-3)	2.1667e-1 (4.60e-4)	2.0882e-1 (2.22e-3)	2.1776e-1 (2.63e-4)
	5	3.6524e-2 (4.17e-4)	4.0774e-2 (1.02e-3)	4.8206e-2 (6.93e-4)	†5.0728e-2 (4.66e-4)	4.2189e-2 (7.74e-4)
	10	7.6384e-3 (2.09e-4)	5.7334e-3 (3.07e-4)	6.6208e-3 (1.81e-4)	3.0998e-3 (2.10e-4)	5.9371e-3 (1.32e-4)
MaF3	3	9.5317e-01(2.69e-03)	9.5500e-01(4.39e-03)	5.6714e-01(3.69e-01)	9.5039e-01(9.64e-03)	9.2584e-01(4.11e-02)
	5	9.9498e-01(4.40e-03)	9.7383e-01(7.63e-02)	8.8143e-01(1.99e-01)	9.9895e-01(5.30e-04)	5.5982e-01(4.51e-01)
	10	9.9973e-01(3.27e-04)	0.0000e+00(0.00e+00)	9.4905e-01(4.74e-02)	9.9797e-01(2.85e-03)	9.0682e-01(2.63e-01)
MaF4	3	5.2396e-01(9.09e-03)	5.1316e-01(1.07e-02)	3.4736e-01(3.12e-02)	5.0557e-01(2.81e-03)	5.0079e-01(2.16e-02)
	5	6.2658e-03(8.85e-04)	6.6174e-02(1.14e-02)	1.1048e-01(6.39e-03)	3.5715e-02(3.54e-03)	3.2587e-02(1.08e-02)
	10	5.1398e-09(1.54e-08)	2.3385e-04(3.01e-05)	7.8946e-05(2.57e-05)	1.2252e-04(1.84e-05)	4.8893e-07(1.18e-06)
MaF5	3	4.7528e+1 (7.47e-2)	4.3513e+1 (9.31e+0)	4.1312e+1 (8.65e+0)	4.5026e+1 (6.36e+0)	4.3007e+1 (8.31e+0)
	5	3.9998e+4 (2.59e+3)	3.9569e+4 (1.49e+3)	3.9645e+4 (1.38e+3)	3.7253e+4 (4.11e+3)	3.8927e+4 (3.70e+3)
	10	8.4552e+16 (1.13e+15)	8.2667e+16 (5.89e+15)	†8.7261e+16 (1.89e+15)	7.4086e+16 (1.03e+16)	8.7774e+16 (1.54e+15)
MaF6	3	1.3070e-1 (1.67e-3)	†1.2766e-1 (1.23e-3)	1.2144e-1 (4.15e-3)	1.2563e-1 (5.77e-4)	1.2595e-1 (5.72e-4)
	5	9.0025e-3 (7.03e-5)	†8.7347e-3 (1.43e-4)	8.1821e-3 (3.07e-4)	7.7744e-3 (1.84e-4)	8.5276e-3 (5.53e-5)
	10	5.6512e-8 (6.54e-10)	†9.7330e-10 (5.33e-9)	2.1957e-8 (2.82e-8)	†3.4540e-8 (3.07e-8)	1.9753e-9 (1.08e-8)
MaF7	3	1.4519e+0 (7.91e-2)	1.4258e+0 (7.18e-2)	1.6036e+0 (7.24e-2)	†1.5748e+0 (7.86e-2)	1.5712e+0 (4.44e-2)
	5	2.1618e+0 (8.35e-2)	1.0694e+0 (2.47e-1)	1.3222e+0 (1.36e-1)	1.9464e+0 (1.00e-1)	2.0189e+0 (6.62e-2)
	10	2.1562e+0 (1.55e-1)	2.1921e-1 (2.10e-1)	1.0848e-1 (2.13e-1)	1.1158e+0 (4.77e-1)	1.5246e+0 (1.90e-1)
MaF8	3	1.1304e+0 (5.42e-1)	7.0504e-1 (5.51e-1)	4.0912e-1 (2.99e-1)	†8.3817e-1 (5.27e-1)	8.8292e-1 (5.51e-1)
	5	2.4860e+0 (9.48e-1)	1.8233e+0 (1.26e+0)	8.7211e-1 (5.91e-1)	†2.4267e+0 (1.07e+0)	2.3984e+0 (1.22e+0)
	10	†1.1601e+0 (1.69e+0)	7.6804e+0 (5.28e+0)	6.8400e+0 (5.72e+0)	1.3678e+0 (1.77e+0)	1.1383e+1 (6.99e+0)
MaF9	3	2.9260e+0 (6.25e-1)	†2.8868e+0 (6.48e-1)	1.9713e+0 (7.33e-1)	2.8285e+0 (5.55e-1)	2.2998e+0 (6.31e-1)
	5	4.9903e+0 (1.96e+0)	2.4133e+0 (2.15e+0)	2.1316e+0 (1.77e+0)	†3.6545e+0 (2.95e+0)	2.5011e+0 (1.90e+0)
	10	1.1691e+1 (8.75e+0)	3.3937e+0 (3.66e+0)	2.6622e+0 (2.82e+0)	8.9556e-1 (1.46e+0)	3.3954e+0 (2.10e+0)
MaF10	3	4.0410e+1 (2.51e+0)	2.8322e+1 (2.94e+0)	2.9734e+1 (2.40e+0)	†3.5486e+1 (2.81e+0)	3.3668e+1 (2.47e+0)
	5	2.1367e+3 (1.72e+2)	2.2803e+3 (1.73e+2)	†3.5049e+3 (4.45e+2)	3.8530e+3 (3.83e+2)	2.8211e+3 (2.98e+2)
	10	2.3813e+9 (2.11e+8)	3.4939e+9 (5.95e+8)	4.7780e+9 (1.33e+9)	†4.6815e+9 (8.27e+8)	3.0053e+9 (3.80e+8)
MaF11	3	5.9095e+1 (1.65e-1)	5.8000e+1 (3.65e-1)	5.8277e+1 (3.03e-1)	5.7045e+1 (4.87e-1)	5.8354e+1 (2.53e-1)
	5	5.8832e+3 (7.29e+1)	5.9028e+3 (7.08e+1)	†5.9332e+3 (5.12e+1)	5.9723e+3 (5.33e+1)	5.8423e+3 (5.66e+1)
	10	9.2900e+9 (7.77e+7)	†9.2008e+9 (1.93e+8)	9.0813e+9 (2.19e+8)	6.0750e+9 (9.49e+8)	9.0182e+9 (1.77e+8)
MaF12	3	3.3788e+1 (1.51e-1)	3.1424e+1 (1.94e+0)	3.2098e+1 (6.70e-1)	3.1839e+1 (1.51e+0)	3.3611e+1 (3.32e-1)
	5	4.2694e+3 (5.29e+1)	3.6534e+3 (2.21e+2)	3.6904e+3 (1.79e+2)	3.5554e+3 (1.03e+2)	4.2634e+3 (8.84e+1)
	10	7.3267e+9 (2.61e+8)	6.3618e+9 (6.89e+8)	6.9937e+9 (3.18e+8)	2.5692e+9 (1.23e+9)	6.1057e+9 (4.64e+8)

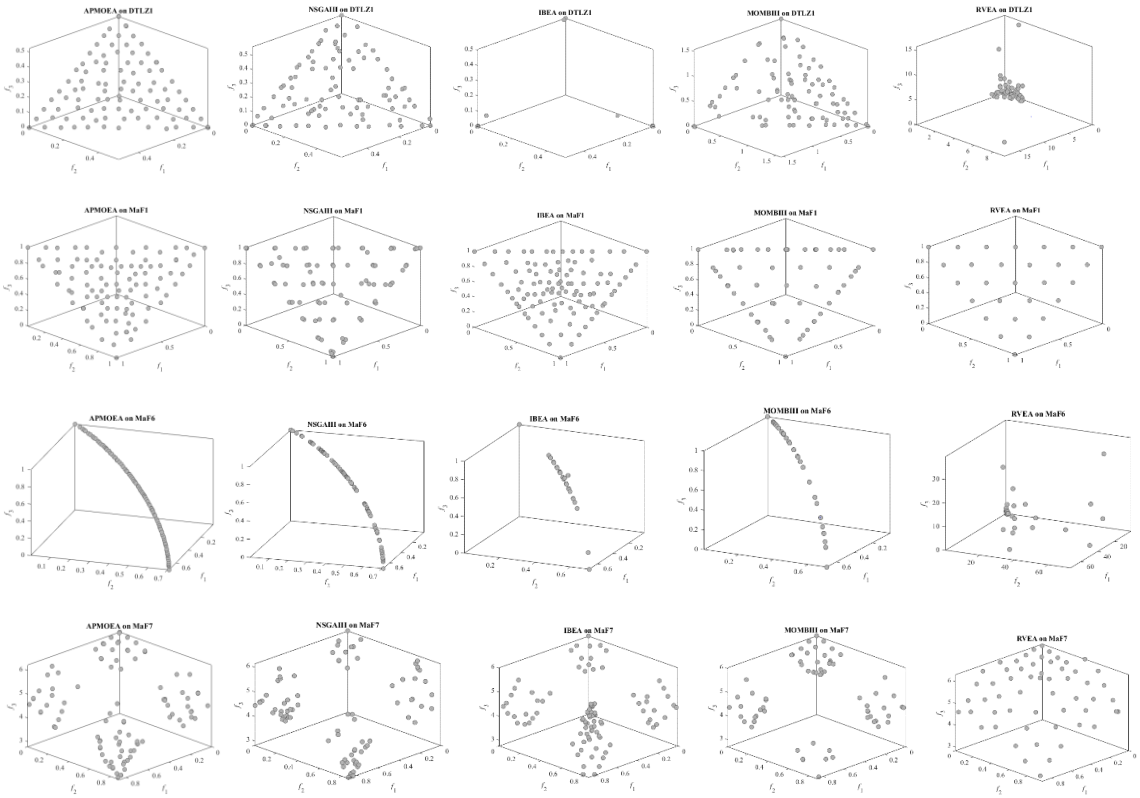
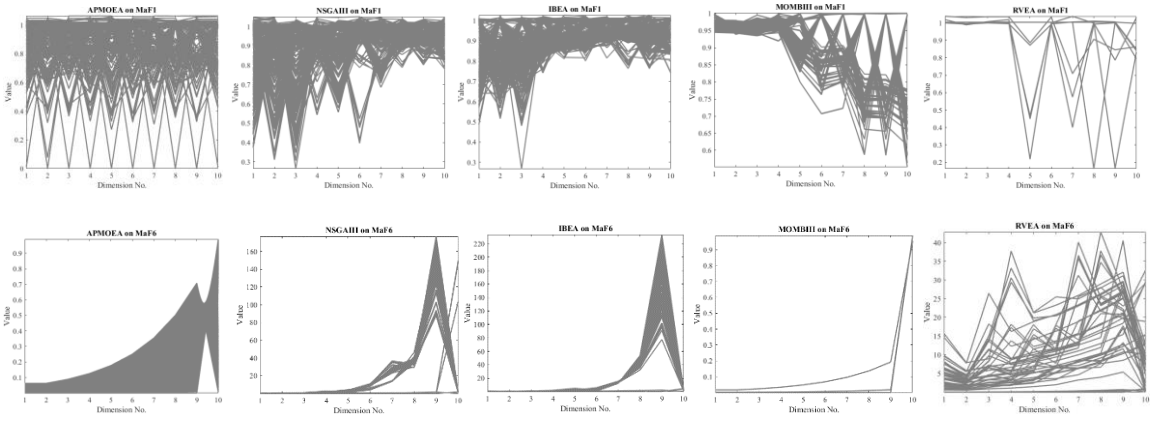


Fig 4.1 Non-dominated solution distributions for each algorithm on three-objective DTLZ1, MaF1, MaF6 and MaF7.



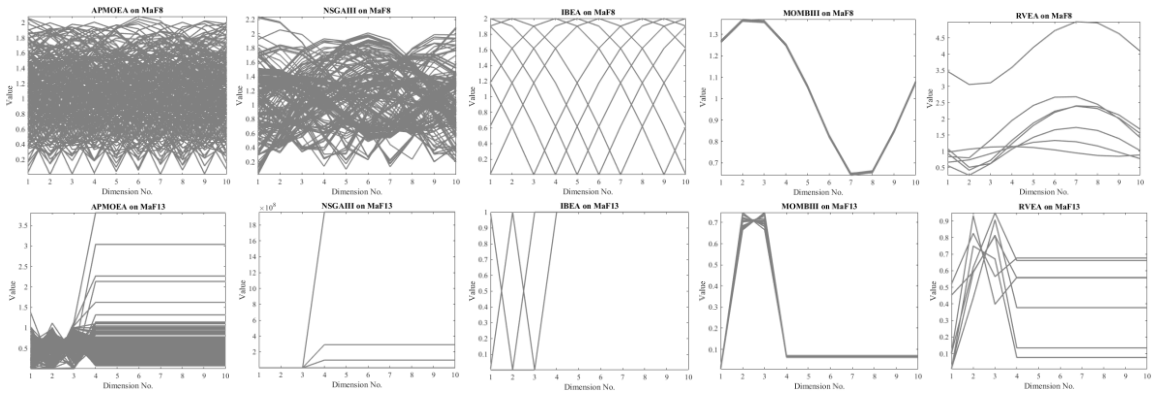


Fig 4.2 Non-dominated solution distributions for each algorithm on ten-objective MaF1, MaF6, MaF8 and MaF13.

TABLE 4.6

NUMBER OF PERFORMANCE RANKINGS FOR APMOEA

1st Rank	2nd Rank	3rd Rank	4th Rank	5th Rank
80	12	8	1	1

4.3 P-metric based Visualization

Visualization of the approximate fronts generated by all four MaOEAs on the same mapped 2-D polar plot for visual comparison. In each figure (for the 5-D or 10-D DTLZ1–DTLZ7), “red dots” represents the true Pareto front, “blue circle” is the approximate front by GrEA, “black star” represents ε-MOEA, “yellow square” refers to NSGA-III, “cyan triangle” corresponds to MOEA/D and “Magenta hexagon” represents APMOEA. In each figure, under the same DTLZ test problem, the mapped 5-D true Pareto front is no different from the mapped 10-D true Pareto front. For instance, the true Parent front of 5-D DTLZ1 in Fig 4.3 is equivalent as that of 10-D DTLZ1 in Fig.

4.4 They seem different because the scaling is different to allow the best visualization of the approximate fronts generated by all five competing MaOEAs. Fig 4.3 shows that MOEA/D generates an approximate front (cyan triangle) with all solutions locating inside one portion of the objective space covered by

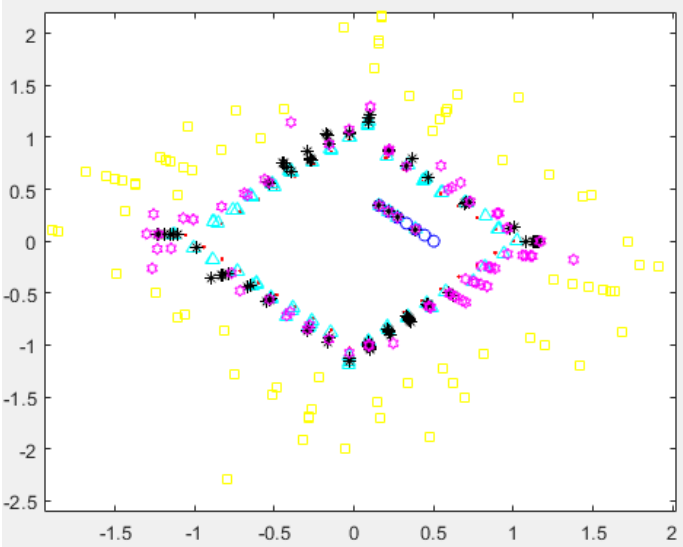


Fig 4.3 5-D DTLZ1.

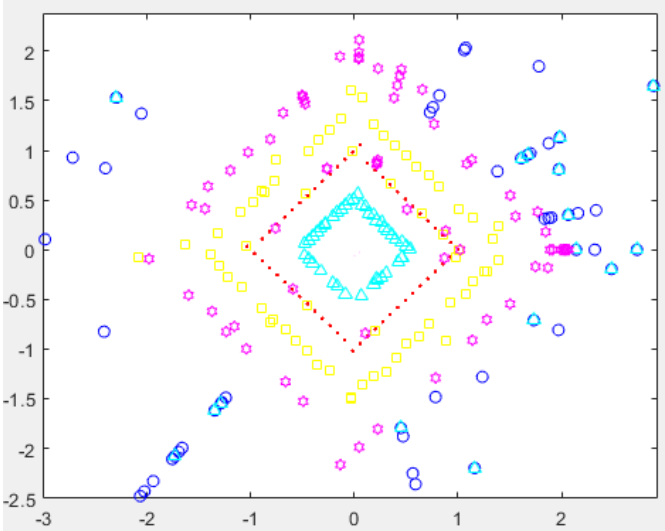


Fig 4.4 10-D DTLZ1.

approximate front of APMOEA (magenta hexagon), which properly reflects its poor diversity performance. In 10-D DTLZ1, from Fig 4.4 the approximate front obtained by ε-MOEA (black-star) is much further away from the true Pareto front. In both 5-D and 10-D DTLZ1 that present a linear Pareto front and contain a large number of local fronts [86], APMOEA provides the best performance while ε-MOEA performs the worst as the convergence power of ε-MOEA is based on Pareto-dominance modification.

In 5-D DTLZ2, ε-MOEA (black-star) has a similar diversity performance but slightly worse convergence performance than NSGA-III (yellow), while MOEA/D (cyan) shows similar convergence performance but poor diversity performance compared to GrEA, and APMOEA. However, Fig 4.5 shows that APMOEA (magenta hexagon) indeed performs better in both convergence and diversity than those three. However, the approximate fronts generated by both GrEA and APMOEA are very close as shown in Fig 4.5 ε-MOEA (black-star) has the worst performance yet again.

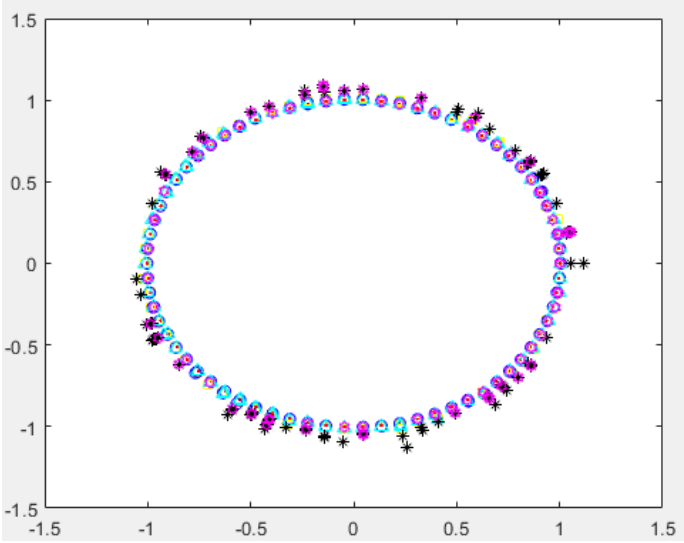


Fig 4.5 5-D DTLZ2.

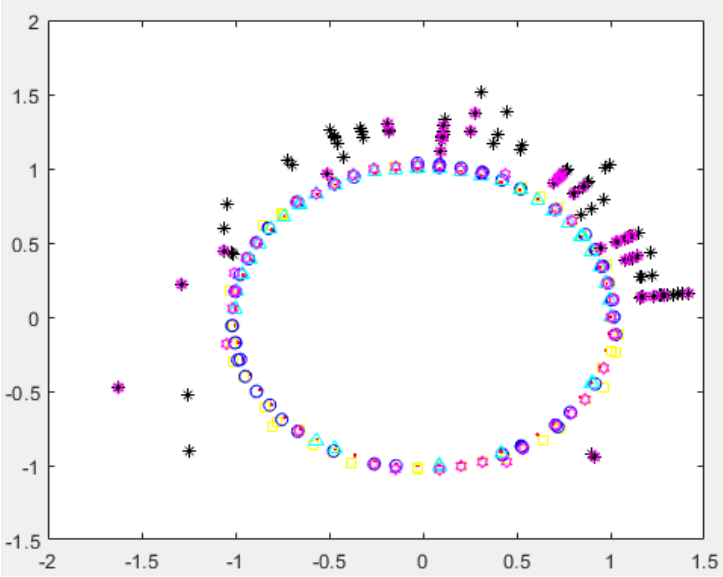


Fig 4.6 10-D DTLZ2.

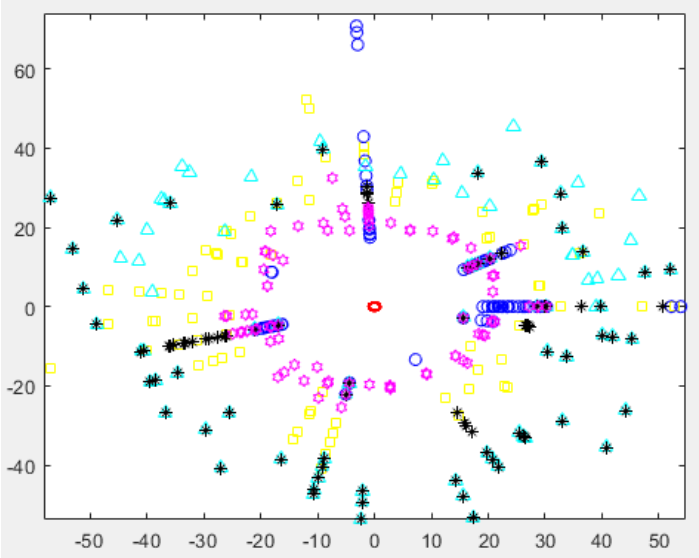


Fig 4.7 5-D DTLZ3.

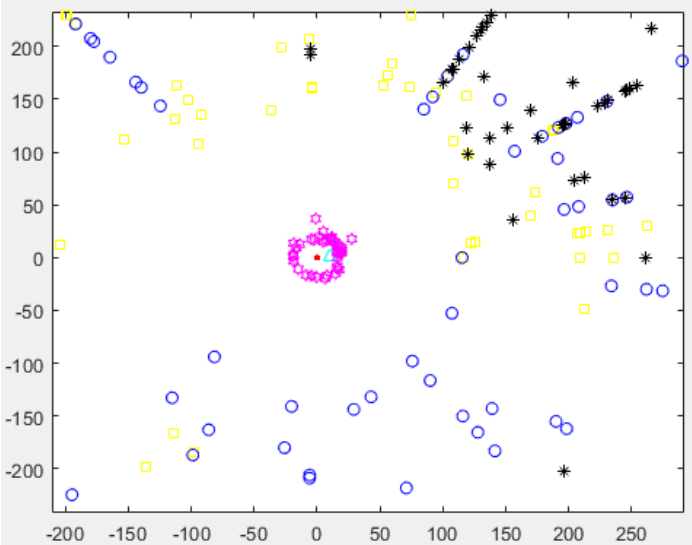


Fig 4.8 10-D DTLZ3.

DTLZ3 also introduces a large number of local Pareto fronts [86]. Again, APMOEA shows its best performance compared to all the others in both 5-D and 10-D problems, while ε-MOEA has an inferior performance. Furthermore, in Figs 4.7 and 4.8, it is easy to observe that ε-MOEA (black), and GrEA (blue) can only converge to several different local Pareto fronts. The other two offers poor convergence and diversity performance. In 5-D DTLZ4, Fig 4.9 shows that, both algorithms generate the approximate fronts with very poor diversity. Furthermore, their convergence performance is also worse than that of APMOEA. In 10-D DTLZ4, Fig 4.10 shows that three algorithms ε-MOEA, NSGA-III and MOEA/D generate the approximate fronts with very poor convergence and diversity. Fig 4.10 shows that, APMOEA performs indeed better than all the other algorithms in both convergence and diversity. [5] Suggests that DTLZ4 generates a non-uniform distribution of solutions along the Pareto front. In 10-D DTLZ5, the Pareto front is a degenerated hypersurface [86].

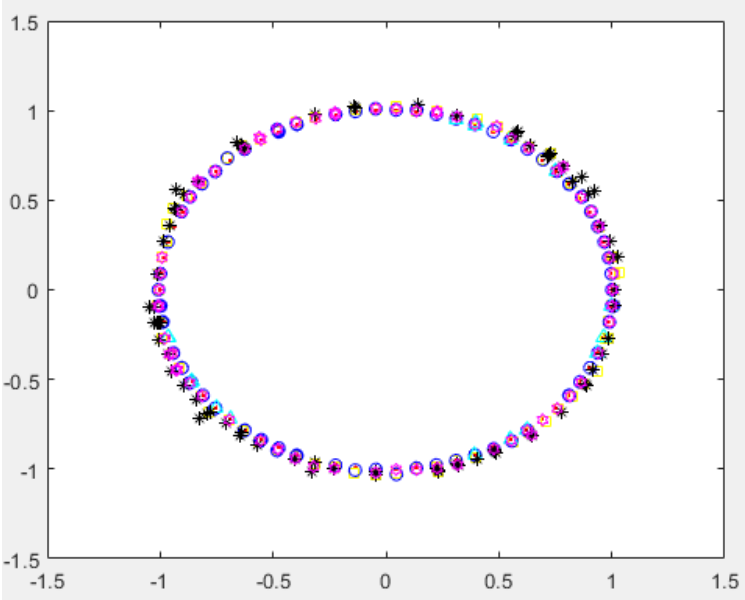


Fig 4.9 5-D DTLZ4.

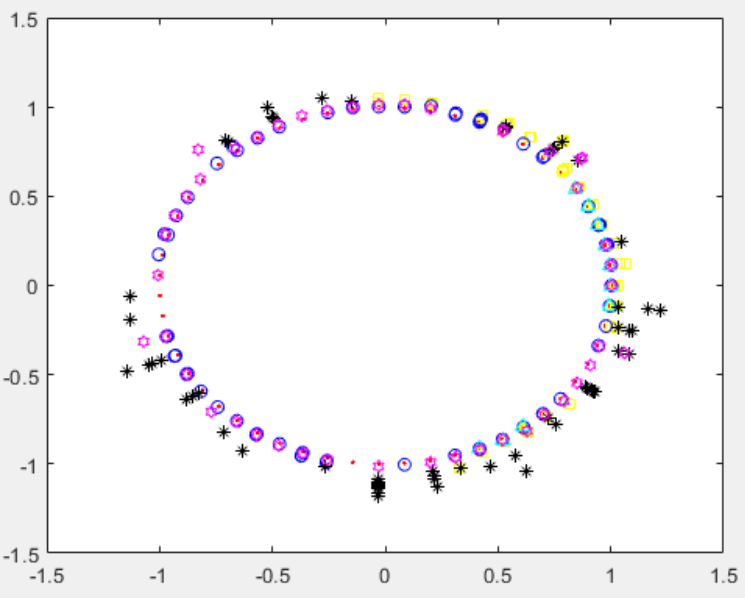


Fig 4.10 10-D DTLZ4.

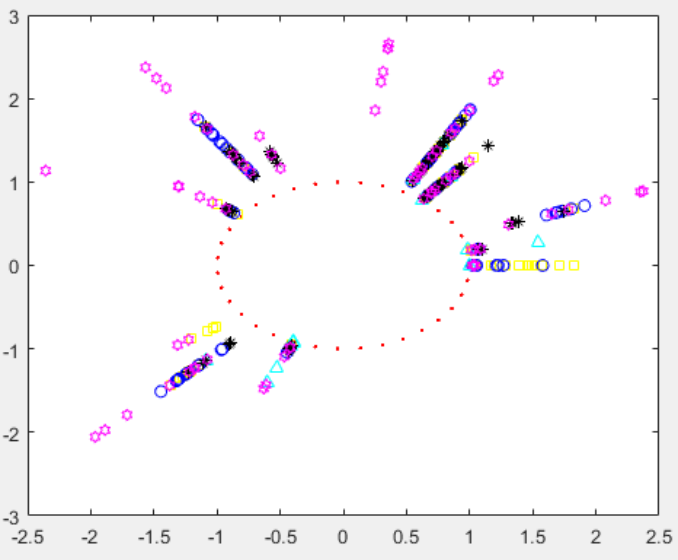


Fig 4.11 5-D DTLZ5.

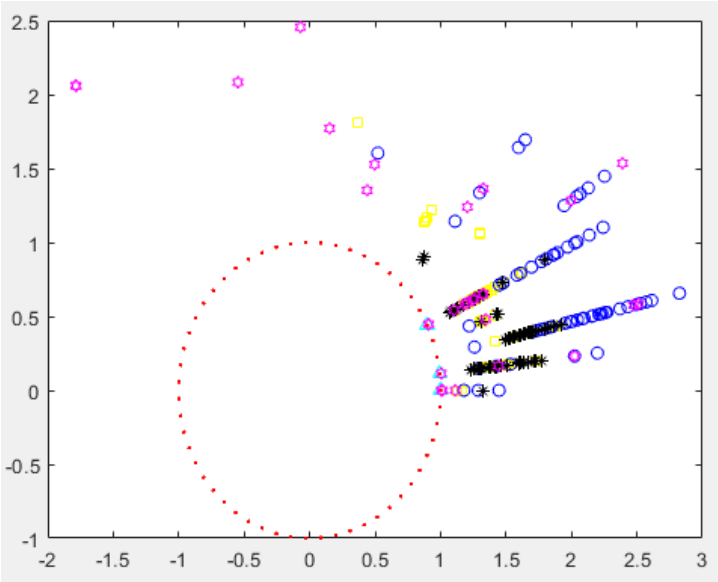


Fig 4.12 10-D DTLZ5.

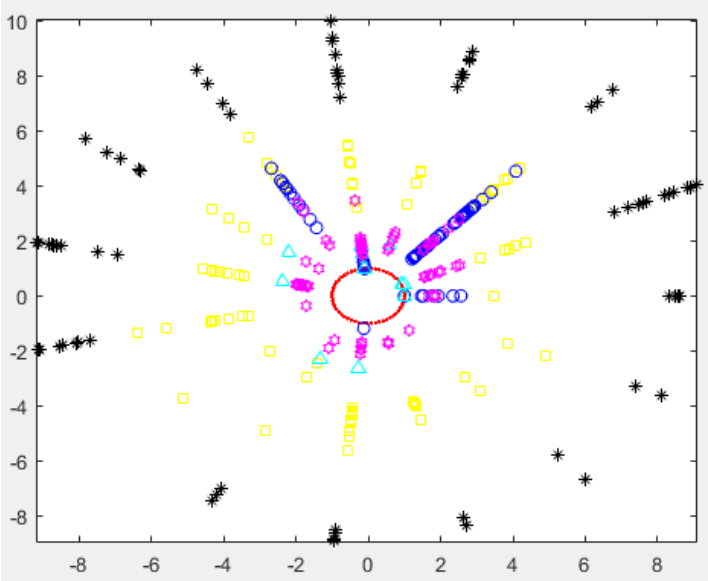


Fig 4.13 5-D DTLZ6.

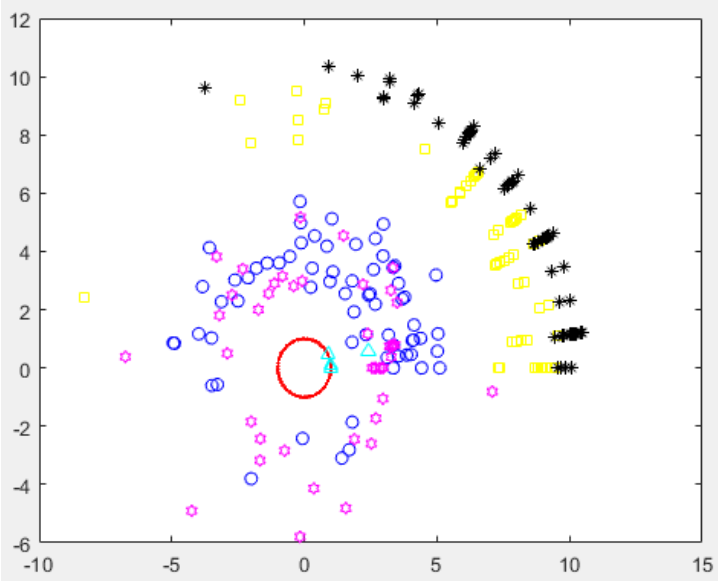


Fig 4.14 10-D DTLZ6.

Here, Figs 4.11 and 4.12 imply that not all MaOEAs chosen can attain both well convergence and diversity performance in DTLZ5. From [16], DTLZ6 has a large number of local Pareto fronts and disconnected Pareto-optimal regions. Again, the difficulty of ε-MOEA in handling lots of local

Pareto fronts results into a poor performance in DTLZ6. APMOEA has shown a good capability in dealing with lots of local fronts in DTLZ6 and it has better performance with respect to convergence and diversity compared to all other algorithms. GrEA performs closer to APMOEA but, it displays a poor performance in 10-D problem. Therefore, the disconnected Pareto-optimal regions plus high-dimensional space make it difficult for grid-based method to achieve a well converged and diversified approximate front. Meanwhile, in 4.14 it is easy to observe that a lot of approximate solutions generated by GrEA (blue), MOEA/D (cyan), and NSGA-III (yellow) converge to several local Pareto fronts.

CHAPTER V

CONCLUSION & FUTURE WORK

5.1 Conclusion

In this thesis, we focus on developing a new algorithm to handle many-objective optimization problems. We focus on developing the algorithm by two different selection schemes as literature proved that non-dominated sorting alone is not effective in tackling many-objectives optimization problems. We consider developing *p*-metric based technique as our second scheme because, it is a practical approach of maintaining convergence and diversity and a visualization tool is helpful in monitoring the performance of the proposed algorithm compared to other state-of-the-art evolutionary algorithms.

Furthermore, we also detail the types of MOEAs in literature and to our best knowledge; we list out the shortcomings of every design. The listed types of MOEAs include Pareto-dominance modification based MOEAs, Decomposition-based MOEAs, Grid-based MOEAs, and Performance Indicator-based MOEAs. The proposed APMOEA belongs to the last category. In addition, this thesis explains general framework of the above-mentioned MOEAs.

In this thesis we also delineate how the performance is assessed for the state-of-the-art MOEAs and we explain the methodologies of widely used performance assessment indicators such as hypervolume indicator, Generational distance and Inverted Generational Distance. In addition to

that we elaborate on the popular benchmark test problems used to test many MOPs and MaOPs to measure the performance of the algorithms.

In this thesis, we have proposed a new *p*-metric based EA, termed APMOEA, for dealing both MOPs and MaOPs. In APMOEA, we employ a two-phase selection, which adopts both the non-dominated sorting and *p*-metric techniques to select solutions. Moreover, we propose a modification method to adjust the direction vectors of *p*-metric dynamically. The performance improvement in both convergence and diversity is gained directly by overcoming two fundamental challenges existing in MaOPs: extremely large objective space and ineffectiveness of Pareto-dominance. On comparison with four state-of-the-art MOEAs, in terms of three performance indicators, APMOEA shows improved performances on most of the test problems, in both multi- and many-objective problems. APMOEA maintains its population distribution not only in problems with regular PF shapes but also in problems, which have irregular, degenerate and disconnected PF shapes which speaks volumes about the robustness of the APMOEA algorithm. In this thesis we utilize IGD and hypervolume metrics to compare APMOEA with NSGA-III, IBEA, MOMBI-II, and RVEA. All the algorithms are tested on 17 test problems (DTLZ1-DTLZ5, MAF1-MAF12) given with 3-, 5-, and 10- objective problems.

In a high-dimensional objective space, visualization presents an essential tool in developing MaOEAs and in solving MaOPs. He and Yen [91] proposed a unique visualization approach which utilized the tool to visualize the performance of the four other algorithms including APMOEA. The tool maps individuals from a high-dimensional objective space into a 2-dimensional polar coordinate system while preserving Pareto dominance relationship, retaining shape and location of Pareto front, and maintaining their distribution. From the resulted polar plot, we can observe the

evolution process, estimate location, range, and distribution of Pareto front, assess quality of the approximate front and tradeoff between objectives.

5.2 Future Work

In this thesis to solve MaOPs, we have developed a new MaOEA (APMOEA) to find the approximate fronts, by combining a new *p*-metric based method to existing non-dominated sorting method to obtain a comprehensive measure among different MaOEAs, and we use a new visualization tool and a *p*-metric specifically designed for high-dimensional objective spaces. Nonetheless, all the above mentioned approaches primarily focus on solving unconstrained optimization problems. In future work, we would like to extend the similar works to Constrained Many-Objective Optimization Problems (CMaOP) because most real-world problems have associated constraints. For the obtained approximate front to be accessible, all constraints must be contended. Accordingly, the concept of satisfying constraints should be employed into each step of the design of an algorithm to solve MaOPs. Firstly, while developing a new Constrained Many-Objective Optimization Evolutionary Algorithm (CMaOEA), there should be a modified fitness calculation method which considers not just convergence and diversity performance of the solution but also the degree of constraint violation. In addition, the performance score of every solution should be a combination of its convergence and diversity measures, and also constraint satisfying degree.

REFERENCES

[1] D. Brockhoff and E. Zitzler, "Objective reduction in evolutionary multiobjective optimization: theory and applications," *Evolutionary Computation*, vol. 17, pp. 135-166, 2009.

[2] K. Deb and D. Saxena, "On finding Pareto-optimal solutions through dimensionality reduction for certain large-dimensional multi-objective optimization problems," *Technical Report* KanGAL No. 2005011, 2005.

[3] H. Singh, A. Isaacs, and T. Ray, "A Pareto corner search evolutionary algorithm and dimensionality reduction in many-objective optimization problems," *IEEE Transactions on Evolutionary Computation, vol*. 15, pp. 539-556, Aug. 2011.

[4] Z. He and G. G. Yen, "Fuzzy-based pareto optimality for many-objective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol.18, no.2, pp.269-285, 2014.

[5] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part 1: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[6] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*, vol. 16. John Wiley & Sons: New York, NY, 2001.

[7] K. Miettinen, *Nonlinear Multiobjective Optimization*, Springer Science & Business Media, 2012.

[8] K. Deb, A. R. Reddy, and G. Singh, "Optimal scheduling of casting sequence using genetic algorithms", *Materials and Manufacturing Processes*, vol.18, no.3, pp.409–432, 2003.

[9] K. Deb, P. Jain, N. K. Gupta, and H. K. Maji, "Multi objective placement of electronic components using evolutionary algorithms", *IEEE Transactions on Components and Packaging Technologies*, vol. 27, no. 3, pp. 480–492, 2004.

[10] K. Deb, K. Mitra, R. Dewri, and S. Majumdar, "Towards a better understanding of the epoxy-polymerization process using multi-objective evolutionary computation," *Chemical Engineering Science*, vol. 59, no. 20, pp. 4261–4277, 2004.

[11] D. K. Saxena and K. Deb, "Trading on infeasibility by exploiting constraint's criticality through multi-objective Optimization: a system design perspective," in *Proceedings of the Congress on Evolutionary Computation,* New Jersey, United States, pp. 919–926, 2007.

[12] L. Batista, F. Campelo, F. Guimaraes, and J. Ramirez, "A comparison of dominance criteria in many-objective optimization problems," in *Proceedings of IEEE Congress on Evolutionary Computation*, Barcelona, Spain, pp. 2359-2366, 2010.

[13] K. Deb, M. Mohan, and S. Mishra, "Evaluating the ε-domination based multi-objective evolutionary algorithm for a quick computation of Pareto-optimal solutions," *Evolutionary Computation*, vol. 13, no. 4, pp. 501-525, 2005.

[14] D. Hadka and P. Reed, "Diagnostic assessment of search controls and failure modes in many objective evolutionary optimization," *Evolutionary Computation*, vol. 20, no. 3, pp. 423-452, 2012.

[15] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712731, 2007.

[16] S. Yang, M. Li, X. Liu, J. Zheng, "A grid-based evolutionary algorithm for many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 17, no. 5, pp. 721-736, 2013.

[17] F. di Pierro, S.-T. Khu, and D. A. Savi´c, "An investigation on preference order ranking scheme for multiobjective evolutionary optimization," *IEEE Transactions on Evolutionary Computation,* vol. 11, no. 1, pp. 17-45, 2007.

[18] M. Li, S. Yang, and X. Liu, "Shift-based density estimation for pareto-based algorithms in many-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 348-365, 2014.

[19] H. Liu, F. Gu, and Q. Zhang, "Decomposition of a multiobjective optimization problem into a number of simple multiobjective subproblems," *IEEE Transactions on Evolutionary Computation,* vol. 18, no. 3, pp. 450-455, 2014.

[20] E. Hughes, "Fitness assignment methods for many-objective problems," in J. Knowles, D. Corne, and K. Deb, editors, *Multi-Objective Problem Solving from Nature: From Concepts to Applications*, Springer: Berlin, Germany, pp. 307-329, 2008.

[21] M. Garza-Fabre, G.T. Pulido, and C.A. Coello Coello, "Alternative fitness assignment methods for many-objective optimization problems," in *Proceedings of Artificial Evolution Conference,* Strasbourg, France, pp. 146-157, 2009.

[22] M. Garza-Fabre, G.T. Pulido, and C.A. Coello Coello, "Ranking methods for many-objective optimization," in *Proceedings of the 8th Mexican International Conference on Artificial Intelligence,* Guanajuato, México, pp. 633-645, 2009.

[23] J. Bader and E. Zitzler, "HypE: An algorithm for fast hypervolume based many-objective optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45-76, 2011.

[24] E. Zitzler and K. Simon, "Indicator-based selection in multiobjective search," in *Proceedings of International Conference on Parallel Problem Solving from Nature*, Birmingham, UK, pp. 832-842, 2004.

[25] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: multiobjective selection based on dominated hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653-1669, 2007.

[26] E. Zitzler, L. Thiele, and K. Deb, "Comparison of multiobjective evolutionary algorithms: empirical results," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 2, pp. 173- 195, 2000.

[27] G.G. Yen and Z. He, "Performance metrics ensemble for multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol.18, no.1, pp.131-144, 2014

[28] I. Das and J.E. Dennis, "Normal-boundary intersections: a new method for generating Pareto optimal points in multicriteria optimization problems," *SIAM Journal of Optim.*, vol. 8, no. 3, pp. 631-657, 1998.

[29] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

[30] G.G. Yen and H. Lu, "Dynamic multiobjective evolutionary algorithm: adaptive cell-based rank and density estimation," *IEEE Transactions on Evolutionary Computations*, vol. 7, no. 3, pp. 253-274, 2003.

[31] E. Zitzler and L. Thiele, "An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Citeseer", *IEEE Transactions on Evolutionary Computation*, vol. 3, no 4, pp. 257–271, 1998

[32] T. Voß, N. Hansen, and C. Igel, "Improved step size adaptation for the MO-CMA-ES," in *Proceedings of the 12th annual conference on Genetic and evolutionary computation*, Portland, United States, pp. 487-494, 2010.

[33] M. Helbig and A. P. Engelbrecht, "Performance measures for dynamic multi-objective optimisation algorithms," *Information Sciences,* vol. 250,pp. 61-81, 2013.

[34] K. Van Moffaert, M. M. Drugan, and A. Nowé, "Hypervolume-based multiobjective reinforcement learning," in *Proceedings of Evolutionary Multi-Criterion Optimization,* Heidelberg, Germany, pp. 352-366, 2013.

[35] L. While, P. Hingston, L. Barone, and S. Huband, "A faster algorithm for calculating hypervolume," *Evolutionary Computation, IEEE Transactions on,* vol. 10, no. 1, pp. 29-38, 2006.

[36] C. M. Fonseca, L. Paquete, and M. López-Ibánez, "An improved dimensionsweep algorithm for the hypervolume indicator," in *Proceedings of 2006 IEEE Congress on Evolutionary Computation,* pp. 1157-1163, 2006.

[37] Q. Yang and S. Ding, "Novel algorithm to calculate hypervolume indicator of pareto approximation set," in *Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques*, pp. 235-244, Springer, 2007.

[38] K. Bringmann and T. Friedrich, "Approximating the least hypervolume contributor: NP-hard in general, but fast in practice," in *Proceedings of Evolutionary Multi-Criterion Optimization*, Heidelberg, Germany, pp. 6-20, 2009.

[39] J. Bader, K. Deb, and E. Zitzler, "Faster hypervolume based search using Monte Carlo sampling," in *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems,* Auckland, New Zealand, pp. 313-326, 2010.

[40] E. J. Hughes, "Evolutionary many-objective optimisation: many once or one many?," in *Proceedings of 2005 IEEE Congress on Evolutionary Computation, 2005. The 2005 IEEE Congress on*, vol. 1, pp. 222-227, IEEE, 2005.

[41] Y. Tang, P. Reed, T. Wagener, "How effective and efficient are multiobjective evolutionary algorithms at hydrologic model calibration?" *Hydrology and Earth System Sciences Discussions*, vol. 10, no. 2, pp. 289-307,2006.

[42] A. J. Nebro, J. Durillo, J. Garcia-Nieto, C. A. Coello Coello, F. Luna, and E. Alba, "Smpso: A new PSO-based metaheuristic for multi-objective optimization," in *Proceedings of 2009 IEEE Symposium on Computational Intelligence in Miulti-Criteria Decision-Making*, pp. 66-73, IEEE, 2009.

[43] J. J. Durillo, J. García-Nieto, A. J. Nebro, C. A. Coello Coello, F. Luna, and E. Alba, "Multi-objective particle swarm optimizers: An experimental comparison," in *Proceedings of Conference on Evolutionary Multi-Criterion Optimization*, Nantes, France, pp. 495-509, 2009.

[44] R. C. Purshouse, C. Jalba, and P. J. Fleming, "Preference-driven coevolutionary algorithms show promise for many-objective optimisation," in *Proceedings of 2011 Conference on Evolutionary Multi-Criterion Optimization*, pp. 136-150, Springer, 2011.

[45] D. A. Van Veldhuizen and G. B. Lamont, "Evolutionary computation and convergence to a pareto front," in *Late breaking papers at the genetic programming 1998 conference*, pp. 221-228, Citeseer, 1998.

[46] D. A. Van Veldhuizen, "Multiobjective evolutionary algorithms: classifications, analyses, and new innovations," tech. rep., DTIC Document, 1999.

[47] G. T. Pulido and C. A. Coello Coello, "Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer," in *Proceedings of 2004 Genetic and Evolutionary Computation Conference*, pp. 225-237, 2004.

[48] J. Knowles and D. Corne, "On metrics for comparing nondominated sets," in *Proceedings of 2002 IEEE Congress on Evolutionary Computation*, New Jersey, United States, pp. 711-716, 2002.

[49] A. J. Nebro, J. J. Durillo, F. Luna, B. Dorronsoro, and E. Alba, "Mocell: A cellular genetic algorithm for multiobjective optimization," *International Journal of Intelligent Systems*, vol. 24, no. 7, pp. 726-746, 2009.

[50] C. A. Coello Coello, G. T. Pulido, and M. S. Lechuga, "Handling multiple objectives with particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 256-279, 2004.

[51] C. A. Coello Coello and M. R. Sierra, "A coevolutionary multi-objective evolutionary algorithm," *Proceedings of 2003 IEEE Congress on Evolutionary Computation,* pp. 482-489, 2003.

[52] C. A. Coello Coello and N. C. Cort_es, "Solving multiobjective optimization problems using an artificial immune system," *Genetic Programming and Evolvable Machines*, vol. 6, no. 2, pp. 163-190, 2005.

[53] Q. Zhang, A. Zhou, and Y. Jin, "Rm-meda: A regularity model-based multiobjective estimation of distribution algorithm," *IEEE Transactions* on *Evolutionary Computation,* vol. 12, no. 1, pp. 41-63, 2008.

[54] Q. Zhang, W. Liu, E. Tsang, and B. Virginas, "Expensive multiobjective optimization by moea/d with gaussian process model," , *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 3, pp. 456-474, 2010.

[55] S. Tiwari, G. Fadel, P. Koch, and K. Deb, "Performance assessment of the hybrid archive-based micro genetic algorithm (amga) on the cec09 test problems," *IEEE Congress* on *Evolutionary Computation,* pp. 1935-1942, 2009.

[56] C.-M. Chen, Y.-p. Chen, and Q. Zhang, "Enhancing moea/d with guided mutation and priority update for multi-objective optimization," *IEEE Congress* on *Evolutionary Computation,* pp. 209-216, 2009.

[57] M. Nasir, A. K. Mondal, S. Sengupta, S. Das, and A. Abraham, "An improved multiobjective evolutionary algorithm based on decomposition with fuzzy dominance," *2011 IEEE Congress* on, *Evolutionary Computation (CEC),* Brisbane, Australia pp. 765-772, 2011.

[58] H. Li and Q. Zhang, "Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 2, pp. 284–302, 2009.

[59] Q. Zhang, A. Zhou, and Y. Jin, "RM-MEDA: A regularity model-based multi-objective estimation of distribution algorithm," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 41–63, 2008.

[60] X. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.

[61] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2014.

[62] K. Tang, X. Yao, P. N. Suganthan, C. MacNish, Y.-P. Chen, C.-M. Chen, and Z. Yang, "Benchmark functions for the CEC 2008 special session and competition on large scale global optimization," *Nature Inspired Computation and Applications Laboratory, USTC, China*, 2007.

[63] K. Tang, X. Li, P. N. Suganthan, Z. Yang, and T. Weise, "Benchmark functions for the CEC2010 special session and competition on large-scale global optimization," tech. rep., Nature Inspired Computation and Applications Laboratory, USTC, China, 2009.

[64] Z. Yang, K. Tang, and X. Yao, "Large scale evolutionary optimization using cooperative coevolution," *Information Sciences,* vol. 178, no. 15, pp. 2985–2999, 2008.

[65] M. N. Omidvar, X. Li, Y. Mei, and X. Yao, "Cooperative co-evolution with differential grouping for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 3, pp. 378–393, 2014.

[66] X. Li, K. Tang, M. N. Omidvar, Z. Yang, and K. Qin, "Benchmark functions for the CEC2013 special session and competition on large-scale global optimization," 2013.

[67] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multi-objective evolutionary algorithms: empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[68] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, "Scalable test problems for evolutionary multi-objective optimization," *TIK-Technical Report*, Zürich, Switzerland, no. 112, 2001.

[69] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable test problems for evolutionary multi-objective optimization*. Zürich, Switzerland, 2005.

[70] S. Huband, L. Barone, L. While, and P. Hingston, "A scalable multi-objective test problem toolkit," in *Proceedings of the International Conference on Evolutionary Multi-Criterion Optimization*, Berlin, Germany, pp. 280–294, 2005.

[71] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multi-objective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[72] K. Deb, "Multi-objective genetic algorithms: Problem difficulties and construction of test problems," *Evolutionary Computation*, vol. 7, no. 3, pp. 205–230, 1999.

[73] H. Jain and K. Deb, "An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, part ii: Handling constraints and extending to an adaptive approach," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 602–622, 2014.

[74] T. Okabe, Y. Jin, M. Olhofer, and B. Sendhoff, "On test functions for evolutionary multi-objective optimization," in *Parallel Problem Solving from Nature*, pp. 792–802, 2004.

[75] D. K. Saxena, Q. Zhang, J. A. Duro, and A. Tiwari, "Framework for many-objective test problems with both simple and complicated pareto-set shapes," in *Proceedings of the Evolutionary Multi-Criterion Optimization*, Ouro Preto, Brazil, pp. 197–211, 2011.

[76] R. Cheng, Y. Jin, K. Narukawa, and B. Sendhoff, "A multi-objective evoltuionary algorithm using Gaussian process based inverse modeling," *IEEE Transactions on Evolutionary Computation*, vol. 19, no. 6, pp. 838–856, 2015.

[77] Y. Jin and B. Sendhoff, "Constructing dynamic optimization test problems using the multi-objective optimization concept," in *Applications of Evolutionary Computing*, pp. 525–536, 2004.

[78] A. Zhou, Y. Jin, Q. Zhang, B. Sendhoff, and E. Tsang, "Prediction-based population re-initialization for evolutionary dynamic multi-objective optimization," in *Evolutionary Multi-Criterion Optimization*, pp. 832–846, 2007.

[79] C.-K. Goh and K. C. Tan, "A competitive-cooperative coevolutionary paradigm for dynamic multi-objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 13, no. 1, pp. 103–127, 2009.

[80] A. Konstantinidis, K. Yang, and Q. Zhang, "An evolutionary algorithm to a multi objective deployment and power assignment problem in wireless sensor networks," in *IEEE Global Telecommunications Conference*, Los Angeles, United States, Nov 2008, pp. 1-6.

[81] Q. Zhang, H. Li, D. Maringer, and E. Tsang, "Moea/d with nbi-style tchebycheff approach for portfolio management," in *IEEE Congress on Evolutionary Computation*, July 2010, pp. 1–8.

[82] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, Apr 2002.

[83] I. Das and J. E. Dennis, "Normal-boundary intersection: A new method for generating the pareto surface in nonlinear multicriteria optimization problems," *SIAM Journal on Optimization*, vol. 8, no. 3, pp. 631–657, 1998.

[84] K. Deb and H.-G. Beyer, "Self-adaptive genetic algorithms with simulated binary crossover," *Evolutionary Computation*, vol. 9, no. 2, pp. 197–221, 2001.

[85] K. Deb and M. Goyal, "A combined genetic adaptive search (GeneAS) for engineering design," *Computer Science and Informatics*, vol. 26, pp. 30–45, 1996.

[86] S. Huband, P. Hingston, L. Barone, and L. While, "A review of multiobjective test problems and a scalable test problem toolkit," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 5, pp. 477–506, 2006.

[87] D. K. Saxena, Q. Zhang, J. A. Duro, and A. Tiwari, "Framework for many-objective test problems with both simple and complicated pareto- set shapes," in *International Conference on Evolutionary Multi-Criterion Optimization*. Ouro Preto, Brazil, pp. 197–211.

[88] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, *Scalable Test Problems for Evolutionary Multiobjective Optimization*. London: Springer, 2005, pp. 105–145.

[89] R. Cheng, M. Li, Y. Tian, X. Zhang, S. Yang, Y. Jin, and X. Yao, "Benchmark functions for the cec'2017 competition on many-objective optimization," School of Computer Science, University of Birmingham, UK, Tech. Rep. CSR-17-01, 2017.

[90] P. Czyżak and A. Jaszkiewicz, "Pareto simulated annealingła meta- heuristic technique for multiple-objective combinatorial optimization," *Journal of Multi-Criteria Decision Analysis*, vol. 7, no. 1, pp. 34–47, 1998.

[91] Z. He and G. G. Yen, "Visualization and performance metric in many- objective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 386–402, 2016.

[92] Y. Tian, R. Cheng, X. Zhang, and Y. Jin, "Platemo: A matlab platform for evolutionary multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 12, no. 4, pp. 73–87, 2017.

[93] Ran Cheng, Yaochu Jin, Fellow, IEEE, Markus Olhofer and Bernhard Sendhoff, "Connections of Reference Vectors and Different Types of Preference Information in A Reference Vector Guided Evolutionary Algorithm for Many-objective Optimization," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 5, pp. 773–791, 2016.

VITA

Chaitanya Kumar Ravipalli

Candidate for the Degree of

Master of Science

Thesis: APMOEA: AN ADVANCED *p*-METRIC BASED MANY-OBJECTIVE EVOLUTIONARY ALGORITHM

Major Field: Electrical & Computer Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Electrical & Computer Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2018.

Completed the requirements for the Bachelor of Science in Electronics & Instrumentation Engineering at Gitam University, Visakhapatnam, Andhra Pradesh, India in 2015.

Experience:

Graduate Teaching Assistant, Oklahoma State University

Graduate Student Researcher, Oklahoma State University