PERFORMANCE METRICS ENSEMBLE FOR

MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

By

ZHENAN HE

Bachelor of Engineering in Automation
University of Science and Technology Beijing
Beijing, China
2008

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, 2011

PERFORMANCE METRICS ENSEMBLE FOR

MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Thesis Approved:

Dr. Gary G. Yen

Thesis Adviser

Dr. Qi Cheng

Dr. Weihua Sheng

Dr. Mark E. Payton

Dean of the Graduate College

ACKNOWLEDGMENTS

At the outset, I would like to thank my advisor Dr. Gary G. Yen, for his patience and guidance. I enjoy very much working under him and it was an excellent learning experience for me. I appreciate him for teaching me how to think about a problem, how to generate an idea to solve the problem and how to validate my idea through the experiment. I would also like to thank him for helping me out with great patience in writing this thesis, for guiding me and giving me tips on structure, grammar, sentence and words. These lessons will benefit me all my life.

I would like to think my committee members, Dr. Qi Cheng and Dr. Weihua Sheng for their timely advice, discussions and feedback. Dr. Cheng's Stochastic System class made me understand the stochastic concept in a depth way, which helps me much in my research work.

Many thanks to my lab mates at the Intelligent Systems and Control Laboratory: Chatkew Jariyatantiwait, Bin Ha and Weiwei Zhang gave me much useful suggestions. I often get inspiration by the group study with them. In summary, I would like to thank my lab mates for providing an intellectually stimulating experience via discussions and presentations throughout my tenure.

Then I strongly like to thank my friends Guanglei An and Xiaowei Yang. They often give me many suggestions in my study and life. A note of thanks for my good friends Suling Duan, Bo Xu, Huanyu Zhao, Lianfan Su and Rui Yang along with all my above mentioned friends for a great spare time and for their support and understanding.

I am forever grateful to my parents for their love, financial and moral support and encouragement.

Lastly, but definitely not the least, I dedicate this work in loving memory my grandfather, who was and will continue to be a great source of inspiration to me. I cherish the memory of him forever.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 PROBLEM STATEMENT

Evolutionary algorithms (EAs) have become established as *the* approach for exploring the Pareto-optimal front in multiobjective optimization problems. EAs usually do not guarantee to identify optimal tradeoffs but attempt to find a good approximation. From No Free Lunch theorem [1], any algorithm elevated performance over one class of problems is exactly paid for in loss over another class. Although various multiobjective EAs (MOEAs) are available today, certainly we are interested in developing a most effective algorithm to search for Pareto solutions for a given problem [2]. Therefore, comparative studies are always conducted. They aim at revealing advantages and weaknesses of the underlying methods and at determining the best performance pertained to specific problem characteristics. The numerous applications of MOEAs boost the significance of performance comparison issues. However, in absence of any established comparison criteria, none of the different sets of estimates based on various metrics for the Pareto-optimal solutions generated can be argued to be better than the others.

Zitzler [2] proposed three optimization goals to be measured: the distance of the resulting nondominated set to the Pareto-optimal front should be minimized, a good (in most cases uniform) distribution of the solutions found- in objective space- is desirable and the extent of the obtained nondominated front should be maximized. In the literature, there are many unary performance metrics used to compare MOEAs. These metrics can be broadly divided into five categories according to the optimization goals. Each category mainly evaluates the quality of a Pareto-optimal set in one aspect only. First, metrics assessing the number of Pareto optimal solutions in the set: *Pareto Dominance Indicator* (NR) [3] measures the ratio of non-dominated solutions contributed by a particular solution set to the non-dominated solutions provided by all solution sets; *Overall Non-dominated Vector Generation and Ratio* (ONVG) [4] counts the number of distinct nondominated points generated; *Ratio of Non-dominated Individuals* (RNI) [5] gives the proportion of the useful solutions known as the Pareto-front in a given population size; and *Error Ratio* (ER) [4] checks the proportion of non true Pareto points in the approximation front. Within the second category, metrics measuring the closeness of the solution to the theoretical Pareto front: *Generational Distance* (GD) [4] measures how far the evolved solution set is from the true Pareto front; and *Maximum Pareto Front Error* (MPFE) [4] focused on the largest distance between the point in the theoretical Pareto front and the point in the approximation front. Third, metrics focusing on distribution of the solutions: *Uniform Distribution* (UD) [5] measures the distribution of an approximation front under a pre-defined parameter σ_{share}; *Spacing* [6] measures how evenly the evolved solutions distribute itself; and *Number of Distinct Choices* (NDC_u) [7] identifies solutions that are sufficiently distinct for a special value u. Fourth, metrics concerning spread of the solutions: *Maximum Spread* (MS) [3] measures how well the true Pareto front is covered by the approximation set. In the last category, metrics considering both closeness and diversity at the same time: *Hypervolume Indicator* (or *S*-metric) [11] calculates the volume covered by the approximation front.

Furthermore, there are some binary performance metrics used to compare a pair of algorithms. The first type of binary performance metrics based on unary quality indicator. It includes ε -indicator. I_ε [10] defines a ε -dominate relation between algorithms, enclosing hypercube Indicator [10] and coverage difference metrics (*D*-metric) [11]. The second type is direct comparison binary metrics: *C metrics* [9] and *R metrics* [5]. *C* metrics [9] consider the dominate relations between algorithms; *R* metrics [5] use the probability that one algorithm is better than the other over a series of functions.

However, the problem arises that no single metric alone can faithfully measure MOEA performance. Every single metric can provide some specific, but incomplete quantification of performance and can only be used effectively under some specified conditions. For example, *UD* does a poor job when the Pareto front is discontinued, while Hypervolume can be misleading if the Pareto optimal front is non-convex [4]. This implies one metric cannot entirely evaluate EAs in all conditions. Every metric focuses on some special characteristics while neglects information in others. Also, every metric has its unique characteristic; no metrics can substitute others completely. Therefore, a single metrics cannot provide a comprehensive measure for MOEAs. Moreover, from [11], a fixed number of indicators are not sufficient to evaluate MOEAs because reducing objective space must losing information.

Different metrics perform differently in different test problems. For a given MOEA, one metric may show well in one test problem, however, given other test problems, it may mislead the conclusion should the measures show poorly. For a specific test problem, we cannot ascertain which metric should be applied in order to faithfully quantify the performance of MOEAs. We need to exploit every metric to find which one is the best. Apparently, this introduces a heavy computational process.

1.2 MOTIVATION

To overcome these disadvantages and arrive at a faithful evaluation of MOEAs, performance metrics ensemble is proposed in this research work. Ensemble methods use multiple metrics to obtain a fair performance than what could be obtained from any of single performance metric alone. Ensemble metrics not only can give the comprehensive comparison between different algorithms, but avoid the choosing process and can be directly used to assessing MOEAs.

There exists no publication in the literature, to our best knowledge, regarding performance metrics ensemble. MOEAs are only evaluated and compared in a single metric at a time. In this paper, we propose double-tournament selection operator to compare many approximation fronts from different MOEAs. Double elimination design allows characteristic poor performance of a quality algorithm under the special environment still to be able to win it all. In every competition, one metric is chosen randomly to compare. After the whole process, every metric could be selected multiple times and a final winning algorithm is to be identified. This final winner has been compared under all the metrics considered so that we can make a fair conclusion.

1.3 ORGANIZATION OF THESIS

Chapter Two provides the consolidated literature review for this thesis. It presents the essential background with reference to knowledge in the areas of Multiobjective Optimization Problem, Multiobjective Evolutionary Algorithms and Performance Metrics.

Chapter Three describes the proposed approach in detail. A novel ensemble method using modified Double-Tournament selection operator is introduced.

In **Chapter Four**, we elaborate on the experiment results for ZDT (1-6) and DTLZ 2 problems.

Finally, conclusion is drawn in **Chapter Five** along with pertinent observations.

CHAPTER TWO

REVIEW OF LITERATURE

This chapter presents the essential background knowledge on Multiobjective Optimization Problem (MOP), Multiobjective Evolutionary Algorithm (MOEA) and Performance Metrics.

2.1 MULTIOBJECTIVE OPTIMIZATION PROBLEMS (MOP)

First, the omnipresent of MOP is discussed. Then, definition of MOP is given. After that, concept of domination is introduced to identify the optimal solution. In addition, the characteristic of reference sets of MOP is summarized. Finally, two series of test problems are presented, which are widely applied to evaluate the performance of multiobjective evolutionary algorithms.

2.1.1 Why Multiobjective Optimization Problems?

In real life environments we always strive to optimize a number of parameters in any design and these parameters are usually highly correlated. Hence, some tradeoff between the criteria is needed to ensure a satisfactory design. For example: in bridge construction, a good design is characterized by low total mass and high stiffness; aircraft design requires simultaneous optimization of fuel efficiency, payload, and weight; a good sunroof design in a sport car could aim at minimizing the noise the driver hears and maximizing the ventilation; and the business portfolio management attempts to simultaneously minimize the risk and maximize the fiscal return. In these real-world optimization problems, the objectives often conflict across a high-dimensional problem space and may also require extensive computational resources.

Neither the problem nor algorithm domains considered within this research is straightforward. Multiobjective Optimization Problems (MOPs) present a possibly uncountable set of solutions that produce vectors whose components represent trade-offs in objective space. Therefore, for an MOP, a number of trade-off solutions are optimal. Without further information, such optimal solutions are equally important.

2.1.2 MOP Definition [39]

In general, an MOP involves k objectives, m constraints and n decision variables:

k objectives: Optimizes $f(x) = \left(f_1(x), \ldots, f_k(x)\right)$ (2-1 A)

m equality and inequality constraints: Subject to $g_i(x) \leq 0, i = 1, \ldots, m$ (2-1 B)

n decision variables: $x = \left(x_1, \ldots, x_n\right)^T$ (2-1 C)

MOP deals with two search spaces: a decision space (Ω) plus an objective space (Λ). Mapping takes place from an n-dimensional decision space to an m-dimensional objective space. The MOP's objective function, $f : \Omega \rightarrow \Lambda$, maps decision variables $x = \left(x_1, \ldots, x_n\right)$ in decision space to vectors $f(x) = \left(f_1(x), \ldots, f_k(x)\right)$ in objective space. Proximity of two solutions in one space does not imply proximity in the other space and search is performed in the decision space.

As stated in [2], the goal of MOPs consists of multiple objectives: the distance of the resulting nondominated set to the Pareto-optimal front should be minimized; a good (in most cases uniform) distribution of the solutions found is desirable; and the extent of the obtained nondominated front should be maximized, i.e., for each objective, a wide range of values should be covered by the nondominated solutions. Apparently, while these candidate solutions are progressing towards Pareto-optimal front, the convergence process will adversely impact the spread of the solution found.

2.1.3 Concept of Domination [39]

Most multi-objective optimization algorithms use the concept of domination. In these algorithms, two solutions are compared on the basis of whether one dominates the other solution or not.

- Pareto Optimality:

  A solution $x \in \Omega$ is said to be Pareto optimal with respect to Ω if and only if there is no $x^{'} \in \Omega$ for which $v = f\left(x^{'}\right) = \left(f_1\left(x^{'}\right), \ldots, f_k\left(x^{'}\right)\right)$ dominates $u = f\left(x\right) = \left(f_1\left(x\right), \ldots, f_k\left(x\right)\right)$.

- Pareto Dominance:

  A vector $u = f\left(x_u\right) = \left(u_1, \ldots, u_k\right)$ is said to dominate $v = f\left(x_v\right) = \left(v_1, \ldots, v_k\right)$ (denoted by $u \preceq v$) if and only if u is worse than v, $\forall i \in \{1, \ldots, k\}, u_i \preceq v_i$ and $\exists i \in \{1, \ldots, k\}, u_i \prec v_i$

- Pareto Optimal Set:

  For a given MOP $f\left(x\right)$, $x \in \Omega$ the Pareto optimal set P^* is defined as:

$$P^* := \left\{ x \in \Omega \middle| \neg \exists x^{'} \in \Omega : f\left(x^{'}\right) \leq f\left(x\right) \right\}$$

- Pareto Front (Non-dominated front):

  For a given MOP $f\left(x\right)$ and Pareto optimal set P^*, the Pareto front PF^* is defined as:

$$PF^* := \left\{ u = f\left(x\right) = \left(f_1\left(x\right), \ldots, f_k\left(x\right)\right) \middle| x \in P^* \right\}$$

Figure 1 explains the relation between decision space and objective space and the corresponding front in each space for a given problem.

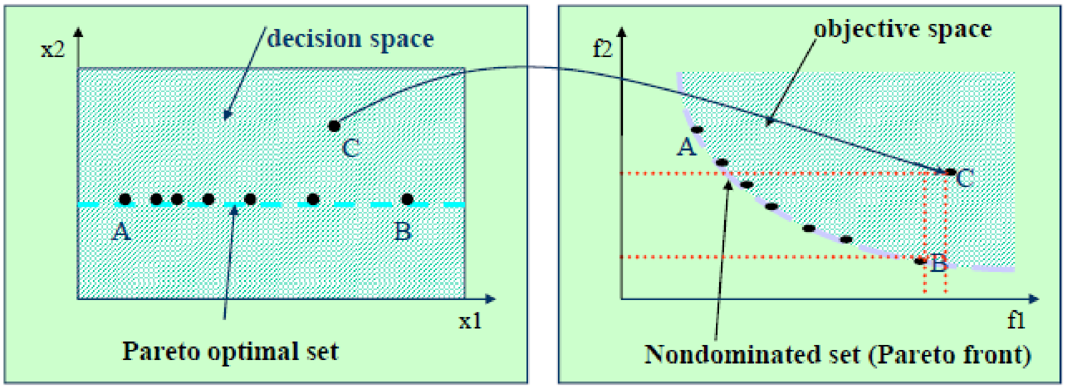


Fig 2.1 The Relation between Decision Space and Objective Space

Moreover, there are some special points in the objective space [39]. *Ideal Objective Vector* (Reference Solutions Z^*) is the lower bound in the Pareto-optimal set. The m-th component of the ideal objective vector is the constrained minimum solution of the following problem: Minimize $f_m(x)$, subject to $x \in S$. $Z^* = f^* = \left(f_1^*, f_2^*, \ldots, f_M^*\right)^T$. *Utopian Objective Vector* (Z^{**}) has each of its components marginally smaller than that of the Ideal Objective Vector**.** $Z^{**} = Z_i^* - \varepsilon_i$ with $\varepsilon_i > 0$ for all $i = 1, 2, \ldots, M$ and Nadir Objective Vector (Z^{nad}) is upper bound in the Pareto-optimal set.

2.1.4 ZDT problems [2]

ZDT problems were proposed by Deb in 1999 and consist of six benchmark functions. ZDT contains several characteristics that cause difficulties for multiobjective evolutionary algorithms: for converging to the Pareto-optimal front, multimodality, deception and isolated optima are applied and for maintaining diversity within the population, convexity or nonconvexity, discreteness, and nonuniformity in the front.

Each of the test functions is structured in the same manner and consists itself of three functions f_1, g, h:

Minimize $f(x) = (f_1(x), f_2(x))$ (2-2)

Subject to $f_2(x) = g(x_2, \ldots, x_n) h(f_1(x_1), g(x_2, \ldots, x_n))$,

where $x = (x_1, \ldots, x_n)$

- ZDT1 (Convex Pareto-optimal front):

$$f_1(x) = x_1 \tag{2-3 A}$$

$$f_2(x) = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}}\right] \tag{2-3 B}$$

$$g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \tag{2-3 C}$$

$x = (x_1, ..., x_n)^T \in [0,1]^n$. Given $n = 30$, the Pareto-optimal front is convex and formed

with $g(x) = 1$.Figure 2.2 shows the true Pareto-optimal front of ZDT1.

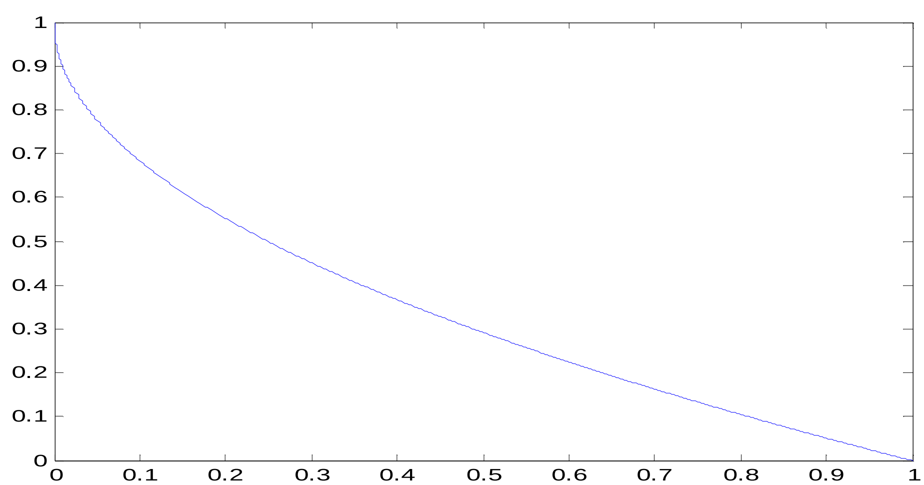


Fig 2.2 Pareto-optimal front of ZDT1

- ZDT2 (Nonconvex Pareto-optimal front):

$$f_1(x) = x_1 \tag{2-4 A}$$

$$f_2(x) = g(x)\left[1 - \left(\frac{f_1(x)}{g(x)}\right)^2\right] \tag{2-4 B}$$

$$g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \tag{2-4 C}$$

$x = (x_1, ..., x_n)^T \in [0,1]^n$. Given $n = 30$, the Pareto-optimal front is nonconvex and formed with $g(x) = 1$.

Figure 2.3 shows the Pareto-optimal front of ZDT2.

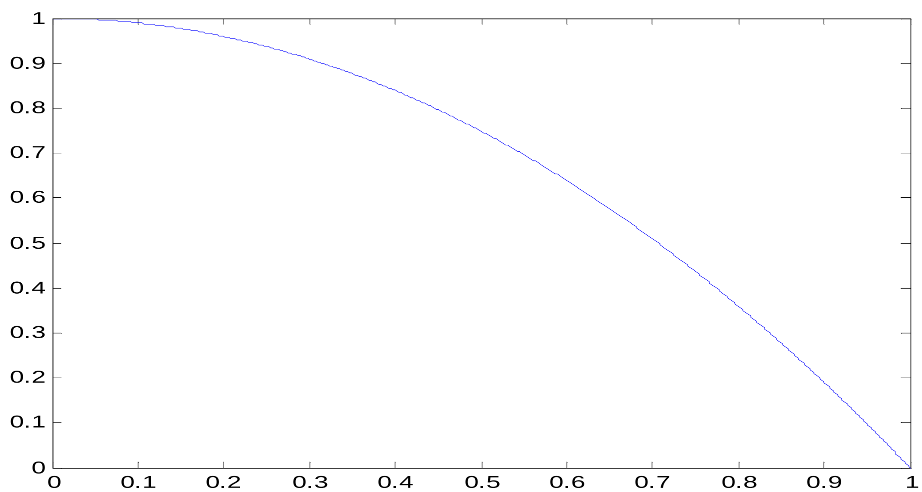


Fig 2.3 Pareto-optimal front of ZDT2

- ZDT3 (Discrete Pareto-optimal front):

$$f_1(x) = x_1 \tag{2-5 A}$$

$$f_2(x) = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}} - \frac{f_1(x)}{g(x)}\sin(10\pi x_1)\right] \tag{2-5 B}$$

$$g(x) = 1 + \frac{9\left(\sum_{i=2}^{n} x_i\right)}{n-1} \tag{2-5 C}$$

$x = (x_1, ..., x_n)^T \in [0,1]^n$, Given $n = 30$, its Pareto-optimal front is disconnected and formed with $g(x) = 1$. The two objectives are disparately scaled in the Pareto-optimal front; f_1 is from 0 to 0.852 and f_2 from -0.773 to 1. The introductions of the sine function in h causes discontinuity in the Pareto-optimal front.

Figure 2.4 shows the Pareto-optimal front of ZDT3.

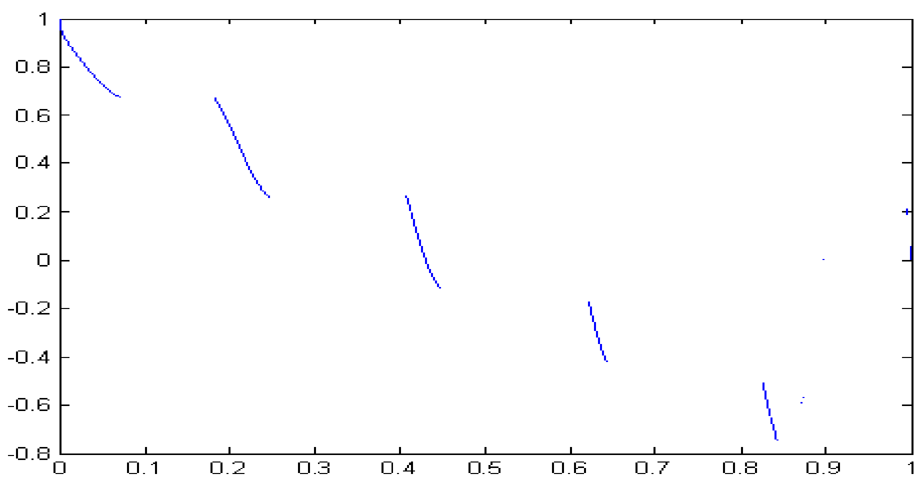


Fig 2.4 Pareto-optimal front of ZDT3

- ZDT4 (Lots of local Pareto-optimal front):

$$f_1(x) = x_1 \tag{2-6 A}$$

$$f_2(x) = g(x)\left[1 - \sqrt{\frac{f_1(x)}{g(x)}}\right] \tag{2-6 B}$$

$$g(x) = 1 + 10(n-1) + \sum_{i=2}^{n}\left[x_i^2 - 10\cos(4\pi x_i)\right] \tag{2-6 C}$$

$x = (x_1, \ldots, x_n)^T \in [0,1]^n \times [-5,5]^{n-1}$. Given $n = 10$. It has many local Pareto-optimal fronts.

The global Pareto-optimal front is formed with $g(x) = 1$, the best local Pareto-optimal front with $g(x) = 1.25$. Not all local Pareto-optimal sets are distinguishable in the objective space.

Figure 2.5 shows the Pareto-optimal front of ZDT4.

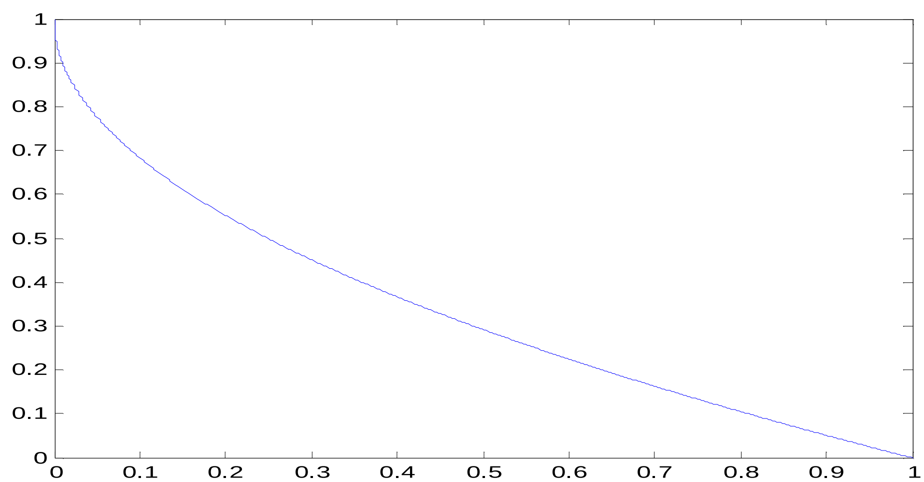


Fig 2.5 Pareto-optimal front of ZDT4

- ZDT5 (Deceptive problem):

$$f_1(x) = 1 + u(x_1) \tag{2-7 A}$$

$$f_2(x) = g(x) / f_1(x) \tag{2-7 B}$$

$$g(x_2,...,x_n)=\sum_{i=2}^{n} v(u(x_i)) \qquad \text{(2-7 C)}$$

$u(x_i)$ gives the number of ones in the bit vector x_i:

$$v(u(x_i))=\begin{cases} 2+u(x_i) & u(x_i)<5 \\ 1 & u(x_i)=5 \end{cases} \qquad \text{(2-7 D)}$$

Given $n=11$, $x_1 \in \{0,1\}^{30}$ and $x_2, \ldots, x_n \in \{0,1\}^5$. The true Pareto-optimal front is formed with $g(x)=10$. The global Pareto-optimal fronts as well as the local ones are convex.

- ZDT6 (Pareto-optimal solutions are nonuniformity):

$$f_1(x)=1-\exp(-4x_1)\sin^6(6\pi x_1) \qquad \text{(2-8 A)}$$

$$f_2(x)=g(x)\left[1-\left(\frac{f_1(x)}{g(x)}\right)^2\right] \qquad \text{(2-8 B)}$$

$$g(x)=1+9\left[\frac{\left(\sum_{i=2}^{n} x_i\right)}{n-1}\right]^{0.25} \qquad \text{(2-8 C)}$$

$x=(x_1,...,x_n)^T \in [0,1]^n$. Given $n=10$. Its Pareto-optimal front is nonconvex. The distribution of the Pareto solutions in the Pareto front is nonuniform, i.e., for a set of uniformly distributed points in the Pareto set in the decision space, their images crowd in a corner of the Pareto front in the objective space. The density of the solutions is lowest near the Pareto-optimal front and highest away from the front.

Figure 2.6 shows the Pareto-optimal front of ZDT6.

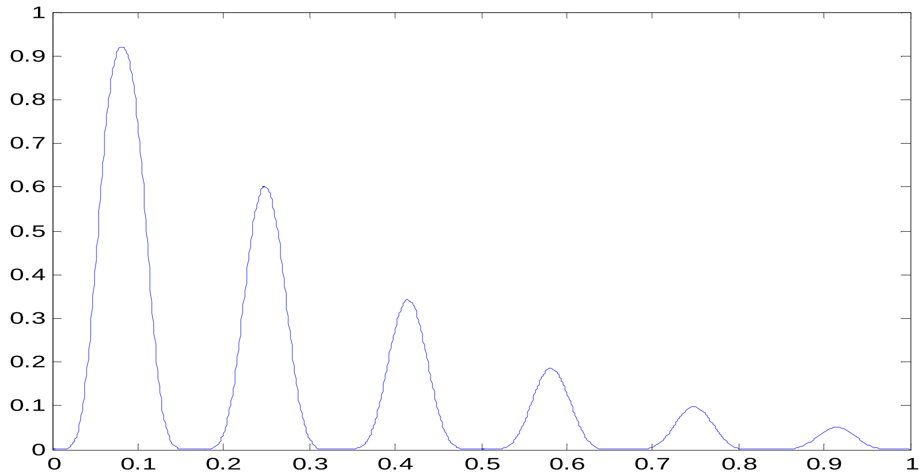


Fig 2.6 Pareto-optimal front of ZDT6

2.1.5 Test Problem DTLZ [12]

DTLZ contains seven benchmark functions. All the functions have more than two objectives. Like ZDT problems, DTLZ also contains problem characteristics that present difficulties for multiobjective evolutionary algorithms. Therefore, DTLZ test MOEAs' ability to deal with high-dimension problems.

- DTLZ 1:

An *M*-objective problem with a linear Pareto-optimal front:

$$\text{Minimize}\begin{cases} f_1(x) = \frac{1}{2}x_1x_2\cdots x_{M-1}\left(1+g\left(x_M\right)\right) \\ f_2(x) = \frac{1}{2}x_1x_2\cdots\left(1-x_{M-1}\right)\left(1+g\left(x_M\right)\right) \\ \ldots\ldots \\ f_{M-1}(x) = \frac{1}{2}x_1\left(1-x_2\right)\left(1+g\left(x_M\right)\right) \\ f_M(x) = \frac{1}{2}\left(1-x_1\right)\left(1+g\left(x_M\right)\right) \end{cases} \tag{2-9 A}$$

Subject to $0 \le x_i \le 1,\ i = 1, 2, \ldots, n$

$$g(x_M) = 100\left(|x_M| + \sum_{x_i \in X_M} (x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right) \tag{2-9 B}$$

The Pareto-optimal solution corresponds to $x_i^* = 0.5$ ($x_i^* \in x_M$) and the objective function values lie on the linear hyperplane: $\sum_{m=1}^{M} f_m^* = 0.5$. $k = 5$ is suggested here. The total number of variables is $M + k - 1$. The search space contains $(11^k - 1)$ local Pareto-optimal fronts. The difficulty in this problem is to converge to the hyperplane.

- DTLZ 2:

An M-objective problem with a Spherical Pareto-optimal front:

$$\text{Minimize}\begin{cases} f_1(x) = (1 + g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2) \\ f_2(x) = (1 + g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2) \\ \text{L L} \\ f_M(x) = (1 + g(x_M))\sin(x_1\pi/2) \end{cases} \tag{2-10 A}$$

Subject to $0 \le x_i \le 1,\ i = 1, 2, \text{K}\ , n$

$$g(X_M) = \sum_{x_i \in X_M} (x_i - 0.5)^2 \tag{2-10 B}$$

The Pareto-optimal solution corresponds to $x_i^* = 0.5$ ($x_i^* \in x_M$) and all objective function values must satisfy: $\sum_{m=1}^{M} (f_m^*)^2 = 1$. $k = 10$ is suggested here. The total number of variables is $M + k - 1$.

- DTLZ 3:

An M-objective problem with a Global Pareto-optimal front:

$$\text{Minimize}\begin{cases} f_1(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\cos(x_{M-1}\pi/2) \\ f_2(x) = (1+g(x_M))\cos(x_1\pi/2)\cdots\sin(x_{M-1}\pi/2) \\ \text{L L} \\ f_M(x) = (1+g(x_M))\sin(x_1\pi/2) \end{cases} \tag{2-11 A}$$

Subject to $0 \le x_i \le 1$, $i = 1,2,\text{K} \ ,n$

$$g(x_M) = 100\left(|x_M| + \sum_{x_i \in X_M}(x_i - 0.5)^2 - \cos(20\pi(x_i - 0.5))\right) \tag{2-11 B}$$

The difficult is that this problem introduces many local Pareto-optimal fronts.

- DTLZ 4:

This problem measures MOEA's ability to maintain a good distribution of solutions:

$$\text{Minimize}\begin{cases} f_1(x) = (1+g(x_M))\cos(x_1^\alpha\pi/2)\cdots\cos(x_{M-1}^\alpha\pi/2) \\ f_2(x) = (1+g(x_M))\cos(x_1^\alpha\pi/2)\cdots\sin(x_{M-1}^\alpha\pi/2) \\ \text{L L} \\ f_M(x) = (1+g(x_M))\sin(x_1^\alpha\pi/2) \end{cases} \tag{2-12 A}$$

Subject to $0 \le x_i \le 1$, $i = 1,2,\text{K} \ ,n$, $\alpha = 100$.

$$g(x_M) = \sum_{x_i \in X_M}(x_i^\alpha - 0.5)^2 \tag{2-12 B}$$

- DTLZ 5:

This problem will test an MOEA's ability to converge to a degenerated curve:

Mapping

$$\theta_i = \frac{\pi}{4(1+g(r))}(1+2g(r)x_i) \tag{2-13 A}$$

$i = 2, 3, \ldots, M-1$, in the test problem of DTLZ 2.

$$g(x_M) = \sum_{x_i \in X_M} x_i^{0.1} \qquad \text{(2-13 B)}$$

- DTLZ 6:

This problem has 2^{M-1} disconnected Pareto-optimal regions in the search space:

$$\text{Minimize} \begin{cases} f_1(x_1) = x_1 \\ f_{M-1}(x_{M-1}) = x_{M-1} \\ \cdots \cdots \\ f_M(x) = (1 + g(x_M)) h(f_1, f_2, \ldots, f_{M-1}, g) \end{cases} \qquad \text{(2-14 A)}$$

Subject to $0 \le x_i \le 1$, $i = 1, 2, \ldots, n$,

$$g(x_M) = 1 + \frac{9}{|x_M|} \sum_{x_i \in x_M} x_i \qquad \text{(2-14 B)}$$

$$h = M - \sum_{i=1}^{M-1} \left[\frac{f_i}{1+g} \left(1 + \sin(3\pi f_i)\right) \right] \qquad \text{(2-14 C)}$$

$k = 20$ is suggested here. The total number of variables is $M + k - 1$. This problem will test an algorithm's ability to maintain subpopulation in different Pareto-optimal regions.

- DTLZ 7:

This problem is constructed by constraint surface approach:

$$\text{Minimize: } f_j(x) = \frac{1}{\left\lfloor \frac{n}{M} \right\rfloor} \sum_{i = \left\lfloor (j-1)\frac{n}{M} \right\rfloor}^{\left\lfloor j \frac{n}{M} \right\rfloor} x_i, \quad j = 1, \ldots, M \qquad \text{(2-15 A)}$$

$$\text{Such that } g_j(x) = f_M(x) + 4 f_j(x) - 1 \ge 0, \quad j = 1, \ldots, M-1 \qquad \text{(2-15 B)}$$

$$g_M(x) = 2 f_M(x) + \min_{\substack{i,j=1 \\ i \neq j}}^{M-1} \left[f_i(x) + f_j(x) \right] - 1 \geq 0 \qquad \text{(2-15 C)}$$

subject to space $0 \leq x_i \leq 1,\ i = 1, 2, \ldots, n$

$n = 10M$. There are a total of M constraints. It is difficult for MOEAs to handle these constraints while searching for the optimal solutions. Moreover, there are some non-dominated solutions in the final population but not the true Pareto-optimal solutions. This problem is called redundant solutions. Because of redundant solutions, the obtained set of solutions may incorrectly find a higher-dimensional surface as the Pareto-optimal front.

2.2 MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAs)

This part includes MOEA introduction, MOEA definition, MOEA's main procedure, and parameters in controlling the behavior of the GA and some examples of MOEAs.

2.2.1 An Introduction to MOEA [39]

MOEA is motivated from the conception of evolution in biology, specifically Darwin's "survival of the fittest" (natural selection) law. Figure 2.7 describes how MOEA works. It includes two steps. In the first step, an ideal Multiobjective Optimizer finds an optimal front consists of multiple trade-off solutions. Then, in the second step, based on some high-level information, one solution is chosen for implementation.

Evolutionary algorithms (EAs) have become an effective tool for exploring the Pareto-optimal front in multiobjective optimization problems that are often too complex to be solved by exact methods, such as linear programming or gradient search. This is because there are few alternatives for searching intractably large spaces for multiple Pareto-optimal solutions. Due to their inherent parallelism and their ability to exploit similarities of solutions by recombination, they are able to approximate the Pareto-optimal front in a single optimization run. The numerous applications and the rapidly growing interest in the area of MOEAs take this fact into account.

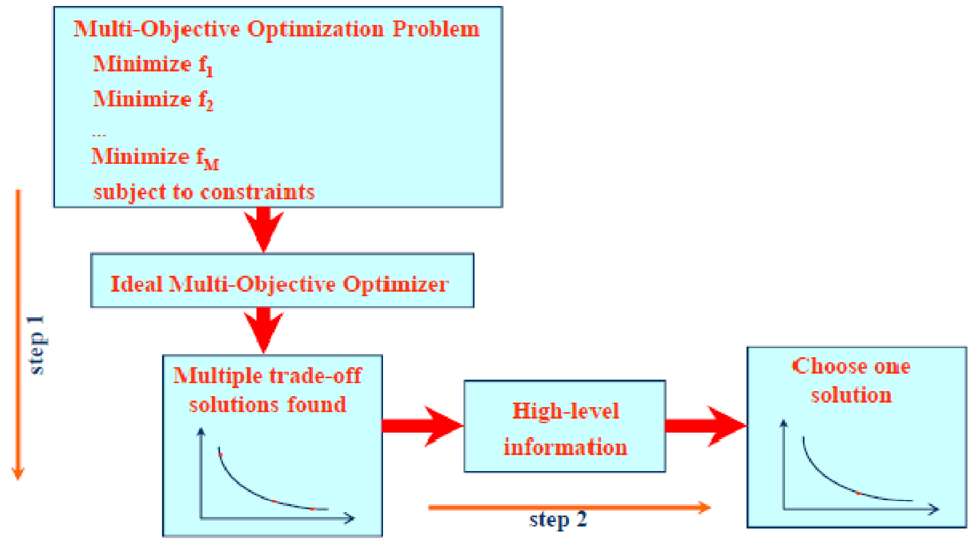


Fig 2.7 The Process of MOEAs to solve MOPs

From Zitzler and Deb [2], the first pioneering studies on evolutionary multiobjective optimization appeared in the mid-eighties by Schaffer in 1984 [26] and Fourman in 1985[27]. After that, several different EA implementations were proposed from 1991 to 1994: Kursawe in 1991[28]; Hajela and Lin in 1992 [29]; Fonseca and Fleming in 1993[30]; Horn in 1994 [31]; and Srinivas and Deb in 1994 [32]. Later, these approaches were successfully applied to various multiobjective optimization problems. In recent years, some researchers have investigated particular topics of evolutionary multiobjective search, such as convergence to the Pareto-optimal front by Van Veldhuizen and Lamont [33] and Rudolph [34], niching by Obayashi [35], and elitism by Parks and Miller [36]; while others have concentrated on developing new evolutionary techniques, such as Laumanns [37] and Zitzler and Thiele [38].

Today, there are some MOEAs frequently used: SPEA 2 by Zitzler [13], NSGA-II by Deb [14], IBEA by Zitzler [15], PESA-II by Corne [16] and MOEA/D by Zhang [17].

Also, there are alternatives to EAs [39]: Evolutionary Programming (EP) and Genetic Programming (GP). EP is a mutation-based evolutionary algorithm to discrete search spaces. Similar to EA, a self-adopting rule is used to update the mutation strengths. Therefore, EP is allowed to search anywhere in the real space likes real-parameter GAs. In GP, a terminal set *T* (constants and variables) and a function set *F* (operators and basic functions) are pre-specified to create initial population [41].

2.2.2 MOEAs Definition [39]

The basic structure of MOEA is shown in Fig 2.8. It includes: initialize population, evaluate population, scale population fitnesses, select solutions for next population and perform crossover and mutation.

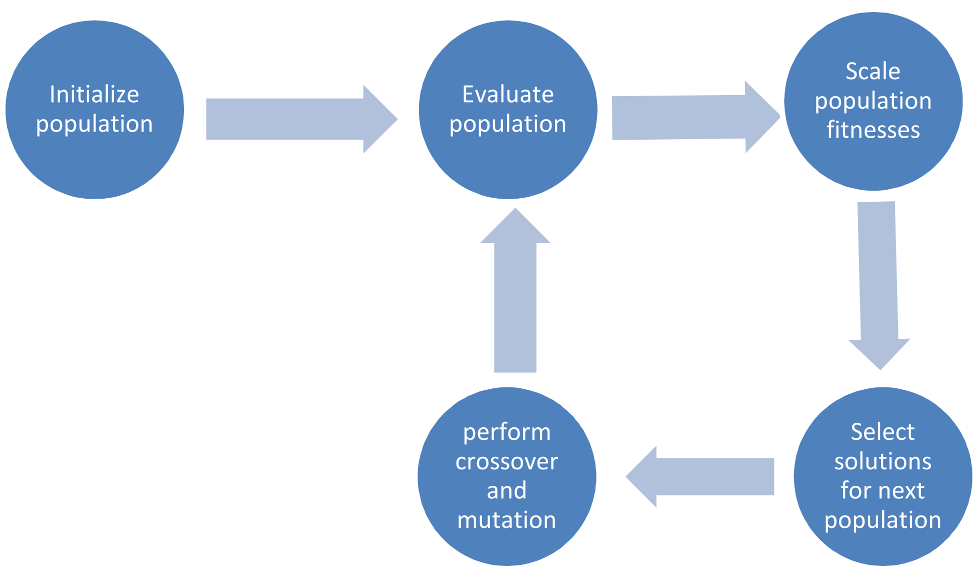


Fig 2.8 Basic Structure of MOEAs

It is a generic population-based metaheuristic algorithm inspired by biological evolution and can be viewed as an evolutionary process. It is characterized by the following components:

A genetic representation (or an encoding) for the feasible solutions to the optimization problem: The genetic representation may differ considerably from the natural form of the parameters of the solutions. Fixed-length and binary encoded strings for representing solutions have dominated GA research since they provide the maximum number of schemata as they are amenable to simple implementation.

A population of encoded solutions evolves over a sequence of generations: The population contains not just a sample of *n* ideas; rather it contains a multitude of notions and rankings of those notions for task performance. Genetic algorithms ruthlessly exploit this wealth of information by reproducing high quality notions according to their performance and crossing these notions with many other high-performance notions from other strings.

A fitness function that evaluates the optimality of each solution: During each generation, the fitness of each solution is evaluated, and solutions are selected for reproduction based on their fitness. The 'goodness' of a solution is determined from its fitness value.

Genetic operators generate a new population from the existing population: The selected solutions then undergo recombination under the action of the crossover and mutation operators.

Control parameters: control every step of evolutionary process.

2.2.3 The main procedure of MOEAs [39]

The first one is Reproduction or Selection Operator. Reproduction Operator makes duplicates of good solutions and eliminates bad solutions while keeping the whole population size constant. The whole process inscludes three steps: identify good solutions in a population, make multiple copies of good solutions and eliminate bad solutions from the population so that multiple copies of good solutions can be placed in the population.

There are three types of Selection Operators to achieve the above task:

Tournament selection is played between two solutions and the better solution is chosen and placed in the mating pool. Each solution can be made to participate in exactly two tournaments and any solution in a population has 0, 1 or 2 copies in new populations.

Proportionate selection assigns each solution copies, the number of which is proportional to their fitness value. For instance, a solution with a fitness value f_i will get f_i / f_{avg} number of copies. f_{avg} denotes the average fitness of all population members. Therefore, the solution with a higher fitness value represents a large range of cumulative probability values and has a higher probability of being copied into the mating pool. Considering the scaling problem, the outcome of operator is dependent on the true value of the fitness rather than relative fitness values.

Ranking selection sorts solutions according to their fitness (true value) from the worst to the best. Each member is assigned a fitness (relative value) equal to the rank of the solution. In most cases, proportionate selection is applied with the ranked fitness values.

The second one is Crossover Operator. The power of GAs arises from crossover. Crossover causes a structured, yet randomized exchange of genetic material between solutions, with the possibility that 'good' solutions can generate 'better' ones.

Crossover occurs only with some probability p_c (the crossover probability or crossover rate). When the solutions are not subjected to crossover, they remain unmodified.

The third one is Mutation Operator. Mutation appears to be more useful than crossover when the population size is small while there is evidence that crossover can be more useful than mutation when the population size is large. Other factors, such as the representation, selection scheme, and the fitness function itself may all have an effect on the relative utility of crossover and mutation.

2.2.4 Control Parameters of Genetic Algorithms [40]

To control the behavior of the GA, the role of the parameters p_c and p_m are often considered. The choice of p_c and p_m is known to critically affect the behavior and performance of the GA.

The crossover probability p_c controls the rate at which solutions are subjected to crossover. The higher the value of p_c the quicker are the new solutions introduced into the population. As p_c increases, however, solutions can be disrupted faster than selection can exploit them. Typical values for p_c are in the range [0.5, 1.0].

Mutation is only a secondary operator to restore genetic material. Large values of p_m transform the GA into a purely random search algorithm, while some mutation is required to prevent the premature convergence of the GA to suboptimal solutions. Typically p_m is chosen in the range [0.005, 0.05].

The balance between converge and spread characteristics of the GA is dictated by the values of p_c and p_m. Increasing values of p_c and p_m promotes exploration at the expense of exploitation. Moderately large values of p_c (0.5-1.0) and small values of p_m (0.001-0.05) are commonly employed in GA practice.

Moderately large values of p_c, ($0.5 < p_c < 1.0$), and small values of p_m ($0.001 < p_m < 0.05$) are essential for the successful working of GAs. The moderately large values of p_c promote the extensive recombination of schemata, while small values of p_m are necessary to prevent the disruption of the solutions.

2.2.5 Some Examples of MOEAs

Today, there are some MOEAs frequently used: SPEA 2 by Zitzler [13], NSGA-II by Deb [14], IBEA by Zitzler [15], PESA-II by Corne [16] and MOEA/D by Zhang [17]. In the experiment conducted in Chapter 4, these five algorithms are compared under the performance metric ensemble.

First, we consider *Strength Pareto Evolutionary Algorithm 2* (*SPEA 2*). SPEA is an effective technique for finding or approximating the Pareto-optimal set for multiobjective optimization problems. It has shown very good performance in comparison to other MOEAs. Based on SPEA, SPEA 2 incorporates a fine-grained fitness assignment strategy, a density estimation technique, and an enhanced archive truncation method.

The main structure of SPEA 2 is presented in Table 2.1. Table 2.1 includes input and output of the algorithm, and the process of the algorithm to deal with MOPs:

- Given input:
  N (population size), $\overline{N}$ (archive size) and T (maximum number of generations)
- Required output:
  A (nondominated set)

Step1: Initialization:
Generate an initial population P_0 and create the empty archive (external set) $\overline{P}_0 = \phi$. Set $t = 0$.

Step2: Fitness assignment:
Calculate fitness values of individuals in P_t and $\overline{P}_t$

Step 3:Environmental selection:
Copy all nondominated individuals in P_t and $\overline{P}_t$ to $\overline{P}_{t+1}$. If size of $\overline{P}_{t+1}$ exceeds $\overline{N}$ then reduce $\overline{P}_{t+1}$ by means of the truncation operator, otherwise if size of $\overline{P}_{t+1}$ is less than $\overline{N}$ then fill $\overline{P}_{t+1}$ with dominated individuals in P_t and $\overline{P}_t$.

Step 4:Termination:
If $t \geq T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in $\overline{P}_{t+1}$.

Step 5:Mating selection:
Perform binary tournament selection with replacement on $\overline{P}_{t+1}$ in order to fill the mating pool.

Step 6: Variation:
Apply recombination and mutation operators to the mating pool and set P_{t+1} to the resulting population. Increment generation counter $(t = t+1)$ and go to Fitness assignment step again.

Table 2.1 The main structure of SPEA 2

In fitness assignment, each individual i in P_t and $\overline{P}_t$ is assigned a strength value $S(i)$ representing the number of solutions it dominates:

$$S(i) = \left| \left\{ j \,\middle|\, j \in P_t + \overline{P}_t \wedge i \succ j \right\} \right| \tag{2-16 A}$$

$|\cdot|$ denotes the cardinality of a set, + stands for multiset union and $\succ$ corresponds to the Pareto dominance relation.

Raw fitness $R(i)$ is determined by the strengths of its dominators in both archive and population:

$$R(i) = \sum_{j \in P_t + \overline{P}_t,\, j \succ i} S(j) \tag{2-16 B}$$

Density information is incorporated into differentiate individuals having identical raw fitness values:

$$D(i) = \frac{1}{\sigma_i^k + 2} \tag{2-16 C}$$

where $k = \sqrt{N + \overline{N}}$ is the square root of the sample size and σ_i^k is the k-th element gives the distance sought.

Finally, for an individual i, its fitness

$$F(i) = R(i) + D(i) \tag{2-16 D}$$

In Environmental Selection, first, all nondominated individuals have fitness lower than one are copied to the archive of the next generation:

$$\overline{P}_{t+1} = \left\{ i \,\middle|\, i \in P_t + \overline{P}_t \wedge F(i) < 1 \right\} \tag{2-17 A}$$

Then, there are three conditions:

If $\left|\overline{P}_{t+1}\right| = \overline{N}$, the environmental selection step is completed

If $\left|\overline{P}_{t+1}\right| < \overline{N}$, the best $\overline{N} - \left|\overline{P}_{t+1}\right|$ dominated individuals with $F(i) \geq 1$ in the previous archive and population are copied to the new archive

If $\left|\overline{P}_{t+1}\right| > \overline{N}$, procedures are made to remove individuals from $\overline{P}_{t+1}$ until $\left|\overline{P}_{t+1}\right| = \overline{N}$

At each iteration, individual i is chosen for removal for which $i \leq_d j$ for all $j \in \overline{P}_{t+1}$:

$$\begin{aligned} i \leq_d j :\Leftrightarrow & \forall 0 < k < \left|\overline{P}_{t+1}\right| : \sigma_i^k = \sigma_j^k \vee \\ & \exists 0 < k < \left|\overline{P}_{t+1}\right| : \left[\left(\forall 0 < l < k : \sigma_i^l = \sigma_j^l\right) \wedge \sigma_i^k < \sigma_j^k\right] \end{aligned} \qquad \text{(2-17 B)}$$

σ_i^k denotes the distance of i to its k -th nearest neighbor in $\overline{P}_{t+1}$. The individual with the minimum distance to another individual is preferred.

Second, we introduce *Non-Dominated Sorting Genetic Algorithm-II* (NSGA-II). A non-dominated sorting based multi-objective evolutionary algorithm NSGA-II, has two advantages: computational complexity is $O\left(mN^2\right)$; a selection operator is presented to create a mating pool by combining the parent and child populations and selecting the best (with respect to fitness and spread) N solutions.

The main structure of NSGA-II is presented in Table 2.2.

The third one is *Region-based Selection in Evolutionary Multiobjective Optimization* (PESA-II). PESA-II proposes a new selection technique, called Region-Based selection, for evolutionary multiobjective optimization algorithms in which the unit of selection is a hyperbox in objective space. A hyperbox is selected and the resulting selected individual is randomly chosen from this hyperbox.

The fourth one is *Indicator-Based Selection in Multiobjective Search* (IBEA). IBEA is combined with arbitrary indicators and adapted to the preferences of the user and moreover does not require any additional diversity preservation mechanism to be used. It provides an approach to allow preference information of the decision maker be integrated into multiobjective search.

Step1: Generate random parent population P_0 and its child population Q_0: Initially, a random parent population P_0 is created. The population is sorted based on the non-domination. Each solution is assigned rank equal to its non-domination level where 1 is the best level. Thus, minimization of fitness is assumed. Binary tournament selection, recombination, and mutation operators are used to create a child population Q_0 of 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 = nondominated-sort (R_t) until $\lvert P_{t+1} \rvert < N$ 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 before the size exceeds N .
Step4: crowding-distance-assignment (F_i) and $P_{t+1} = P_{t+1} \cup F_i$ Calculate crowding distance in F_i and include k -th non-dominated front in the parent population.
Step5: Sort ($P_{t+1}, \geq_n$) and $P_{t+1} = P_{t+1}\left[0:N\right]$ Sort in descending order and choose the first N elements of P_{t+1} .
Step6: Q_{t+1} =make-new-pop (P_{t+1}) This population of size N is now used for selection, crossover and mutation to create a new population.

Table 2.2 The main structure of NSGA-II

Table 2.3 presents the input and output of the algorithm, and the process of the algorithm to deal with multiobjective problem.

The final one is MOEA/D: *A Multiobjective Evolutionary Algorithm Based on Decomposition*. MOEA/D decomposes a multiobjective optimization problem into a number of scalar optimization subproblems and optimizes them simultaneously. Each subproblem is optimized by only using information from its several neighboring subproblems, which makes MOEA/D with lower computational complexity at each generation.

- Given input:
  α (population size), N (maximum number of generations) and κ (fitness scaling factor)
- Required output:
  A (Pareto set approximation)

Step1: Initialization:

Generate an initial population P of size α; set the generation counter m to 0.

Step2: Fitness assignment:

First scale objective and indicator values, and then use scaled values to assign fitness values.

Determine for each objective f_i its lower bound $\underline{b_i} = \min_{x \in P} f_i(x)$ and upper bound $\overline{b_i} = \max_{x \in P} f_i(x)$

Scale each objective to the interval [0, 1], i, e, $f_i^{'}(x) = \left(f_i(x) - \underline{b_i}\right) / \left(\overline{b_i} - \underline{b_i}\right)$

Calculate indicator values $I(x^1, x^2)$ using the scaled objective values $f_i^{'}$, instead of the original f_i, and determine the maximum absolute indicator value $c = \max_{x^1, x^2 \in P} \left| I(x^1, x^2) \right|$

For all $x^1 \in P$ set $F(x^1) = \sum_{x^2 \in P \setminus \{x^1\}} -e^{-I(\{x^2\}, \{x^1\})/(c \cdot \kappa)}$

Step3: Environmental selection:

Iterate the following three steps until the size of population P does not exceed α:

Choose an individual $x^* \in P$ with the smallest fitness value, i.e. $f(x^*) \le f(x)$ for all $x \in P$.

Remove x^* from the population.

Update the fitness values of the remaining individuals, i.e. $F(x) = F(x) + e^{-I(\{x^*\}, \{x\})/(c \cdot \kappa)}$

Step4: Termination:

If $t \ge T$ or another stopping criterion is satisfied then set A to the set of decision vectors represented by the nondominated individuals in P.

Step5: Mating selection:

Perform binary tournament selection with replacement on P in order to fill the mating pool.

Step6: Variation:

Apply recombination and mutation operators to the mating pool and add the resulting population to P. Increment generation counter ($t = t + 1$) and go to Fitness assignment step.

Table 2.3 The main structure of IBEA

Table 2.4 presents the input and output of the algorithm, and the process of the algorithm to deal with multiobjective problem.

- Input:
  a stopping criterion; the number of the subproblems considered in MOEA/D; a uniform spread of N weight vectors: $\lambda^1, \ldots, \lambda^N$; and T, the number of the weight vectors in the neighborhood of each weight vector.
- Output: External Population (*EP*)

Step 1: Initialization:

Set $EP = \varnothing$. Compute the Euclidean distances between any two weight vectors and then work out the closest weight vectors to each weight vector. For each $i = 1, \ldots, N$, set $B(i) = \{i_1, \ldots, i_T\}$, where $\lambda^1, \ldots, \lambda^{i_T}$ are the T closest weight vectors to λ^i.

Generate an initial population $x^1, \ldots, x^N$ randomly or by a problem-specific method. Set $FV^i = F\left(x^i\right)$

Initialize $z = \left(z_1, \ldots, z_m\right)^T$ by a problem-specific method.

Step 2: Update

For $i = 1, \ldots, N$, do

Reproduction: Randomly select two indexes k, l from $B(i)$, and then generate a new solution y from x^k and x^l by using genetic operators.

Improvement: Apply a problem-specific repair/ improvement heuristic on y to produce $y^{'}$

Update of z^*: for each $j = 1, \ldots, m$, if $z_j < f_j\left(y^{'}\right)$, then set $z_j = f_j\left(y^{'}\right)$.

Update of Neighboring Solutions: if $g^{te}\left(y' \lambda^j, z\right) \le g^{te}\left(x^j \middle| \lambda^j, z\right)$, for each index $j \in B(i)$, set $x^j = y^{'}$ and $FV^j = F\left(y^{'}\right)$. $g^{te}\left(x^j \middle| \lambda^j, z^*\right) = \max_{1 \le i \le m}\left\{\lambda_i^j f_i(x) - z_i^*\right\}$, $\lambda^1, \ldots, \lambda^N$ be a set of even spread weight vectors and z^* be the reference point.

Update of *EP*: Remove *EP* from all the vectors dominated by $F\left(y^{'}\right)$ and add $F\left(y^{'}\right)$ to *EP* if no vectors in *EP* dominate $F\left(y^{'}\right)$.

Step 3: Stopping Criteria

If stopping criteria is satisfied, then stop and output. Otherwise, go to Step 2.

Table 2.4 The main structure of MOEA/D

2.3 Performance Metrics

In this part, first some important concepts are introduced. These are applied to define relations between different approximation fronts. Then, we will explain both unary and binary performance metrics in detail.

2.3.1 Outperformance Relations

Hansen and Jaszkiewicz [9] have focused on the problem of evaluating approximations to the true Pareto front. They define a number of outperformance relations that classify the relationships between two sets of nondominated objective individuals while the preferences information of the test problem is unknown. Based on these, Knowles and Corne [18] further give two definitions about Monotony and Relativity.

- Weak Outperformance [9]

  A weakly outperforms B ($A\,O_W\,B$): A weakly outperforms B if and only if nondominated points of $A \cup B$ are the same as the whole points in A, and $A \neq B$. Therefore, each point $z_2 \in B$ is covered by a point $z_1 \in A$; 'cover' means is equal to or dominates z_2. Additional, there is at least one point $z_1 \in A$ which is not contained in B. Adding to B a new non-dominated individual can generate a new approximation front that weakly outperform B.

- Strong Outperformance [9]

  A strongly outperforms B ($A\,O_S\,B$): A strongly outperforms B if and only if nondominated points of $A \cup B$ are the same as the whole points in A, and B contains another dominated points. Therefore, each point $z_2 \in B$ is covered by a point $z_1 \in A$. Additional, there is at least one point $z_2 \in B$ that is dominated by a point $z_1 \in A$ and is not contained in B. Adding to B a new individual that dominates at least one point in B can generate a new approximation front that strongly outperform B.

- Complete Outperformance [9]

A completely outperforms B ($A O_C B$): A completely outperforms B if and only if nondominated points of $A \cup B$ are the same as the whole points in A, and none of nondominated points of $A \cup B$ belongs to B. Therefore, each point $z_2 \in B$ is dominated by a point $z_1 \in A$. Adding to B a new individual that dominates all points in B can generate a new approximation front that completely outperform B.

- Incomparable Outperformance [9]

  A incomparable outperforms B is defined as some points in A dominate points in B and some points in B dominate points in A. In this condition, we cannot state that which one is better.

- Compatibility and Weak compatibility [9]

  Let $O = O_W, O_S$ or O_C:

  Compatibility: A comparison metric R is compatible with an outperformance relation O if for each pair of nondominated sets A and B, such that $A O B$, R will evaluate A as being better than B.

  Weak compatibility: A comparison metric R is compatible with an outperformance relation O if for each pair of nondominated sets A and B, such that $A O B$, R will evaluate A as being no worse than B.

- Monotony and Weak Monotony [18]

  Monotony: Given a nondominated set A, adding a non-dominated point improves its evaluation. Compatibility with O_W is necessary and sufficient for ensuring monotony [18].

  Weak Monotony: Given a nondominated set A, adding a non-dominated point does not degrade its evaluation.

- Relativity and Weak Relativity [18]

  Relativity: The evaluation of Z^* is uniquely optimal, i.e., all other nondominated sets have a strictly inferior evaluation. Compatibility with O_W is sufficient but not necessary for ensuring relativity [18].

  Weak Relativity: The evaluation of Z^* is non-uniquely optimal, i.e., all other nondominated sets have a non-superior evaluation.

From above theories, we can find $AO_C\,B \Rightarrow AO_S\,B \Rightarrow AO_W\,B$ and $O_C \subset O_S \subset O_W$. Complete Outperformance is the strongest of the relation and easiest to be compatible; Weak Outperformance is the weakest of the relation and most difficult to be compatible.

2.3.2 Compatibility and Completeness

Zitzler [10] has proposed a method that links Comparison Methods and Dominance Relations to reveal differences in performance between MOEAs, and make the statement that an algorithm outperforms another one. What conclusions can be drawn with respect to the dominance relations is emphasized.

Quality Indicator:

In order to quantify quality differences between approximation sets, quality measures are necessary used to map approximation sets to the real numbers by applying common metrics to the resulting real numbers. Based on this observation, Zitzler defines what a quality measure is:

An m -ary quality indicator I is a function $I:\Omega^m \to R$, which assigns each vector $(A_1, A_2, \ldots, A_m)$ of m approximation sets a real value, $I(A_1, A_2, \ldots, A_m)$.

Often, not a single indicator but rather a combination of different quality indicators is used in order to assess approximation sets. The combination quality indicator vector can be regarded as a function that assigns each approximation set a vector of multiple real numbers.

Comparison Method:

Here, we use Pseudo-Boolean function E to compare different approximation sets. It maps vectors of real numbers to Booleans:

$E(I) := (I = 1)$ means E is true if and only if $I = 0$.

A comparison method $C_{I,E}$ is based on a combination of one or more quality indicators I and a Boolean function E .

Given two approximation sets $A, B \in \Omega$, $I = (I_1, I_2, \ldots, I_k)$ a combination of quality indicators, and $E : IR^k \times IR^k \to \{\text{false,true}\}$ a Boolean function, from Zitzler's theory, If $C_{I,E}$ is defined by combination of unary indicators and Boolean function E , $C_{I,E}(A, B) = E(I(A), I(B))$; If $C_{I,E}$ is defined by combination of binary indicators and Boolean function E , $C_{I,E}(A, B) = E(I(A, B), I(B, A))$

Compatibility and Completeness:

Compatibility: for any $A, B \in \Omega$, the result of $C_{I,E}(A, B)$ can indicate that A is better than B , or B is better than A .

Completeness: for any $A, B \in \Omega$, the relation that A is better than B , or B is better than A can decide the value of $C_{I,E}(A, B)$.

For a particular quality indicator, if there exists a Boolean function such that the resulting comparison method is compatible and in addition complete with respect to the various dominance relations (strong, weak or complete), this quality indicator can be evaluated to be powerful quality indicator.

However, Zitzler [10] has proven there exists no comparison method based on a finite combination of unary quality indicators that is compatible and complete at the same time. It means for any combination of a finite number of unary quality indicators, a Boolean function cannot be found such that this comparison method is both compatible and complete.

From the above reasons, the number of performance metrics that determine a better approximation set from two sets is infinite. Furthermore, to be able to detect whether an approximation front weakly dominates or dominates another approximation front, the number of performance metrics should be greater than or equal to the number of objectives.

2.3.3 Unary performance metrics

The first type of performance metrics concerns about assessing the Number of Pareto Optimal Solutions in the Set. *Ratio of Non-dominated Individuals* (RNI) [5] gives the proportion of the useful solutions known as the Pareto-front in a given population size; *Error Ratio* (ER) [4] evaluates the proportion of non true Pareto points in the approximation front; *Overall Nondominated Vector Generation and Ratio* (ONVG) [4] counts the number of distinct non-dominated points generated; and *Pareto Dominance Indicator* (NR) [3] measures the ratio of non-dominated solutions contributed by a particular solution set to the non-dominated solutions provided by all solution sets.

Ratio of Non-dominated Individuals (*RNI*) [5]

The performance measure is:

$$RNI(X) = \frac{|nondom\_indiv|}{P} \quad (2\text{-}18)$$

$nondom\_indiv$: Non-dominated individuals in population X with size P.

If $RNI = 1$, all the individuals in X are non-dominated; and if $RNI = 0$, none of the individuals in X is non-dominated. It is always required to have enough qualified individuals to construct a Pareto front. Therefore, RNI is significant in that it checks the proportion of non-dominated individuals in population X.

Knowles and Corne in [18] have stated that: RNI is not weakly compatible with any outperformance relation. It exhibits monotony. Clearly, add a non-dominated point will make the RNI value better. It violates relativity. True Pareto front cannot be sure to have more numbers of non-dominated points than other approximation fronts.

Error Ratio (*ER*) [4]

It is defined as the proportion of non true Pareto points in reference set:

$$ER(X) = \frac{\left(\sum_{i=1}^{n} e_i\right)}{P} \quad (2\text{-}19)$$

Individual i is a point in approximation front X. P is the number of individuals in X.

$e_i = 0$ means individual i is in true Pareto set; and $e_i = 1$ means individual i is not in true Pareto set. Lower value of ER implies a small proportion of non true Pareto points in X and represents better nondominated sets. It is a reference metric using true Pareto front as reference set.

Knowles and Corne in [18] have stated that: ER is only weakly compatible with O_C. It is not weakly compatible with O_S or O_W. If one algorithm generates 100 points, one in the Pareto front, the other 99 points are very far from the Pareto front; its error ratio is 0.99. However, if another algorithm also generates 100 points, all these 100 points are very close to Pareto front, its error ratio is 1. Although the second algorithm's error ratio is larger than the first one, we can see clearly the second one is better than the first one. *ER* strongly violates monotony: Add a nondominated but non-Pareto optimal points in an approximation set, makes the ER score worse. The advantage is easy to understand and easy to calculate. It is scaling independent. The disadvantage is the true Pareto front information is needed. It is incompatible with the outperformance relations.

Overall Nondominated Vector Generation and Ratio *(ONVG)* [4]

It measures the total number of nondominated vectors found in approximation front during MOEA execution. It is defined as:

$$ONVG = |PF_{known}| \tag{2-20}$$

PF_{known} represents approximation front. From [19], too few vectors in PF_{known} make the front's representation poor and too many vectors may overwhelm the decision maker.

Knowles and Corne in [18] have stated that: *ONVG* is not weakly compatible with any outperformance relation. It does not exhibit either weak monotony or weak relativity. The advantage is easy to calculate and scaling independent while the disadvantage is A outperformance B on this metric does not mean A is clearly better than B.

Pareto Dominance Indicator (*NR*) [3]

Considering the different PFs, $A_1, A_2, \ldots A_n$ evolved by algorithms, this metric measures the ratio of nondominated solutions that is contributed by a particular solution set A_i to the nondominated solutions provided by all solution:

$$NR(A_1, A_2, ..., A_n) = \frac{|B \cap A_1|}{B} \quad (2\text{-}21\ A)$$

$$B = \{b_i \mid \forall b_i, \neg\exists a_j \in (A_1 \cup A_2 ... \cup A_n) \prec b_i\} \quad (2\text{-}21\ B)$$

A_1 is the solution set under evaluation.

Zitzler show that no combinations of unary performance metrics can provide a clear indication of whether an evolved set is better than another in the Pareto dominance sense, this metric can be a complement to another metrics [1]. It is weak compatible with O_S and O_C .

The second type of performance metrics focuses on measuring the closeness of the solution to the True Pareto Front. *Generational Distance* (GD) [4] measures how far the evolved solution set is from the rue Pareto front; and *Maximum Pareto Front Error* (MPFE) [4] identifies the largest distance between the point in the theoretical Pareto front and the point in the approximation front.

Final Generational Distance (*GD*) [4]

$$GD = \frac{\left(\sum_{i=1}^{n} d_i^p\right)^{1/p}}{n} \quad (2\text{-}22)$$

n is the number of vectors in the approximation front, d_i is the distance in objective space between individual i and the nearest member of PF_{true} .This metric is a value representing how “far” the approximation front is from true Pareto front. Lower value of *GD* represents better performance. It measure general process towards true Pareto front [18]. It is a reference metric using true Pareto front as reference set.

Knowles and Corne in [18] have proven that *GD* is not weakly compatible with O_W , but is compatible with O_S . It violates weak monotony which implies adding a non-dominated point to an approximation fronts cannot improve its *GD* value. It exhibits weak relativity since any subset of Pareto front has an optimal *GD*. For a constant size of non-dominated set, *GD* is compatible with O_S. But it is not used for non-dominated sets that are changing in cardinality. It cannot be reliably differentiate between different levels of complete outperformance. The true Pareto front information is needed.

Maximum Pareto Front Error (*MPFE*) [4]

It measures the largest distance between any vector in the approximation front and the corresponding closest vector in true Pareto front. It finds the largest distance between individual j in approximation front and individual i in the true Pareto optimal front.

$$MPFE = \max_j \left(\min_i \left| f_1^i(x) - f_1^j(x) \right|^p + \left| f_2^i(x) - f_2^j(x) \right|^p \right)^{1/p} \tag{2-23}$$

It is a reference metric using true Pareto front as reference set. In terms of Pareto compatibility, it is not weakly compatible with outperformance relation. It violates weak monotony. It exhibit weak relativity since any subset of Pareto front is optimal. It is cheap to compute. It helps us to focus on how far the worst point is. However, from [19], for a non-dominated set, a good performance in *MPFE* does not ensure it is better than another one with a much worse *MPFE*. The true Pareto front information is needed.

The third type of performance metrics measures distribution of the Solutions. *Uniform Distribution* (UD) [5] measures the distribution of an approximation front under a pre-defined parameter σ_{share}; *Spacing* [6] measures how evenly the evolved solutions distribute itself; and *Number of Distinct Choices* (NDC_u) [7] identifies solutions that are sufficiently distinct for a special value u.

Uniform Distribution (UD) [5]

It measures the distribution of non-dominated individuals on the found trade-off surface. For a given set of nondominated individuals $X^{'}$ in a population X :

$$UD(X^{'}) = \frac{1}{1+S_{nc}} \tag{2-24 A}$$

$$S_{nc} = \sqrt{\frac{\sum_{i}^{N_{x^{'}}} \left(nc\left(x_i^{'}\right) - \bar{nc}(X^{'}) \right)}{N_{x^{'}} - 1}} \tag{2-24 B}$$

$$nc\left(x_i^{'}\right) = \sum_{j, j \neq i}^{N_{x^{'}}} f(i, j) \tag{2-24 C}$$

$$f\left(i, j\right) = \begin{cases} 1, dis(i, j) \langle \delta_{share} \\ 0. else \end{cases} \tag{2-24 D}$$

$$\bar{nc}\left(X^{'}\right) = \frac{\sum_{i}^{N_{x^{'}}} nc\left(x_i^{'}\right)}{N_{x^{'}}} \tag{2-24 E}$$

σ_{share} is pre-defined by decision maker. S_{nc} is the standard deviation of niche count of the overall set of non-dominated individuals. $N_x^{'}$ is the size of the set $x^{'}$. $nc\left(x_i^{'}\right)$ is the niche count of the i^{th} individual $x^{'}$ and $dis\left(i, j\right)$ is the distance between individual i and j in the objective domain.

Knowles and Corne in [18] have proven that: UD is not even weakly compatible with O_W. It violates monotony in that an additional point cannot make sure the distribution is improved. It violates relatively. An approximation front far from the Pareto front can have the same *UD* score to the Pareto front. It has low computational overhead and provides the opportunity for Decision maker to choose well distributed front according to real application need by assigning different value to σ_{share}. However, it is difficult to choose σ_{share} without any reference information.

Spacing [6]

This metric is a value measuring the distribution of vectors throughout approximation front. It provides information about the distribution of vectors obtained. Because approximation front's "beginning" and "end" are known, a suitably defined metric judge how well PF_{known} is distributed.

$$S = \sqrt{\frac{1}{n-1}\sum_{i+1}^{n}\left(\bar{d} - d_i\right)^2} \quad \text{(2-25 A)}$$

$$d_i = \min_j \left(\left| f_1^i(x) - f_1^j(x) \right|^2 + \left| f_2^i(x) - f_2^j(x) \right|^2 \right)^{1/2} \quad \text{(2-25 B)}$$

d_i is minimum distance between two solutions in the approximation front. Knowles and Corne in [18] have stated that: Spacing is not even weakly compatible with O_W. It violates monotony in that an additional point cannot make sure the distribution is improved. It violates relatively. An approximation front far from the Pareto front can have the same Spacing score to the Pareto front. It has low computational overhead and can be generalized to more than two dimensions by extending the definition of d_i. But the use of normalized distances may be problematic.

Number of Distinct Choices (NDC_u) [7]

In this metric, only those solutions that are sufficiently distinct from one another should be accounted for as useful design options. The quality $NT_u(q,P)$ indicates whether or not there is any point $p_k \in P$ that falls into the region.

$T_u(q)$ is decided by $1/u^m$, $0 < u < 1$.m is the dimension of objective-space

$$NT_u(q,P) = \begin{cases} 1, \exists p_k \in P, p_k \in T_u(q) \\ 0, \forall p_k \in P, p_k \notin T_u(q) \end{cases} \quad \text{(2-26 A)}$$

$NDC_u(P)$ is the number of distinct choices for a pre-specified value of u:

$$NDC_u(P) = \sum_{l_m=0}^{v-1} \cdots \sum_{l_2=0}^{v-1} \sum_{l_1=0}^{v-1} NT_u(q, P) \quad \text{(2-26 B)}$$

From [7], for a pre-specified value of u, an observed Pareto solution set with a higher value of the quantity $NDC_u(P)$ is preferred to a set with a lower value.

In terms of Pareto compatibility: NDC_u is not even weakly compatible with O_W. It violates monotony in that an additional point cannot make sure the distribution is improved. It violates relatively. An approximation front far from the Pareto front can have the same Spacing score to the Pareto front. It provides the opportunity for Decision maker to choose well distributed front according to real application need by a pre-specified value of u. However, it is not easy to compute.

The fourth category of performance metrics concerns spread of the solutions. *Maximum Spread* (MS) [3] measures how well the rue Pareto front is covered by the approximation set.

Maximum Spread (*MS*) [3]

It addresses the range of objective function values and takes into account the proximity to PF_{true}. This metric is applied to measure how well the PF_{true} is covered by the PF_{known}.

$$MS = \sqrt{\frac{1}{M} \sum_{i=1}^{M} \left[\frac{\min\left(f_i^{\max}, F_i^{\max}\right) - \max\left(f_i^{\min}, F_i^{\min}\right)}{\left(F_i^{\max} - F_i^{\min}\right)} \right]^2} \quad \text{(2-27)}$$

$f_i^{\max}$ and $f_i^{\min}$ are the maximum and minimum of the i th objective in PF_{known}, respectively; $F_i^{\max}$ and $F_i^{\min}$ are the maximum and minimum of the i th objective in PF_{true}, respectively. If $MS(A) > MS(B)$, the solution A is preferred to B.

The last type of performance metrics considers both closeness and diversity:

Hypervolume Metric [11]

The hyperarea difference metric [11] is also called *S*-metric and can be used to quantitatively evaluate the difference between the size of the objective space dominated by an observed Pareto front and that of the space dominated by the true Pareto front.

The true Pareto front set dominates the largest volume in solution space while an approximation front may only dominate a portion of true Pareto front dominated volume. A quantitative measure is obtained as to how much worse an observed Pareto front is when compared to the true Pareto front. Although in real problem, the true Pareto front is usually unknown; it should still be possible to compare an approximation front to another so as to draw a conclusion that which front is better.

Let $HD(P)$ represent the hypervolume difference quantity between the inferior regions of the true Pareto solution set P_t and *the* inferior region of the observed Pareto solution set P [7]:

$$HD(P) = space\left(S_{in}(P_t) - S_{in}(P)\right) = 1 - space\left(S_{in}(p)\right) \tag{2-28}$$

In [4], Veldhuizen also propose a Hyperarea Ratio metric defined as:

$$HR = \frac{H_1}{H_2} \tag{2-29}$$

H_1 is the hyperarea of PF_{known} while H_2 is that of PF_{true}. In the proposed performance metrics ensemble to be presented in Chapter 3, we adopt this modified Hyperarea Ratio Metric.

It is compatible with all the outperformance relations [18]. Each algorithm can be assessed independently of the other algorithms in this metrics. Hypervolume Metric differentiates between different degrees of complete outperformance of two sets, so it can evaluate how much better an approximation set is than the other approximation set. It is scaling independent, non-cardinal and its meaning is intuitive. It can be misleading if the Pareto optimal front is non-convex [3].

Therefore, it focuses on the volume of the objective space dominated by an approximation set and calculates the hypervolume of the multi-dimensional region enclosed by approximation front and a 'reference point' [18].

A reference point is also called the upper boundary of the region. The choice of the upper boundary determines which approximation front has the maximum dominated hypervolume. In many cases, we need to find the reference point to represent the upper boundary of the region. The reference point should be chosen so that it is dominated by all Pareto-optimal solutions.

Auger and Zitzler [20] have proposed a method to define the reference point $r = (r_1, r_2)$ for a specific problem:

Let u be an integer larger than or equal to 2. Assume that f is continuous on $[x_{\min}, x_{\max}]$, non-increasing, differentiable on $[x_{\min}, x_{\max}]$ and that f' is continuous on $[x_{\min}, x_{\max}]$:

The leftmost extremal point:

If $\lim_{x \to x_{\min}} -f(x) < +\infty$:

$$R_2 := \sup_{x \in [x_{\min}, x_{\max}]} \left\{ f'(x)(x - x_{\max}) + f(x) \right\} \tag{2-30}$$

When R_2 is finite, the leftmost extremal point is contained in optimal u-distributions if the reference point $r = (r_1, r_2)$ is such that r_2 is strictly larger than R_2. r_2 can be chosen to be R_2.

If $\lim_{x \to x_{\min}} -f(x) = +\infty$, the left extremal point of the front is never included in optimal *u*-distributions. So, $r_2 = +\infty$.

The rightmost extremal point:

when f is strictly negative on $[x_{\min}, x_{\max}]$:

$$R_1 := \sup_{x \in [x_{\min}, x_{\max}]} \left\{ x + \frac{f(x) - f(x_{\min})}{f'(x)} \right\} \tag{2-31}$$

When R_1 is finite, the rightmost extremal point is contained in optimal u-distributions if the reference point $r = (r_1, r_2)$ is such that r1 is strictly larger than R_1. r_1 can be chosen to be R_1.

If $f(x_{\max}) = 0$, the right extremal point of the front is never included in optimal u-distributions. So, $r_1 = +\infty$.

Hypervolume metric also has a large computational overhead. Largest computation is defined [21] by Klee's Measure Problem (KMP) and lowest computation is defined by the UniformGap Problem. KMP is the problem of computing the length of the union of a collection of intervals on the real line and defines the upper boundary of the computation. It can be solved with computational complexity in optimal $O(n \log n)$ time.

First, measure of a union of hyper-rectangles in d dimensions: $O(n^{d-1} \log n) \Rightarrow O(n^{d-1}) \Rightarrow O(n^{d/2} \log n)$. Second, the weakly dominated hypervolume for a point set $P \subseteq IR^d_{\geq 0}$ as a special case of Klee's measure problem: The polytope Π^d is patterned by the collection of hyper-rectangles $\{R_p\}_{p \in P}$ with $R_p := \{x \in IR^d_{\geq 0} : x \leq p\}$ spanned by the points in P and the reference point $r = 0 \in IR^d_{\geq 0}$. Third, the set of hyper-rectangles is the input and the desired hypervolume output. Finally, we get an upper bound of computation time in $O(n \log n + n^{d/2} \log n)$. The best upper bound currently known for $d > 3 : O(n \log n + n^{d/2})$

The UniformGap Problem defines the lower boundary of the computation. This problem uses the fixed-degree algebraic decision tree, which is the standard model used in computational geometry and is used to prove lower bounds for (geometric) decision problems.

It captures the behavior of a (loop-unrolled) algorithm that branches depending on the outcome of evaluations of bounded-degree polynomials. A lower bound on the complexity of a given problem can then be derived by establishing a lower bound on the depth of any such tree resembling any valid algorithm to solve this problem. Moreover, linear-time reduction from problem A to problem B means the lower bound for problem A is a lower bound for problem B

.

2.3.4 Binary performance metrics

The first type of binary performance metrics based on unary quality indicator. It includes ε-*indicator* I_{ε}, *enclosing hypercube Indicator* and *coverage difference metrics* (*D*-metric).

First, ε-*indicator* I_{ε} [10] can be used to compare algorithms directly without reference front information. This is defined as:

$$I_{\in}\left(A,B\right)=\inf_{\varepsilon\in R}\left\{\forall z^{2}\in B\exists z^{1}\in A:z^{1}\geq_{\varepsilon}z^{2}\right\} \quad \text{(2-32 A)}$$

$$z^{1}\geq_{\varepsilon}z^{2}\Leftrightarrow\forall 1\leq i\leq n:z_{i}^{1}\leq\varepsilon\cdot z_{i}^{2} \quad \text{(2-32 B)}$$

$I_{\in}\left(A,B\right)$ reflects the value of ε. For every individual z^{2} in B, there must exist an individual z^{1} in A dominate $\varepsilon\cdot z^{2}$.

For any pair $A,B\in\Omega$, $A \text{ f f } B\Rightarrow I_{\in}\left(A,B\right)<1$ .Therefore, if A is better than B, $\varepsilon<1$

However, $I_{\in}(A,B) < 1$ can only imply that A is not worse than B in strictly dominates and better relations.

Second, *Enclosing Hypercube Indicator* [10] is defined as:

$$I_1^{HC}(A) = \sup_{a \in R}\{\{(a,a,...,a)\} > A\} \quad \text{(2-33 A)}$$

$$I_2^{HC}(A) = \inf_{b \in R}\{\{(b,b,...,b)\} < A\} \quad \text{(2-33 B)}$$

$$I_2^{HC}(A) < I_1^{HC}(B) \Rightarrow A > B \quad \forall 1 \le i \le n+1 \quad \text{(2-33 C)}$$

$I_2^{HC}(A)$ is the point that worse than all individuals in A. $I_1^{HC}(B)$ is the point that better than all individuals in B.

Finally, *coverage difference metrics* (*D*-metric) [11] is defined as the size of the space dominated by A and not dominated by B (regarding the objective space), $A,B \subseteq X$ be two sets of decision vectors.

$$D(A,B) = S(A+B) - S(B) \quad \text{(2-34 A)}$$

where $S(A)$ is the Hypervolume Difference Metric (*S*-metric).

Zitzler [11] suggest that (ideally) the *D* metric is used in combination with the *S* metric where the values may be normalized by a reference volume V, where (for a maximization problem) V is given by:

$$V = \prod_{i=1}^{k}\left(f_i^{\max} - f_i^{\min}\right) \quad \text{(2-34 B)}$$

f_i^{max} and f_i^{min} represent the maximum respectively minimum value for the objective f_i. Thus, the value $D_1(A,B) = \frac{D(A,B)}{V}$ represents the relative size of the region (in the objective space) dominated by A and not dominated by B.

The second type is direct comparison binary metrics: C metrics and R metrics.

C metrics [9] maps the ordered pair (A,B) to the interval $[0,1]$:

$$C(A,B) = \frac{\left|\{b \in B \mid \exists a \in A : a \le b\}\right|}{|B|} \tag{2-35}$$

$C(A,B)=1$: all decision vectors in B are weakly dominated by A; $C(A,B)=0$: none of the points in B are weakly dominated by A. It is compatible with O_S and O_C, but incompatible with O_W. Only if $C(A,B)=1$ and $C(B,A)<1$, it is compatible with O_W. It is Non-symmetric: $C(A,B)$ is not necessarily equal to $-C(B,A)$. It has low computational overhead. Scale and reference point independent. However, there are situations when the metric C cannot decide if an obtained front is better than the other.

R metrics [5] consist of three sub-metrics: $R1(A,B,U,p)$, $R2(A,B,U,p)$ and $R3(A,B,U,p)$.

$R1(A,B,U,p)$ calculates the probability that approximation A is better than approximation B over an entire set of utility functions.

$$R1(A,B,U,p) = \int_{u \in U} C(A,B,u)\, p(u)\, du \tag{2-36 A}$$

$$C(A,B,u) = \begin{cases} 1, u^*(A) > u^*(B) \\ 1/2, u^*(A) = u^*(B) \\ 0, u^*(A) < u^*(B) \end{cases} \tag{2-36 B}$$

$u^*(A) = \max_{z \in A}\{u(z)\}$, $u^*(B) = \max_{z \in B}\{u(z)\}$ and $p(u)$ is an intensity function expressing the probability density of the utility. If $R1(A,B,U,p) > 0.5$, A is the winner; if $R1(A,B,U,p) < 0.5$, B is the winner. In terms of Pareto compatibility: $R1$ is only weakly compatible with O_W and is not compatible with O_C. It has low computational overhead and Scale independent but $R1$ is cycle-inducing.

$R2(A,B,U,p)$ calculates the expected difference in the utility of an approximation A with another one B.

$$R2(A,B,U,p) = \int_{u \in U} \left(u^*(A) - u^*(B)\right) p(u)\, du \tag{2-37}$$

$R2$ is compatible with O_W. It can differentiate between different levels of complete outperformance. However, each utility function in U must be appropriately scaled with respect to the others and its relative importance.

$R3(A,B,U,p)$ calculates the ratio of the best utility values. That is the expected proportion of superiority.

$$R3(A,B,U,p) = \int_{u \in U} \frac{\left(u^*(A) - u^*(B)\right)}{u^*(A)} p(u)\, du \tag{2-38}$$

CHAPTER THREE

METHODOLOGY

This chapter first explains the motivation of the designed performance metrics ensemble. Then, we describe the proposed approach in detail.

3.1 MOTIVATION

Chapter 2 has introduced a large number of available performance metrics with different characteristics. However, none of these metrics alone can faithfully measure MOEA performance independently. Every metric can provide some specific but limited information and can only be used effectively in some specified conditions. For example, *UD* does a poor job when the Pareto front is discontinued and Hypervolume can be misleading if the Pareto optimal front is non-convex [4]. This means one metric cannot entirely evaluate MOEAs in all conditions. Every metric focuses on some special characteristics while neglects information in others. Also, every metric has its unique character; no metrics can substitute others completely. Therefore, a single metrics cannot provide a comprehensive measure for MOEAs. Moreover, because reduce objective space must losing information, a fixed number of indicators are not sufficient to make a comprehensive measure for MOEAs [11].

Meanwhile, different metrics perform differently in different test problems. For a given MOEA, one metric may do well in one test problem, however, in other test problems, it may be misleading. For a specific test problem, we cannot know which metric is better. We need to try various metric to identify which one is the best. This is a heavy computational process.

To overcome these challenge and arrive at a faithful evaluation of a given MOEAs, performance metrics ensemble is necessary. Ensemble method uses multiple metrics to obtain a fair performance than what could be obtained from any of single performance metric alone. Ensemble metrics not only can give the comprehensive comparison between different algorithms, but avoid the choosing process and can be directly used to assessing MOEAs.

3.2 OVERVIEW OF PERFORMANCE METRICS ENSEMBLE

The proposed framework is shown in Figure 3.1

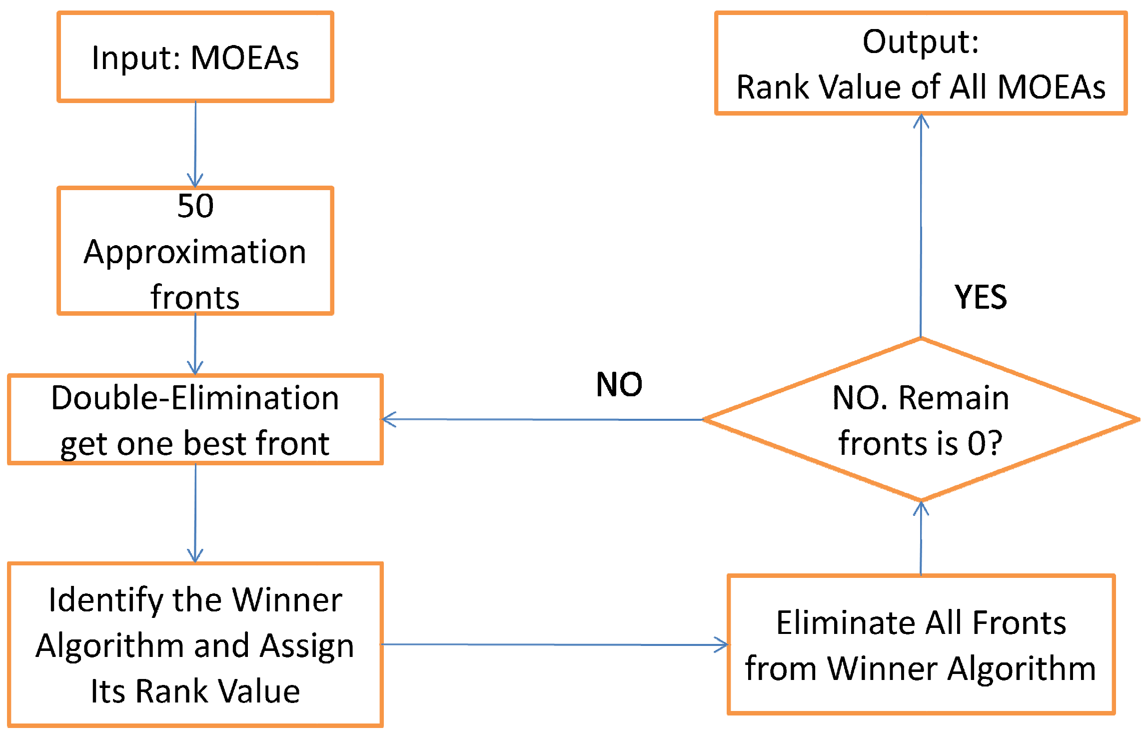


Fig 3.1 The proposed framework

Table 3.1 explains the whole process of ensemble method in detail. 50 independent trials given the same initial populations to each and every candidate MOEAs are performed, resulting 50 approximation fronts from each chosen MOEA for comparison. Then, in the proposed Double-Tournament Selection, every individual (i.e., approximation front) has two opportunities to compete. After one winner is found, identify which algorithm it is from; remove all the fronts of that algorithm and compare others again. Finally, all the algorithms will be assigned a rank value.

Input: A number of MOEAs for comparison

Output: Rank Value of the chosen MOEAs

Step 1: Generate 50 approximation fronts from each MOEA:
The selected MOEAs run 50 times for a given benchmark problem under the same initial conditions. Every time, each MOEA generate an approximation fronts while each front represents its algorithm. After that, a single performance metric in the performance metrics ensemble is randomly chosen to measure the quality of each front, and the best approximation front is picked up based on that performance pertained to the chosen benchmark problem at hand. After 50 running times, we get 50 approximation fronts.

Step 2: Use Double-Tournament Selection to get the best one individual:

i. Every 2 individuals are randomly picked to form competition pair. One performance metric is randomly chosen in each competition. Each pair will generate a winner and a loser. After all the competition, two parts are created: W_1 contains all the winners named winner bracket; the other L_1 contains all the losers named loser bracket.

ii. In both W_1 and L_1, the same competition is performed again and each package can be divided into two sub-packages. One sub-package contains winners, the other losers. W_1 is divided into W_{11} (winners) and W_{12} (losers); L_1 is divided into L_{11} (winners) and L_{12} (losers). Then, individual of W_{12} will compete with individual of L_{11} one by one. Winners from these competitions consist of a new loser bracket L_{13}. We reserve W_{11} and L_{13}.Therefore, the population is reduced to an half.

iii. In both W_{11} and L_{13}, do as Step ii again. Every time, reduce the population by half. Finally, only one individual wins at the very end.

Step 3: Assign every MOEA a rank value
Identify from which MOEA this winner front comes from. Assign this algorithm rank value 1. Then, eliminate all the approximate fronts generated by this algorithm in the 50 approximate fronts. Go back to step 2 and compare remaining fronts from all MOEA (less the winner with rank 1) again. Finally, we will assign each algorithm a rank value implying its ranking order through the proposed performance metrics ensemble.

Table 3.1 The Whole Process of Ensemble Method

3.3 ENSEMBLE METHOD WITH DOUBLE-TOURNAMENT SELECTION

3.3.1 Double-Tournament Selection

The Modified Double Tournament algorithm selects an individual using tournament selection: at the initial step, the tournament contestants are chosen at random from the population. Then, at the following step, they are each the winners of last tournament selection.

For example, imagine if the tournament has a pool size of 32: first, the 16 "qualifier" tournaments are held as normal in tournament selection, and the whole pool is divided into two parts: winner bracket contains 16 winners and loser part 16 losers. Then, in each of the part, 8 normal tournament selections are processed so that the part is divided again. In both parts, there are 8 new winners and 8 new losers. Afterward, we compete 8 winners from loser bracket with 8 losers from winner bracket. Here, two individuals come into one tournament selection should be from different part, that means, one individual from winner bracket is compared with one individual from loser bracket. Therefore, we get 8 winners from this step. This process reduces the total number of individuals from 32 to 16. Continue to do it, $16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$, finally, we obtain the ultimate winner. This ultimate winner defeats all other 31 competitors.

The motivation for applying Double-Tournament Selection is that it gives every individual two chances to take part in the competition. This advantage is helpful to reserve good individual. Because of the stochastic process, one approximation front from a quality MOEA may lose the competition at time when a metric measuring the very deficient aspect of problem characteristics is applied. If this happens in the single elimination tournament, the front will be lost forever and it would not have any chance to compete again. However, in the Double-Tournament Selection, even it loses once, it has an opportunity to compete again and hopefully win at all.

For example, in NCAA basketball tournament, the last year's champion team will versus the winner of the 64^{th} and 65^{th} team. Of course, the probability that the champion team wins is very large, but basketball game bears a huge number of uncertain factors, there exists probability that the 64^{th} team wins. In this condition, if single elimination is used, the last year's champion team loses at all. We will not see the excellent performance of this team in the following games of this year. This will be a great loss for audience! In Double-Tournament Selection, the champion team will have another chance to compete in the loser bracket; it can make up its mistake in the last game and win again.

Therefore, in MOEAs comparison, if a really good front loses its game at one competition, it has the opportunity to win at another time. Double elimination design allows specific characteristic-poor performance of a quality algorithm under the special environment still to be able to survive through competitions and win it all.

3.3.2 Ensemble Method with Double-Tournament Selection

The goal of Ensemble Metrics is that Ensemble Metrics can benefit us for its specific ensemble advantages which accomplish the work that single metric cannot.

In the following chapter, we choose five state-of-the-art MOEAs to compare and five performance metrics to assess, every algorithm need to run 50 times because of the stochastic nature of MOEAs. In every running time, one algorithm produces one approximation front. Given the same initial population, five fronts from five different algorithms go to competition in a pool under evaluation of a randomly chosen performance metric and one best front wins according to this metric. After 50 running times, 50 winners are generated. Here, maybe many of 50 winners come from the same MOEA or none of 50 winners represents a specific algorithm.

In every 50 running time, the probability of each metric to be chosen is 0.2, so the average times each metric to be used is 10. This guarantee every metric to be chosen often and the 50 winners are decided by all five metrics collectively.

Then, these 50 winners are taken as the input to Double-Tournament Selection. Here, we just consider 50 fronts as 50 individuals without concerning about its representing algorithm.

Process:

1) Every 2 of 50 individuals are randomly picked to form a competition pair. For every competition, one metric is randomly chosen. Then, the total 25 pairs generate 25 winners and 25 losers. Create two parts: one contains 25 winners named winner bracket; the other contains 25 losers named loser bracket.
2) In each of the part, first randomly choose an individual, the remaining 24 individuals form 12 pairs of competitions; every pair uses a randomly chosen metric, too. Therefore, every part has 12 new winners and 12 new losers. Put the first chosen individual into both winners and losers. This will make the number of both winners and losers to be 13.
3) 13 losers in winner bracket and 13 winners in loser bracket (i.e., each individual losss only once) are combined together to compete. Every competition contains one individual from winner bracket and one from loser bracket. Then, 13 winners are generated and consist of a new loser bracket. The other 13 losers and 13 losers in original loser part (i.e. each losses twice) are eliminated.
4) Now, both new winner bracket and new loser bracket contain 13 individuals. We finish this step that reduces the number of competitors from 50 to 26 (i.e., 13 winners + 13 losers).
5) Then continue to do Step 2 and Step 3. We reduce the number of competitors from 26 to 14 (7 winners + 7 losers), then from 14 to 8 (4 winners + 4 losers), then from 8 to 4 (2 winners + 2 losers), then from 4 to 2 (1 winners + 1 losers) and finally from 2 down to 1. The last remaining individual is the final champion.
6) Check which evolutionary algorithm regarding the final winner comes from. Then we can conclude which algorithm is the best.
7) Remove all the fronts come from the best algorithm and compare other fronts from Step 1 to Step 6 again. So we can arrive at the second best one.
8) Continue to do Step 7; finally, we obtain the ranking of all the algorithms. The whole

process is end.

In this modified Double-Tournament Selection, to be the final winner, the MOEA must defeat others under all the performance metrics in double elimination because of stochastic mechanism for choosing metrics in every competition time. Therefore, the rank of all the algorithms is based on all the metrics collectively.

Figure 3.2 uses graphs to explain each step of Double-Tournament Selection:

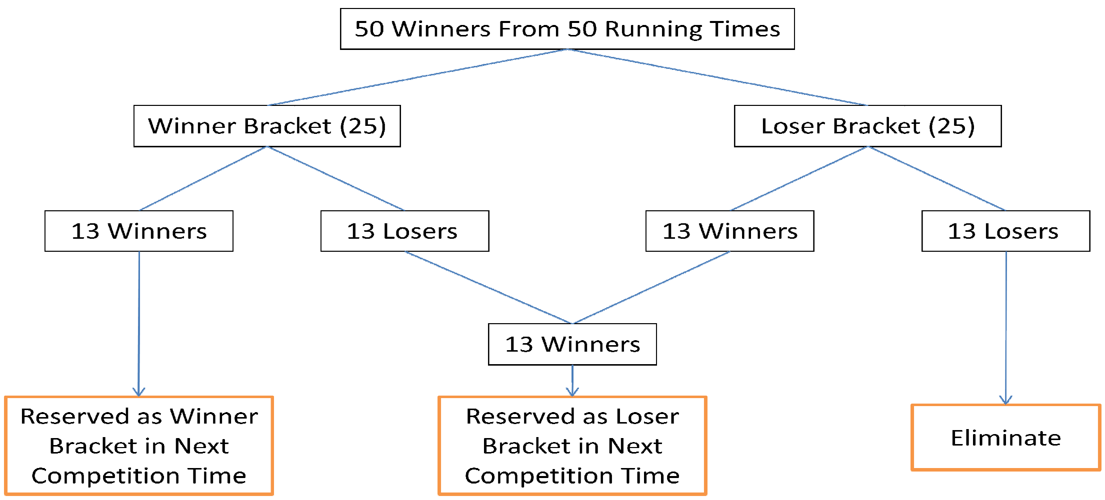


Fig 3.2 (a) From 50 individuals to 26 individuals

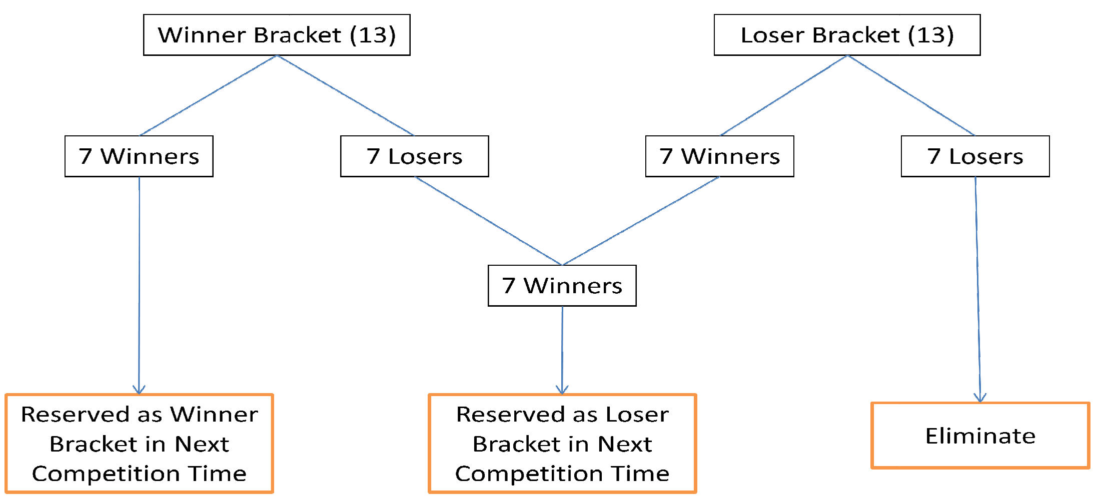


Fig 3.2 (b) From 26 individuals to 14 individuals

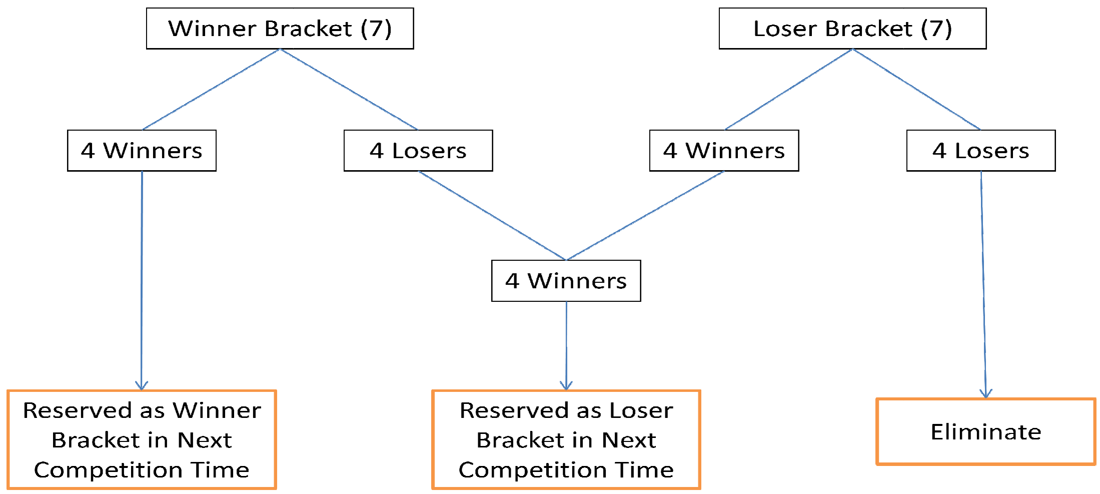


Fig 3.2 (c) From 14 individuals to 8 individuals

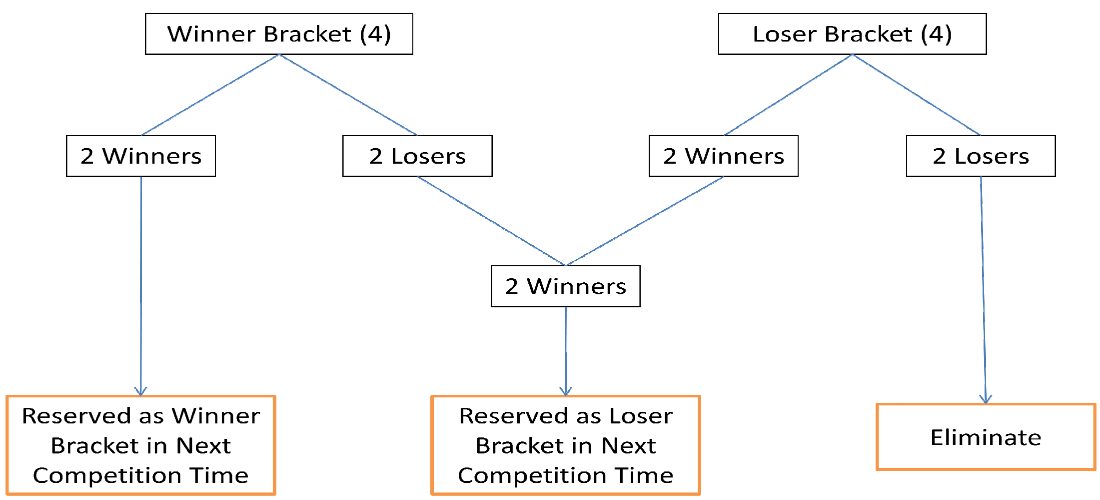


Fig 3.2 (d) From 8 individuals to 4 individuals

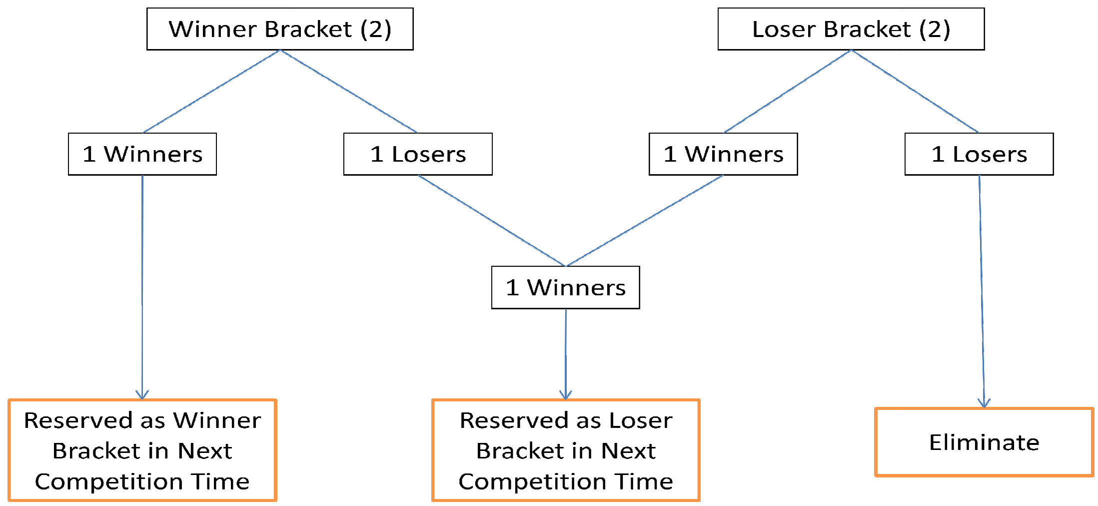


Fig 3.2 (e) From 4 individuals to 2 individuals

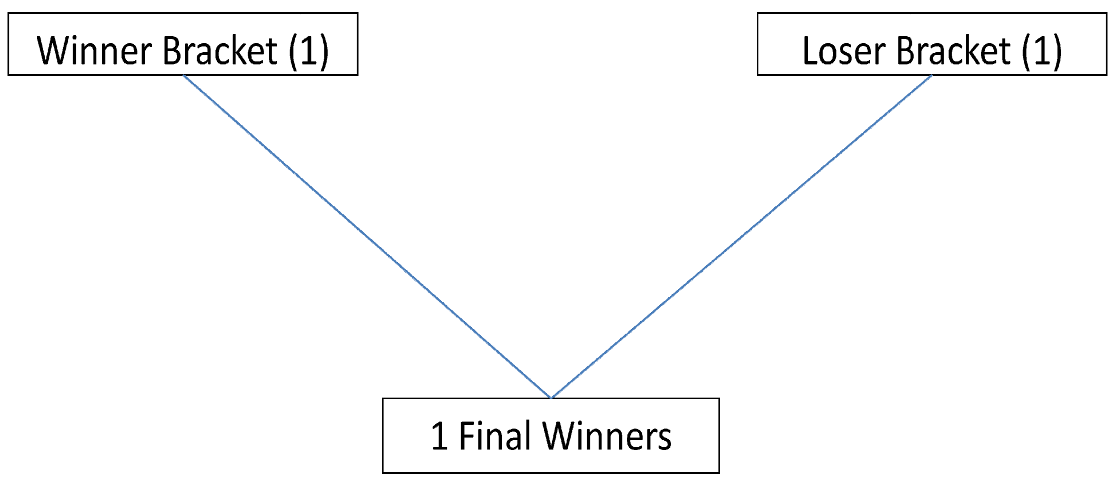


Fig 3.2 (f) From 2 individuals to 1 individuals

CHAPTER 4

FINDINGS

4.1 EXPERIEMENT RESULTS

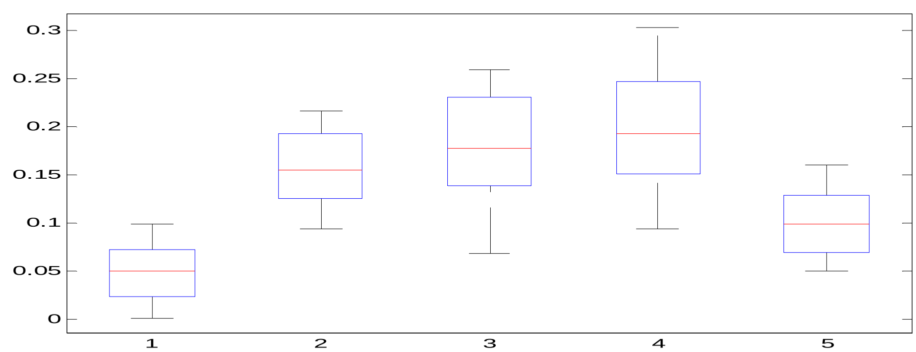
This part will show all experiment results in benchmark functions: ZDT1, ZDT2, ZDT3, ZDT4, ZDT6 and DTLZ2, respectively.

4.1.1 ZDT 1

First, box plot for every performance metric measure is presented:

- GD Metric

For each algorithm, the less the GD value, the better the algorithm's performance:

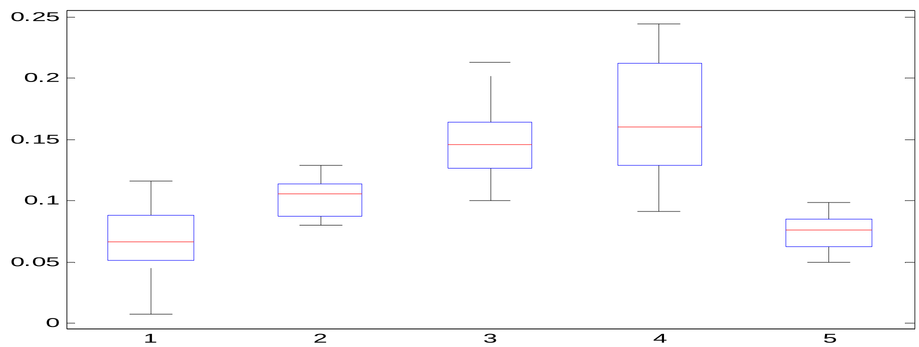


4.1(a) GD metric value in ZDT 1

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- Spacing Metric

For each algorithm, the less the Spacing value, the better the algorithm's performance:

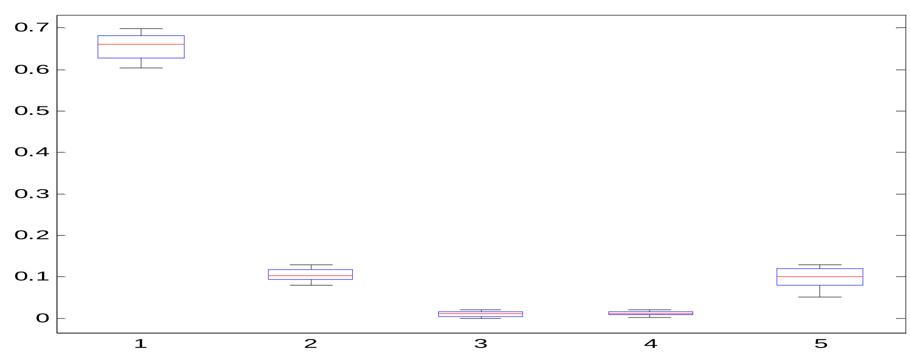


4.1(b) Spacing metric value in ZDT 1

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- NR Metric

For each algorithm, the more the NR value, the better the algorithm's performance:

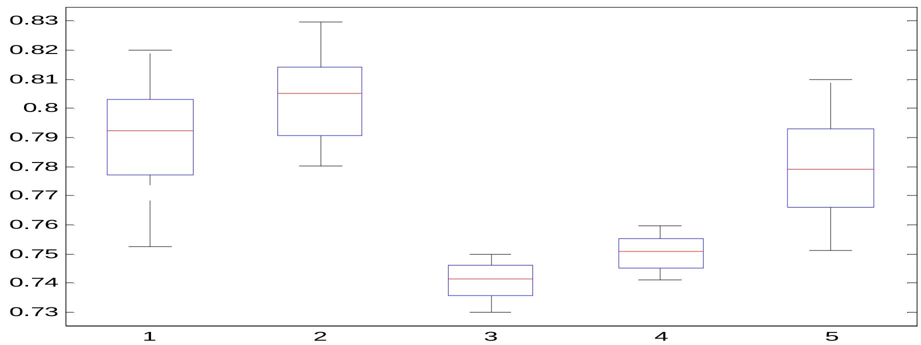


4.1(c) NR metric value in ZDT 1

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *S*-metric

For each algorithm, the more the *S* value, the better the algorithm's performance

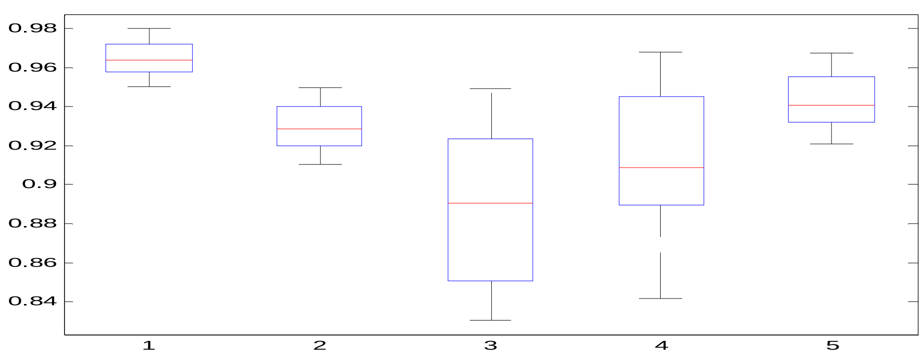


4.1(d) *S*-metric value in ZDT 1

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *MS* Metric

For each algorithm, the more the MS value, the better the algorithm's performance



4.1(e) MS metric value in ZDT 1

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

Then, experiment results using ensemble performance metrics in ZDT 1 is given:

Step 1 generates 50 fronts as the initial population of Double-Tournament Selection: in these 50 winner fronts, SPEA 2 wins 18 times, NSGA-II wins 12 times, IBEA wins 2 times, PESA-II wins 4 times and MOEA/D wins 14 times. In this step, metrics are totally chosen 50 times: GD is chosen 17 times, NR is 12 times, Spacing is 8 times, *S*-metric is 6 times and MS is 7 times.

Step 2 is the first step of Double-Tournament Selection that 50 fronts are competed to generate 26 winners: in these 26 winner fronts, SPEA 2 wins 10 times, NSGA-II wins 7 times, IBEA wins 0 times, PESA-II wins 1 time and MOEA/D wins 8 times. In this step, metrics are totally chosen 62 times in two parts: The total 50 fronts are divided into 2 groups in first 25 times: GD is chosen 4 times, NR is 4 times, Spacing is 6 times, S-metric is 6 times and MS is 5 times. 26 winners are generated from both Winner group and Loser group in 37 times: GD is chosen 7 times, NR is 4 times, Spacing is 8 times, S-metric is 8 times and MS is 10 times.

Step 3 is the second step of Double-Tournament Selection that 26 fronts are compared to generate 14 winners: in these 14 winner fronts, SPEA 2 wins 5 times, NSGA-II wins 4 times, IBEA wins 0 times, PESA-II wins 1 time and MOEA/D wins 4 times. In this step, metrics are totally chosen 19 times: GD is chosen 5 times, NR is 4 times, Spacing is 5 times, S-metric is 2 times and MS is 3 times.

In the third step (Step 4) of Double-Tournament Selection that 14 fronts are compared to generate 8 winners: in these 8 winner fronts, SPEA 2 wins 3 times, NSGA-II wins 3 times, IBEA wins 1 time, PESA-II wins 0 times and MOEA/D wins 1 time. In this step, metrics are totally chosen 10 times: GD is chosen 2 times, NR is 2 times, Spacing is 3 times, S-metric is 2 times and MS is 1 time.

Step 5 is the fourth step of Double-Tournament Selection that 8 fronts are compared to generate 4 winners: in these 4 winner fronts, SPEA 2 wins 2 times, NSGA-II wins 1 time, IBEA wins 0 times, PESA-II wins 0 times and MOEA/D wins 1 time. In this step, metrics are totally chosen 6 times: GD is chosen 0 times, NR is 1 time, Spacing is 1 time, S-metric is 2 times and MS is 2 times.

In the fifth step (Step 6) of Double-Tournament Selection that 4 fronts are compared to generate 2 winners: in these 2 winner fronts, SPEA 2 wins 1 time, NSGA-II wins 1 times, IBEA wins 0 time, PESA-II wins 0 times and MOEA/D wins 0 time. In this step, metrics are totally chosen 3 times: GD is chosen 1 time, NR is 1 time, Spacing is 0 times, S-metric is 0 times and MS is 1 time.

In the final step (Step 7) of Double-Tournament Selection that 2 fronts are compared to generate 1 winner. The final winner is SPEA 2 and GD is chosen to compare.

In Step 8, remove all the fronts from SPEA 2 in 50 fronts obtained in the first step, continue step 1 to step 7, NSGA-II is the second best one and MOEA/D is the third one. After all the remaining fronts come from the same algorithm, we get the final rank value for ZDT 1:

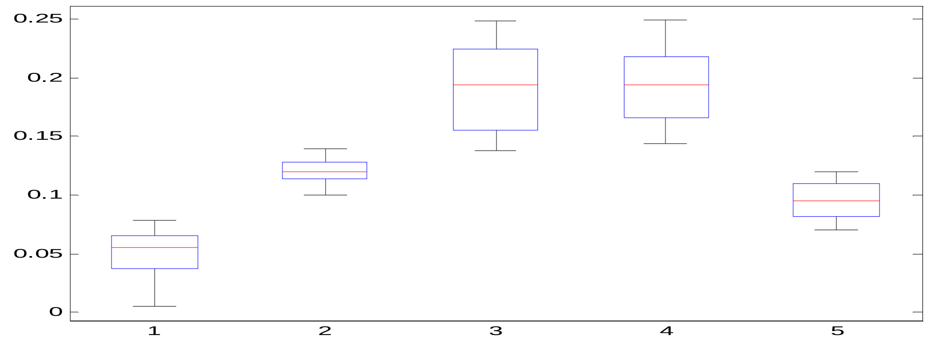
Rank 1: SPEA 2; Rank 2: NSGA-II; Rank 3: MOEA/D; Rank 4: PESA-II; Rank 5: IBEA.

4.1.2 ZDT2

First, box plot for every performance metric measure is presented:

- GD Metric

For each algorithm, the less the GD value, the better the algorithm’s performance:

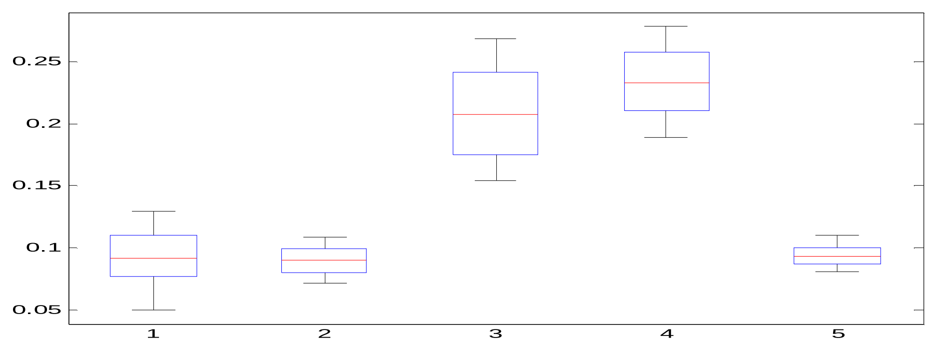


4.2(a) GD metric value in ZDT 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- Spacing Metric

For each algorithm, the less the Spacing value, the better the algorithm's performance:

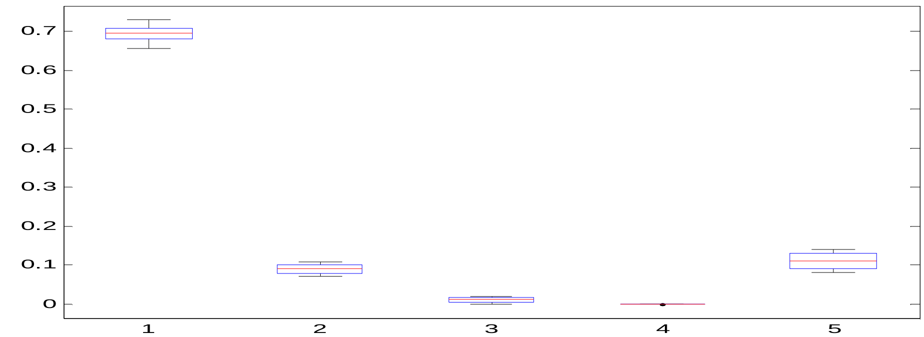


4.2(b) Spacing metric value in ZDT 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- NR Metric

For each algorithm, the more the NR value, the better the algorithm's performance:

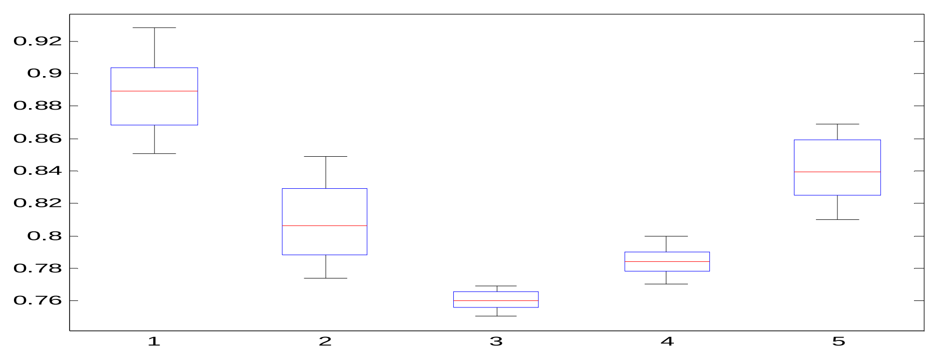


4.2(c) NR metric value in ZDT 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *S*-metric

For each algorithm, the more the *S* value, the better the algorithm's performance:

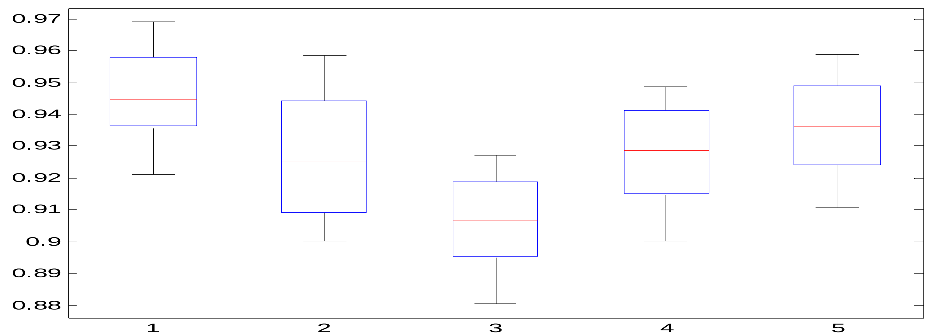


4.2(d) *S*-metric value in ZDT 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *MS* Metric

For each algorithm, the more the MS value, the better the algorithm's performance:



4.2(e) MS metric value in ZDT 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

Then, experiment results using ensemble performance metrics in ZDT 2 is given:

Step 1 generates 50 fronts as the initial population of Double-Tournament Selection: in these 50 winner fronts, SPEA 2 wins 13 times, NSGA-II wins 11 times, IBEA wins 8 times, PESA-II wins 5 times and MOEA/D wins 13 times. In this step, metrics are totally chosen 50 times: GD is chosen 9 times, NR is 11 times, Spacing is 10 times, S-metric is 11 times and MS is 9 times.

Step 2 is the first step of Double-Tournament Selection that 50 fronts are competed to generate 26 winners: in these 26 winner fronts, SPEA 2 wins 7 times, NSGA-II wins 6 times, IBEA wins 3 times, PESA-II wins 3 times and MOEA/D wins 7 times. In this step, metrics are totally chosen 62 times in two parts: The total 50 fronts are divided into 2 groups in first 25 times: GD is chosen 6 times, NR is 3 times, Spacing is 4 times, S-metric is 7 times and MS is 5 times. 26 winners are generated from both Winner group and Loser group in 37 times: GD is chosen 9 times, NR is 5 times, Spacing is 6 times, S-metric is 8 times and MS is 9 times.

Step 3 is the second step of Double-Tournament Selection that 26 fronts are compared to generate 14 winners: in these 14 winner fronts, SPEA 2 wins 5 times, NSGA-II wins 2 times, IBEA wins 1 time, PESA-II wins 2 times and MOEA/D wins 4 times. In this step, metrics are totally chosen 19 times: GD is chosen 7 times, NR is 4 times, Spacing is 5 times, S-metric is 2 times and MS is 1 time.

In the third step (Step 4) of Double-Tournament Selection that 14 fronts are compared to generate 8 winners: in these 8 winner fronts, SPEA 2 wins 2 times, NSGA-II wins 1 time, IBEA wins 1 time, PESA-II wins 1 time and MOEA/D wins 3 times. In this step, metrics are totally chosen 10 times: GD is chosen 1 time, NR is 3 times, Spacing is 2 times, S-metric is 2 times and MS is 2 times.

Step 5 is the fourth step of Double-Tournament Selection that 8 fronts are compared to generate 4 winners: in these 4 winner fronts, SPEA 2 wins 2 times, NSGA-II wins 1 times, IBEA wins 0 times, PESA-II wins 0 times and MOEA/D wins 1 times. In this step, metrics are totally chosen 6 times: GD is chosen 1 time, NR is 1 time, Spacing is 2 times, S-metric is 2 times and MS is 0 times.

In the fifth step (Step 6) of Double-Tournament Selection that 4 fronts are compared to generate 2 winners: in these 2 winner fronts, SPEA 2 wins 1 time and MOEA/D wins 1 time. In this step, metrics are totally chosen 3 times: GD is chosen 0 time, NR is 0 time, Spacing is 1 time, S-metric is 1 time and MS is 1 time.

In the final step (Step 7) of Double-Tournament Selection that 2 fronts are compared to generate 1 winner. The final winner is SPEA 2 and NR is chosen to compare.

In Step 8, remove all the fronts from SPEA 2 in 50 fronts obtained in the first step, continue step 1 to step 7, NSGA-II is the second best one and MOEA/D is the third one. After all the remaining fronts come from the same algorithm, we get the final rank value for ZDT 2:

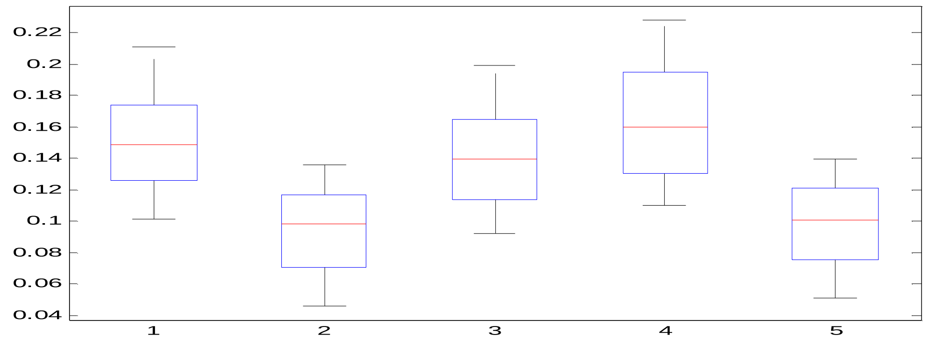
Rank 1: SPEA 2; Rank 2: NSGA-II; Rank 3: MOEA/D; Rank 4: IBEA; Rank 5: PESA-II.

4.1.3 ZDT 3

First, box plot for every performance metric measure is presented:

- GD Metric

For each algorithm, the less the GD value, the better the algorithm's performance:

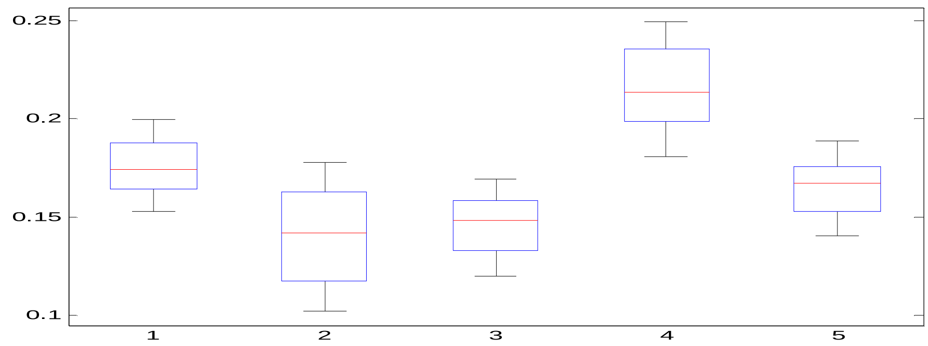


4.3(a) GD metric value in ZDT 3

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- Spacing Metric

For each algorithm, the less the Spacing value, the better the algorithm's performance:

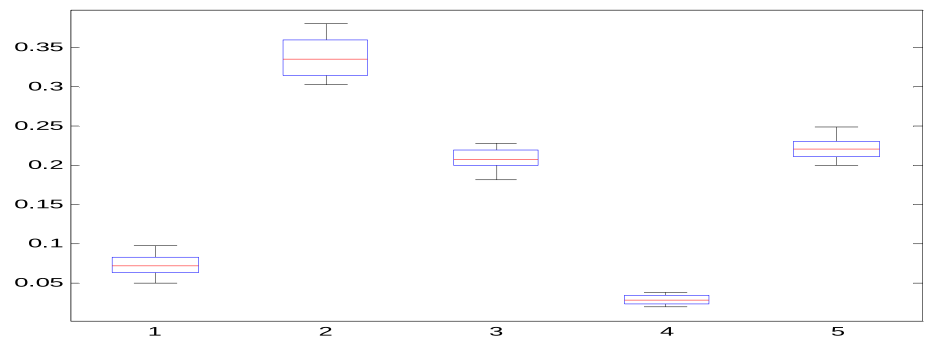


4.3(b) Spacing metric value in ZDT 3

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- NR Metric

For each algorithm, the more the NR value, the better the algorithm's performance:

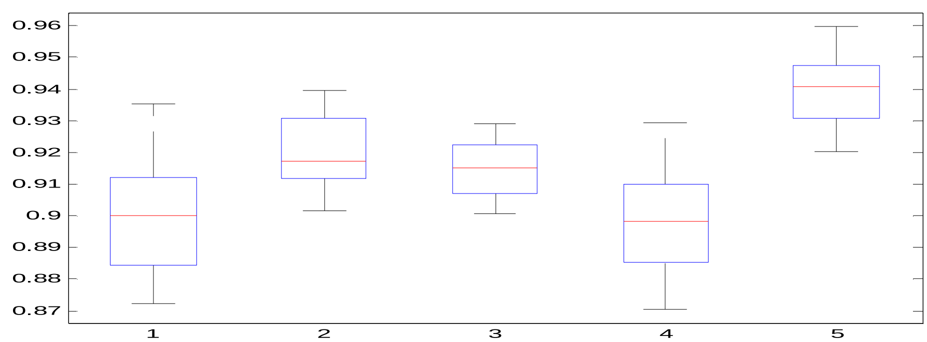


4.3(c) NR metric value in ZDT 3

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *S*-metric

For each algorithm, the more the *S* value, the better the algorithm's performance:

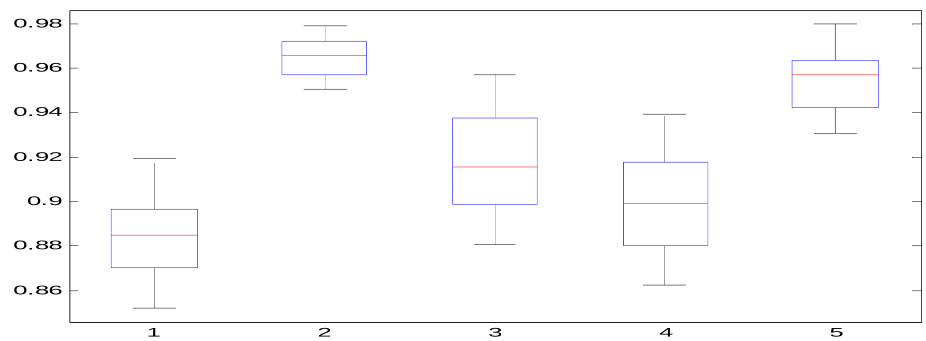


4.3(d) *S*-metric value in ZDT 3

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *MS* Metric

For each algorithm, the more the MS value, the better the algorithm's performance:



4.3(e) MS metric value in ZDT 3

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

Then, experiment results using ensemble performance metrics in ZDT 3 is given:

Step 1 generates 50 fronts as the initial population of Double-Tournament Selection: in these 50 winner fronts, SPEA 2 wins 11 times, NSGA-II wins 12 times, IBEA wins 9 times, PESA-II wins 7 times and MOEA/D wins 11 times. In this step, metrics are totally chosen 50 times: GD is chosen 10 times, NR is 13 times, Spacing is 7 times, S-metric is 12 times and MS is 8 times.

Step 2 is the first step of Double-Tournament Selection that 50 fronts are competed to generate 26 winners: in these 26 winner fronts, SPEA 2 wins 6 times, NSGA-II wins 6 times, IBEA wins 5 times, PESA-II wins 3 times and MOEA/D wins 6 times. In this step, metrics are totally chosen 62 times in two parts: the total 50 fronts are divided into 2 groups in first 25 times: GD is chosen 4 times, NR is 5 times, Spacing is 4 times, S-metric is 5 times and MS is 7 times. 26 winners are generated from both Winner group and Loser group in 37 times: GD is chosen 7 times, NR is 6 times, Spacing is 7 times, S-metric is 10 times and MS is 7 times.

Step 3 is the second step of Double-Tournament Selection that 26 fronts are compared to generate 14 winners: in these 14 winner fronts, SPEA 2 wins 2 times, NSGA-II wins 4 times, IBEA wins 3 times, PESA-II wins 2 times and MOEA/D wins 3 times. In this step, metrics are totally chosen 19 times: GD is chosen 4 times, NR is 5 times, Spacing is 3 times, S-metric is 4 times and MS is 3 times.

In the third step (Step 4) of Double-Tournament Selection that 14 fronts are compared to generate 8 winners: in these 8 winner fronts, SPEA 2 wins 1 time, NSGA-II wins 3 times, IBEA wins 1 time, PESA-II wins 1 time and MOEA/D wins 2 times. In this step, metrics are totally chosen 10 times: GD is chosen 2 times, NR is 2 times, Spacing is 3 times, S-metric is 1 time and MS is 2 times.

Step 5 is the fourth step of Double-Tournament Selection that 8 fronts are compared to generate 4 winners: in these 4 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 2 times, IBEA wins 0 times, PESA-II wins 0 times and MOEA/D wins 2 times. In this step, metrics are totally chosen 6 times: GD is chosen 2 times, NR is 0 time, Spacing is 0 times, S-metric is 2 times and MS is 2 times.

In the fifth step (Step 6) of Double-Tournament Selection that 4 fronts are compared to generate 2 winners: in these 2 winner fronts, NSGA-II wins 2 times. In this step, metrics are totally chosen 3 times: GD is chosen 1 time, NR is 2 times, Spacing is 0 time, S-metric is 0 time and MS is 0 time.

In the final step (Step 7) of Double-Tournament Selection that 2 fronts are compared to generate 1 winner. The final winner is NSGA-II and S-metric is chosen to compare.

In the Step 8, remove all the fronts from NSGA-II in 50 fronts obtained in the first step, continue step 1 to step 7, MOEA/D is the second best one and IBEA is the third one. After all the remaining fronts come from the same algorithm, we get the final rank value for ZDT 3:

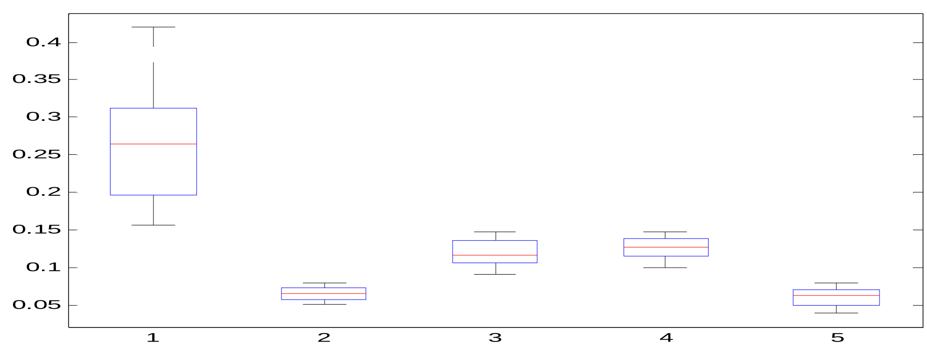
Rank 1: NSGA-II; Rank 2: MOEA/D; Rank 3: IBEA; Rank 4: SPEA 2; Rank 5: PESA-II.

4.1.4 ZDT 4

First, box plot for every performance metric measure is presented:

- GD Metric

For each algorithm, the less the GD value, the better the algorithm's performance:

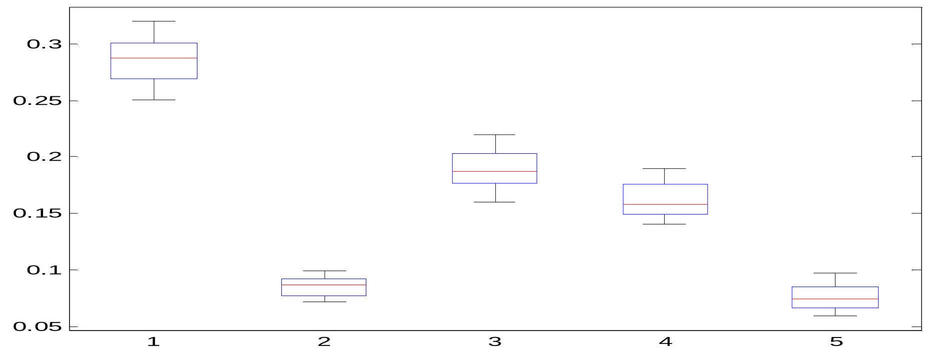


4.4(a) GD metric value in ZDT 4

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- Spacing Metric

For each algorithm, the less the Spacing value, the better the algorithm's performance:

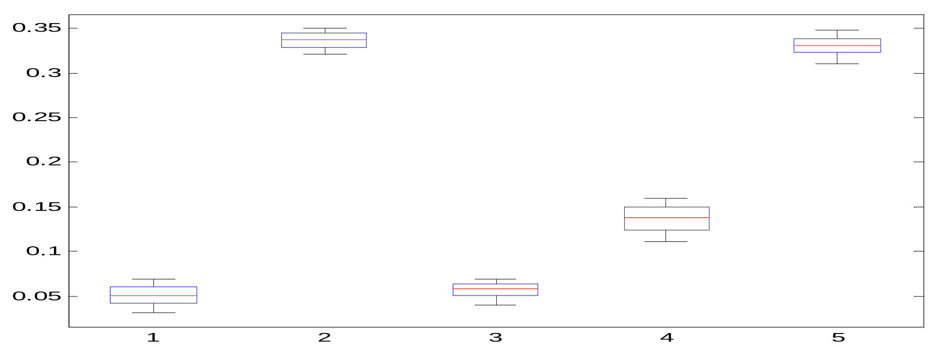


4.4(b) Spacing metric value in ZDT 4

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- NR Metric

For each algorithm, the more the NR value, the better the algorithm's performance:

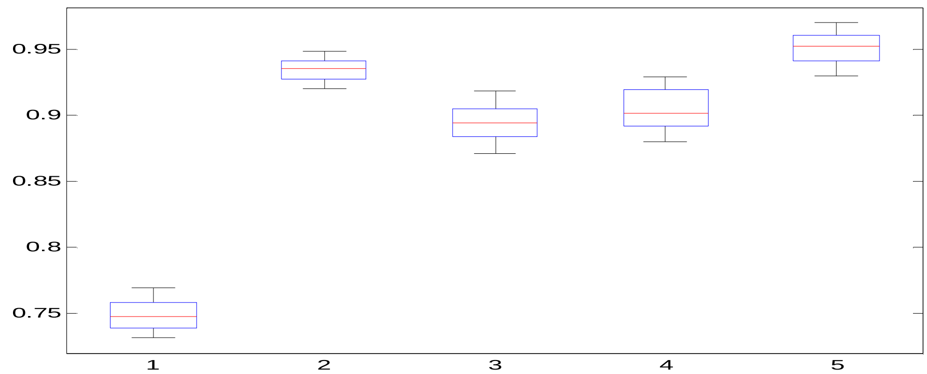


4.4(c) NR metric value in ZDT 4

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *S*-metric

For each algorithm, the more the S value, the better the algorithm's performance:

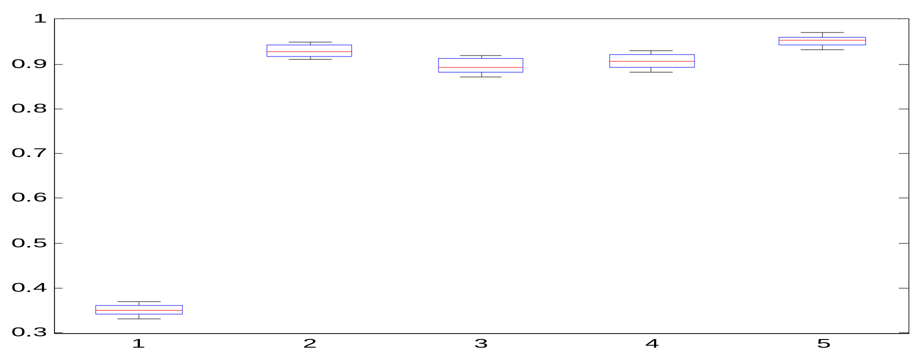


4.4(d) *S*- metric value in ZDT 4

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- MS Metric

For each algorithm, the more the MS value, the better the algorithm's performance:



4.4(e) MS metric value in ZDT 4

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

Then, experiment results using ensemble performance metrics in ZDT 4 is given:

Step 1 generates 50 fronts as the initial population of Double-Tournament Selection: in these 50 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 15 times, IBEA wins 9 times, PESA-II wins 9 times and MOEA/D wins 17 times. In this step, metrics are totally chosen 50 times: GD is chosen 9 times, NR is 15 times, Spacing is 7 times, S-metric is 12 times and MS is 7 times.

Step 2 is the first step of Double-Tournament Selection that 50 fronts are competed to generate 26 winners: in these 26 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 9 times, IBEA wins 5 times, PESA-II wins 5 times and MOEA/D wins 7 times. In this step, metrics are totally chosen 62 times in two parts: the total 50 fronts are divided into 2 groups in first 25 times: GD is chosen 6 times, NR is 4 times, Spacing is 4 times, S-metric is 5 times and MS is 6 times. 26 winners are generated from both Winner group and Loser group in 37 times: GD is chosen 5 times, NR is 7 times, Spacing is 9 times, S-metric is 5 times and MS is 11 times.

Step 3 is the second step of Double-Tournament Selection that 26 fronts are compared to generate 14 winners: in these 14 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 5 times, IBEA wins 3 times, PESA-II wins 2 times and MOEA/D wins 4 times In this step, metrics are totally chosen 19 times: GD is chosen 5 times, NR is 6 times, Spacing is 3 times, S-metric is 4 times and MS is 1 time.

In the third step (Step 4) of Double-Tournament Selection that 14 fronts are compared to generate 8 winners: in these 8 winner fronts, SPEA 2 wins 0 time, NSGA-II wins 3 times, IBEA wins 1 time, PESA-II wins 1 time and MOEA/D wins 3 times. In this step, metrics are totally chosen 10 times: GD is chosen 3 times, NR is 1 time, Spacing is 2 times, S-metric is 2 times and MS is 2 times.

Step 5 is the fourth step of Double-Tournament Selection that 8 fronts are compared to generate 4 winners: in these 4 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 2 times, IBEA wins 1 time, PESA-II wins 0 times and MOEA/D wins 1 time. In this step, metrics are totally chosen 6 times: GD is chosen 1 time, NR is 0 time, Spacing is 1 time, S-metric is 2 times and MS is 2 times.

In the fifth step (Step 6) of Double-Tournament Selection that 4 fronts are compared to generate 2 winners: in these 2 winner fronts, NSGA-II wins 1 time and MOEA/D wins 1 time. In this step, metrics are totally chosen 3 times: GD is chosen 1 time, NR is 0 times, Spacing is 1 time, S-metric is 0 time and MS is 1 time.

In the final step (Step 7) of Double-Tournament Selection that 2 fronts are compared to generate 1 winner. The final winner is MOEA/D and GD is chosen to compare. This result is the same as [14].

In the Step 8, remove all the fronts from MOEA/D in 50 fronts obtained in the first step, continue step 1 to step 7, NSGA-II is the second best one and PESA-II is the third one. After all the remaining fronts come from the same algorithm, we get the final rank value for ZDT 4:

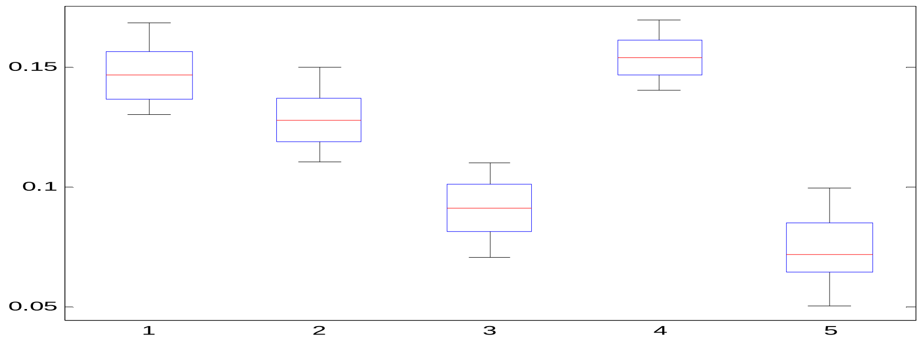
Rank 1: MOEA/D; Rank 2: NSGA-II; Rank 3: PESA-II; Rank 4: IBEA; Rank 5: SPEA 2.

4.1.5 ZDT 6

First, box plot for every performance metric measure is presented:

- GD Metric

For each algorithm, the less the GD value, the better the algorithm's performance:

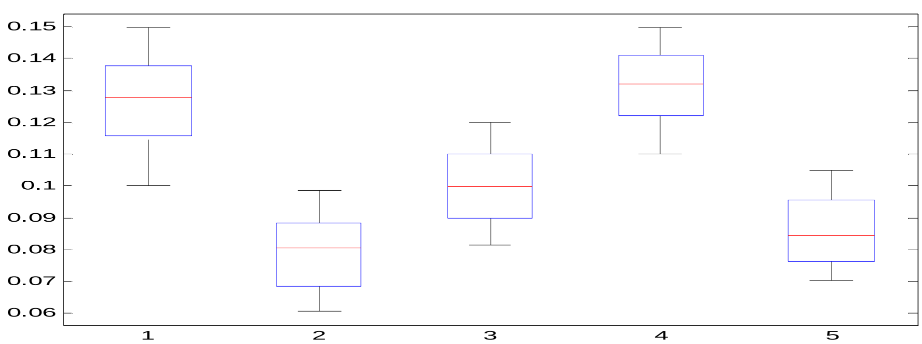


4.5(a) GD metric value in ZDT 6

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- Spacing Metric

For each algorithm, the less the Spacing value, the better the algorithm's performance:

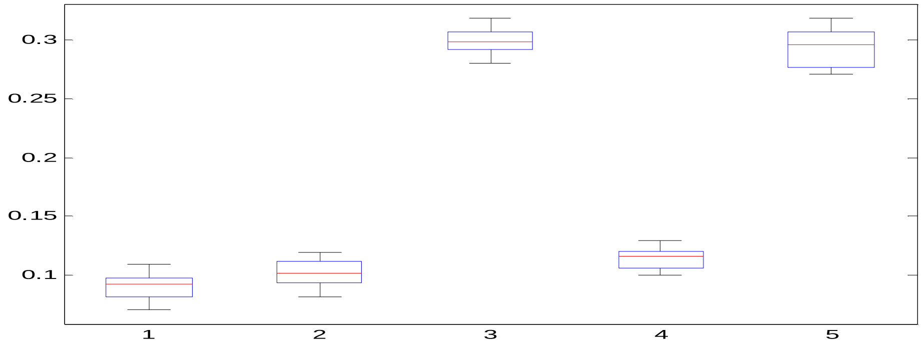


4.5(b) Spacing metric value in ZDT 6

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- NR Metric

For each algorithm, the more the NR value, the better the algorithm's performance:

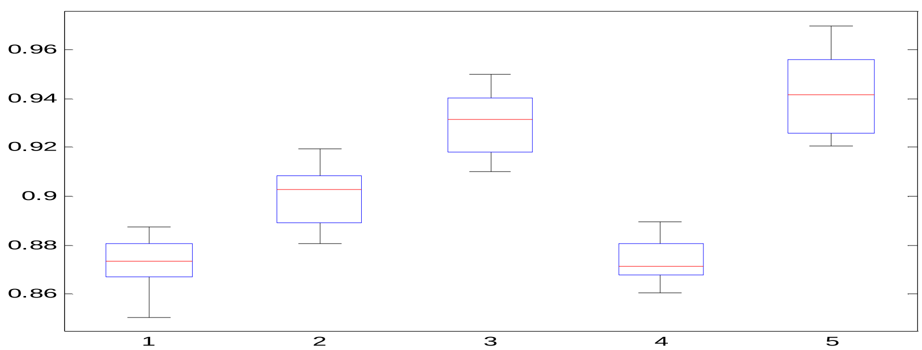


4.5(c) NR metric value in ZDT 6

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *S*-metric

For each algorithm, the more the *S* value, the better the algorithm's performance:

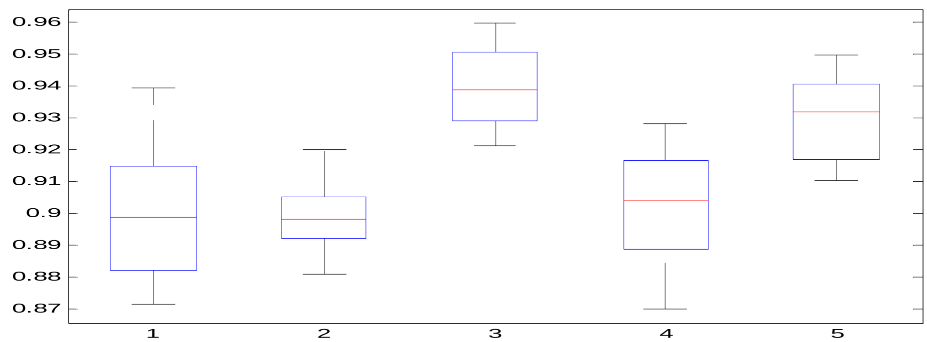


4.5(d) *S*- metric value in ZDT 6

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- MS Metric

For each algorithm, the more the MS value, the better the algorithm's performance:



4.5(e) MS metric value in ZDT 6

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

Then, experiment results using ensemble performance metrics in ZDT 6 is given:

Step 1 generates 50 fronts as the initial population of Double-Tournament Selection: in these 50 winner fronts, SPEA 2 wins 8 times, NSGA-II wins 11 times, IBEA wins 13 times, PESA-II wins 6 times and MOEA/D wins 12 times. In this step, metrics are totally chosen 50 times: GD is chosen 11 times, NR is 12 times, Spacing is 9 times, S-metric is 12 times and MS is 6 times.

Step 2 is the first step of Double-Tournament Selection that 50 fronts are competed to generate 26 winners: in these 26 winner fronts, SPEA 2 wins 5 times, NSGA-II wins 5 times, IBEA wins 7 times, PESA-II wins 2 times and MOEA/D wins 7 times. In this step, metrics are totally chosen 62 times in two parts: the total 50 fronts are divided into 2 groups in first 25 times: GD is chosen 5 times, NR is 7 times, Spacing is 4 times, S-metric is 4 times and MS is 5 times. 26 winners are generated from both Winner group and Loser group in 37 times: GD is chosen 6 times, NR is 5 times, Spacing is 8 times, S-metric is 9 times and MS is 9 times.

Step 3 is the second step of Double-Tournament Selection that 26 fronts are compared to generate 14 winners: in these 14 winner fronts, SPEA 2 wins 2 times, NSGA-II wins 3 times, IBEA wins 4 times, PESA-II wins 1 time and MOEA/D wins 4 times. In this step, metrics are totally chosen 19 times: GD is chosen 6 times, NR is 5 times, Spacing is 2 times, S-metric is 4 times and MS is 2 times.

In the third step (Step 4) of Double-Tournament Selection that 14 fronts are compared to generate 8 winners: in these 8 winner fronts, SPEA 2 wins 0 time, NSGA-II wins 2 times, IBEA wins 3 times, PESA-II wins 0 time and MOEA/D wins 3 times. In this step, metrics are totally chosen 10 times: GD is chosen 2 times, NR is 1 time, Spacing is 1 time, S-metric is 2 times and MS is 4 times.

Step 5 is the fourth step of Double-Tournament Selection that 8 fronts are compared to generate 4 winners: in these 4 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 0 times, IBEA wins 2 times, PESA-II wins 0 times and MOEA/D wins 2 times. In this step, metrics are totally chosen 6 times: GD is chosen 3 times, NR is 0 time, Spacing is 1 time, S-metric is 1 time and MS is 1 time.

In the fifth step (Step 6) of Double-Tournament Selection that 4 fronts are compared to generate 2 winners: in these 2 winner fronts, IBEA wins 1 time and MOEA/D wins 1 time. In this step, metrics are totally chosen 3 times: GD is chosen 0 time, NR is 0 times, Spacing is 0 time, S-metric is 2 time and MS is 1 time.

In the final step (Step 7) of Double-Tournament Selection that 2 fronts are compared to generate 1 winner. The final winner is MOEA/D and Spacing is chosen to compare.

In the Step 8, remove all the fronts from MOEA/D in 50 fronts obtained in the first step, continue step 1 to step 7, IBEA is the second best one and NSGA-II is the third one. After all the remaining fronts come from the same algorithm, we get the final rank value for ZDT 6:

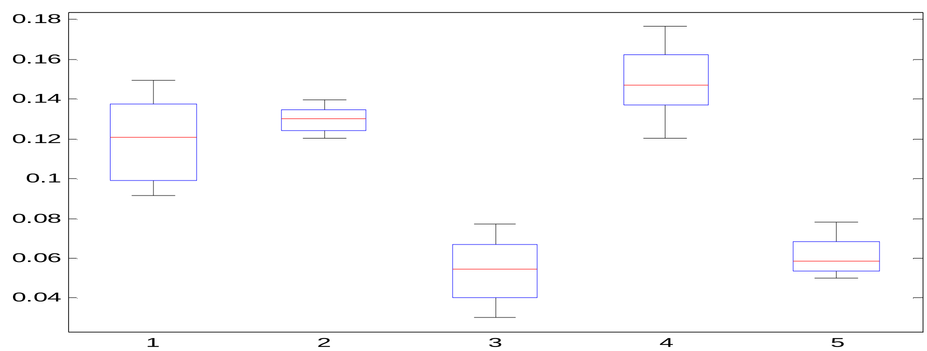
Rank 1: MOEA/D; Rank 2: IBEA; Rank 3: NSGA-II; Rank 4: SPEA 2; Rank 5: PESA-II.

4.1.6 DTLZ 2

First, box plot for every performance metric measure is presented:

- GD Metric

For each algorithm, the less the GD value, the better the algorithm's performance:

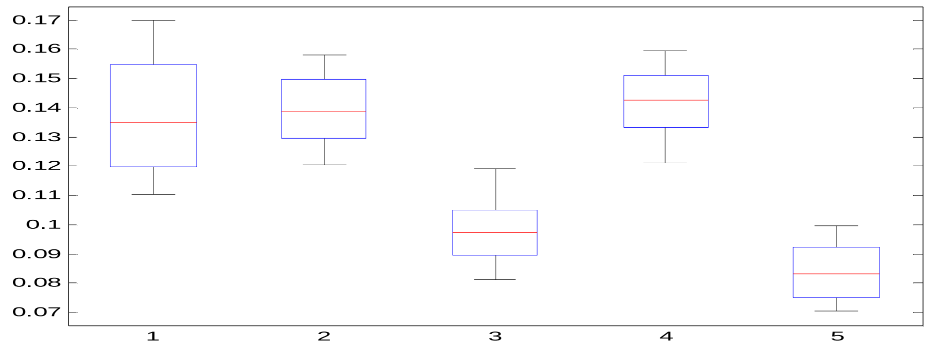


4.6(a) GD metric value in DTLZ 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- Spacing Metric

For each algorithm, the less the Spacing value, the better the algorithm's performance:

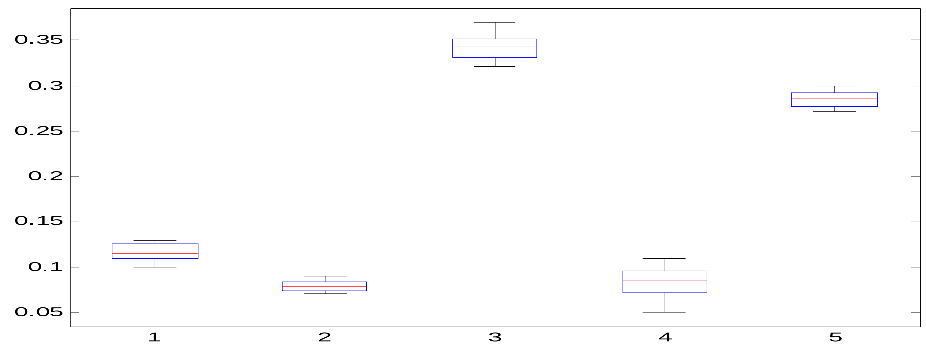


4.6(b) Spacing metric value in DTLZ 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- NR Metric

For each algorithm, the more the NR value, the better the algorithm's performance:

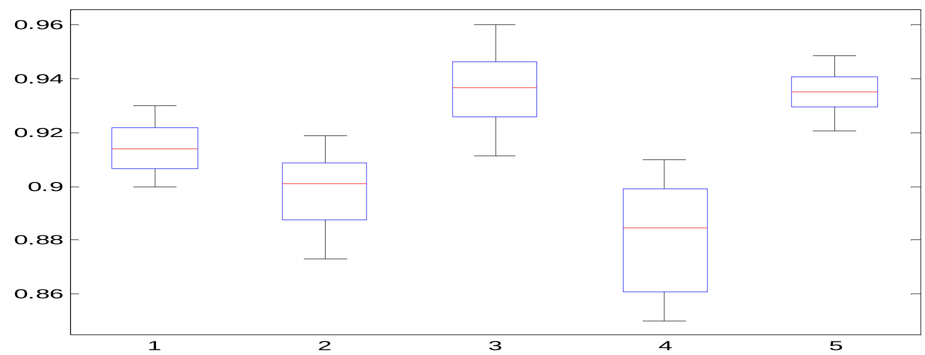


4.6(c) NR metric value in DTLZ 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- *S*-metric

For each algorithm, the more the *S* value, the better the algorithm's performance:

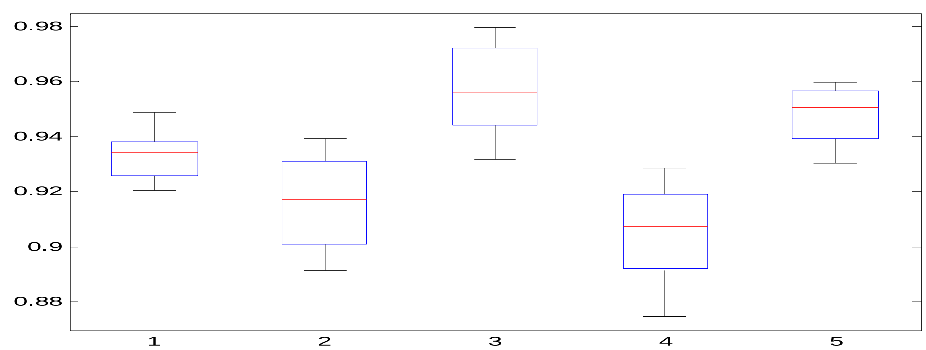


4.6(d) *S*-metric value in DTLZ 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

- MS Metric

For each algorithm, the more the MS value, the better the algorithm's performance:



4.6(e) MS metric value in DTLZ 2

Here, in graph's x axis, '1' represents SPEA 2, '2' represents NSGA-II,'3'represents IBEA, '4' represents PESA-II and '5' represents MOEA/D. y axis shows the metric value.

Then, experiment results using ensemble performance metrics in ZDT 6 is given:

Step 1 generates 50 fronts as the initial population of Double-Tournament Selection: in these 50 winner fronts, SPEA 2 wins 11 times, NSGA-II wins 8 times, IBEA wins 13 times, PESA-II wins 6 times and MOEA/D wins 12 times. In this step, metrics are totally chosen 50 times: GD is chosen 7 times, NR is 15 times, Spacing is 9 times, S-metric is 11 times and MS is 8 times.

Step 2 is the first step of Double-Tournament Selection that 50 fronts are competed to generate 26 winners: in these 26 winner fronts, SPEA 2 wins 5 times, NSGA-II wins 5 times, IBEA wins 7 times, PESA-II wins 2 times and MOEA/D wins 7 times. In this step, metrics are totally chosen 62 times in two parts: the total 50 fronts are divided into 2 groups in first 25 times: GD is chosen 6 times, NR is 4 times, Spacing is 7 times, S-metric is 3 times and MS is 5 times. 26 winners are generated from both Winner group and Loser group in 37 times: GD is chosen 6 times, NR is 6 times, Spacing is 7 times, S-metric is 10 times and MS is 8 times.

Step 3 is the second step of Double-Tournament Selection that 26 fronts are compared to generate 14 winners: in these 14 winner fronts, SPEA 2 wins 2 times, NSGA-II wins 0 times, IBEA wins 6 times, PESA-II wins 1 time and MOEA/D wins 5 times. In this step, metrics are totally chosen 19 times: GD is chosen 2 times, NR is 2 times, Spacing is 5 times, S-metric is 4 times and MS is 6 times.

In the third step (Step 4) of Double-Tournament Selection that 14 fronts are compared to generate 8 winners: in these 8 winner fronts, SPEA 2 wins 1 time, NSGA-II wins 1 time, IBEA wins 3 times, PESA-II wins 0 time and MOEA/D wins 3 times. In this step, metrics are totally chosen 10 times: GD is chosen 3 times, NR is 2 times, Spacing is 1 time, S-metric is 2 times and MS is 2 times.

Step 5 is the fourth step of Double-Tournament Selection that 8 fronts are compared to generate 4 winners: in these 4 winner fronts, SPEA 2 wins 0 times, NSGA-II wins 0 times, IBEA wins 2 time, PESA-II wins 0 times and MOEA/D wins 2 times. In this step, metrics are totally chosen 6 times: GD is chosen 2 times, NR is 1 time, Spacing is 1 time, S-metric is 1 time and MS is 1 time.

In the fifth step (Step 6) of Double-Tournament Selection that 4 fronts are compared to generate 2 winners: in these 2 winner fronts, IBEA wins 1 time and MOEA/D wins 1 time. In this step, metrics are totally chosen 3 times: GD is chosen 1 time, NR is 0 times, Spacing is 1 time, *S*-metric is 0 time and MS is 1 time.

In the final step (Step 7) of Double-Tournament Selection that 2 fronts are compared to generate 1 winner. The final winner is IBEA and MS is chosen to compare.

In the Step 8, remove all the fronts from IBEA in 50 fronts obtained in the first step, continue step 1 to step 7, MOEA/D is the second best one and SPEA 2 is the third one. After all the remaining fronts come from the same algorithm, we get the final rank value for DTLZ 2:

Rank 1: IBEA; Rank 2: MOEA/D; Rank 3: SPEA 2; Rank 4: NSGA-II; Rank 5: PESA-II.

4.2 ANALYSIS OF EXPERIMENT RESULTS

4.2.1 Ensemble Performance Metrics give the same rank values to exist papers

In ZDT3, the final rank result is: Rank 1: NSGA-II; Rank 2: MOEA/D; Rank 3: IBEA; Rank 4: SPEA 2; Rank 5: PESA-II. The experiment result in [14] has also suggested that MOEA/D generate a worse result than NSGA-II.

In ZDT6, the final rank result is: Rank 1: MOEA/D; Rank 2: IBEA; Rank 3: NSGA-II; Rank 4: SPEA 2; Rank 5: PESA-II. [12] shows IBEA performs better than NSGA-II and SPEA 2 in ZDT 6. [14] gives the same result that MOEA/D is better than NSGA-II in ZDT 6.

In DTLZ 2, the final rank result is: Rank 1: IBEA; Rank 2: MOEA/D; Rank 3: SPEA 2; Rank 4: NSGA-II; Rank 5: PESA-II. This result is nearly the same as previous experiment: [10]

has suggested that SPEA 2 seems to have advantages over NSGA-II in higher dimensional objective space. In [12], IBEA is also better than SPEA 2 and NSGA-II. The experiment result in [14] has also identified that MOEA/D generate a better result than NSGA-II.

4.2.2 Summary of EAs to Solve Different Characteristics of Test Functions

- SPEA 2

  SPEA 2 is the final winner in problem ZDT 1 and ZDT 2. Although ZDT 1 has a convex Pareto-optimal front while ZDT 2 has the nonconvex counterpart to ZDT1, Both ZDT1 and ZDT2 have some common characteristics: they do not have local Pareto-optimal fronts and their global Pareto-optimal fronts are continuous. From the above reason, we can state that, if the test problem has continues global Pareto-optimal fronts and do not have local Pareto-optimal fronts, SPEA 2 will perform well in this problem.

- NSGA-II

  NSGA-II has the best performance in ZDT 3, which represents the discreteness feature and has a Pareto-optimal front consisting of several noncontiguous convex parts. Therefore, if there is a test problem with discrete Pareto-optimal front, we can propose that NSGA-II is the best algorithm to solve this problem.

- MOEA/D

  MOEA/D wins all other algorithms in both ZDT4 and ZDT6. ZDT4 is difficult to solve because it has many local Pareto-optimal fronts, a large number of local Pareto-optimal fronts make the global Pareto front is not easy to find and EAs need to exhibit their ability to deal with multimodality. ZDT6's Pareto-optimal solutions are nonuniformly distributed along the global Pareto front. The front is biased for solutions which have a large $f_1(x)$ value. Therefore, MOEA/D will exhibit its good performance when encounters the test problem which has lots of local Pareto-optimal fronts or Pareto-optimal solutions is not uniformly distributed its global Pareto front.

- IBEA

  IBEA wins at all in DTLZ 2, which is the only test problem chosen in this experiment has more than two objectives. We may not absolutely say that IBEA is best for solve problems with high-dimension objectives. But we can make a comparatively conclusion that IBEA can perform better than others in some test problems with high-dimension objectives.

In summary, from the above discussion, we can see more clearly that every algorithm can only be superior to another algorithm over some set of test problems, and then it must be inferior in other problems with different characteristics. This is also expected by No Free Lunch Theory.

4.3 WHY USE DOUBLE-ELIMINATION METHOD TO ENSEMBLE

First, we go through experiments in all benchmark functions considering the performance metrics' values of every algorithm in this benchmark function. In ZDT1, SPEA 2 is the final winner and it wins under all four metrics but is inferior to NSGA-II in *S*-metric. In ZDT2, SPEA 2 is the final winner and it wins under all four metrics but it is a little bit worse than NSGA-II in Spacing metric. In ZDT3, NSGA-II is the final winner and it wins under all four metrics but is inferior to MOEA/D in *S*-metric. In ZDT4, MOEA/D is the final winner and it wins under all four metrics but it is a little bit worse than NSGA-II in *NR* metric. In ZDT6, MOEA/D is the final winner but is inferior to IBEA in *MS* metric and a little bit worse than NSGA-II in Spacing metric. In DTLZ 2, IBEA is the final winner and it wins under all four metrics but is inferior to MOEA/D in Spacing metric.

From above results, to be a final winner does not mean win others in all performance metrics. In ZDT1, if we use *S*-metric to compare SPEA 2 and NSGA-II in Single-Elimination, SPEA 2 will be lost and we cannot find the best algorithm for this problem. However, if that condition happens in Double-Elimination, SPEA 2 also has another opportunity to win again.

Therefore, Double-Elimination can provide one more chance for every competitor, this helps to find the best one winner.

CHAPTER 5

CONCLUSION

. There are five types of unary metrics: 1) Metrics assessing the number of Pareto optimal solutions in the set: Pareto Dominance Indicator (*NR*), Overall Nondominated Vector Generation and Ratio (*ONVG*), Ratio of Non-dominated Individuals (*RNI*) and Error Ratio (*ER*). 2) Metrics measuring the closeness of the solution to the theoretical Pareto front: Generational Distance (*GD*) and Maximum Pareto Front Error (*MPFE*). 3) Metrics focusing on distribution of the solutions: Uniform Distribution (*UD*), Spacing and Number of Distinct Choices (NDC_u). 4) Metrics concerning spread of the solutions: Maximum Spread (*MS*). 5) Metrics considering both closeness and diversity: Hypervolume Indicator (or *S*-metric).

There are two types of binary metrics: 1) binary performance metrics based on unary quality indicator: ε -indicator I_{ε} , enclosing hypercube Indicator and coverage difference metrics (*D*-metric). The second type is direct comparison binary metrics: *C* metrics and *R* metrics.

An ensemble method is introduced to compare EAs by combining a large number of single metrics using modified Double Tournament Selection. Double elimination design give every individual two chances to competition allows characteristic poor performance of a quality algorithm under the special environment still to be able to win at all. Therefore, this ensemble mechanism can maximum protects the qualified individual from being lost by some stochastic factors in a comparison time. This ensures the final result is the really best one and the whole ensemble process is effective and precise.

Ensemble method can overcome the lost information problem by the single metric which only provides some specific but limited information and is only used effectively in some specified conditions. The Comprehensive comparison of the proposed algorithms on benchmark test functions under ensemble performance metrics show that:

SPEA 2 performs well in the problem has continues global Pareto-optimal fronts and do not have local Pareto-optimal fronts.

NSGA-II is the best algorithm to solve this problem with discrete Pareto-optimal front.

MOEA/D will exhibit its good performance when encounters the test problem which has lots of local Pareto-optimal fronts or Pareto-optimal solutions is not uniformly distributed its global Pareto front.

IBEA can perform better than others in some test problems with high-dimension objectives.

Furthermore, we are benefit from ensemble method that it is not necessary to spend much time to choose a suitable performance metric for a specific test problem. We do not need to try every metric to find which one is the best. Ensemble method avoids the choosing process which is a heavy computational process and can be directly used to assessing EAs.

From above statement, multiple performance metrics ensemble by applying Double-Tournament Selection can obtain better evaluation performance than could be obtained from any of single performance metric.

Performance metric ensemble is just the first step. In the future research work, combine benchmark functions together to test EAs based on this ensemble approach is needed to be focused. A comprehensive evaluation of EAs in all the test functions and under all the performance metrics is our ultimate goal.

REFERENCES

[1] Wolpert, D.H.; Macready, W.G.; , "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation,* vol.1, no.1, pp.67-82, Apr 1997

[2] E. Zitzler, L. Thiele, K. Deb. Comparison of Multiobjective Evolutionary Algorithms: Empirical Results. *IEEE Transactions on Evolutionary Computation,* 8(2):173–195, 2000

[3] Chi-Keong Goh and Kay Chen Tan. A Competitive-Cooperative Coevolutionary Paradigm for Dynamic Multiobjective Optimization. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION,* VOL. 13, NO. 1, FEBRUARY 2009.

[4] David A. Van Veldhuizen. Multiobjective Evolutionary Algorithms: Classifications, Analyses, and New Innovations. PhD thesis, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio, May 1999.

[5] K.C.Tan, T.H.Lee and E.F.Khor. Evolutionary Algorithms for Multi-Objective Optimization: Performance Assessments and Comparisons. *Artificial Intelligence Review,* Vol. 17, No. 4, pp. 253--290, June 2002.

[6] Schott, J. R. Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. MS thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Cambridge, Massachusetts, May 1995.

[7] Jin Wu and Shapour Azarm. Metrics for Quality Assessment of a Multiobjective Design Optimization Solution Set. *Transactions of the ASME,* 18/Vol, 123, MARCH 2001.

[8] Zitzler, E., and Thiele, L. ‘‘Multiobjective Optimization Using Evolutionary Algorithms—A Comparative Case Study,’’ In *Eiben, A. E., et al., Proc. 5th International Conference: Parallel Problem Solving from Nature—PPSNV,* Amsterdam, The Netherlands, Springer, pp.

292–301, 1998.

[9] Hansen, M.P., Jaszkiewicz, A., Evaluating the quality of approximations to the nondominated set, Technical Report IMM-REP-1998-7, Institute of Mathematical Modeling, Technical University of Denmark, 1998.

[10] E. Zitzler, L. Thiele, M. Laumanns, C. M. Foneseca, and V. Grunert da Fonseca. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, 7(2):117–132, 2003.

[11] E. Zitzler., Evolutionary algorithms for multiobjective optimization: Methods and applications, PhD thesis, Swiss Federal Institute of Technology Zurich, 1999.

[12] K. Deb, L. Thiele, M. Laumanns, and E. Zitzler, “Scalable multiobjective optimization test problems,” in *Proc. Congr. Evol. Comput.*, May 2002, vol. 1, pp. 825–830.

[13] E. Zitzler, M. Laumanns and L. Thiele, *SPEA2: Improving the Strength Pareto Evolutionary Algorithm*, Technical Report TIK-Report 103, Swiss Federal Institute of Technology, 2001.

[14] K. Deb, S. Agrawal, A. Pratab, and T. Meyarivan, “A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II,” in *Proc. Conf. Parallel Problem Solving from Nature VI*, pp. 849-858, 2000.

[15] Eckart Zitzler and Simon K. Indicator-Based Selection in Multiobjective Search. In Proc. 8th International Conference on Parallel Problem Solving form Nature (PPSN VIII). 2004.

[16] David W. Corne, Nick R. Jerram, Joshua D. Knowles, Martin J. Oates, Martin J. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. Proceedings of the Genetic and Evolutionary Computation Conference (GECCO’2001).2001.

[17] Qingfu Zhang; Wudong Liu; Tsang, E.; Virginas, B.; , "Expensive Multiobjective Optimization by MOEA/D With Gaussian Process Model," *Evolutionary Computation, IEEE Transactions on* , vol.14, no.3, pp.456-474, June 2010.

[18] Knowles, J.; Corne, D.; , "On metrics for comparing nondominated sets," *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on* , vol.1, no., pp.711-716, 12-17 May 2002

[19] Van Veldhuizen, D.A.; Lamont, G.B.; , "On measuring multiobjective evolutionary algorithm performance," *Evolutionary Computation, 2000. Proceedings of the 2000 Congress on* , vol.1, no., pp.204-211 vol.1, 2000

[20] Anne Auger, Johannes Bader, Dimo Brockhoff, and Eckart Zzitzler. Theory of the Hypervolume Indicator: Optimal μ -Distributions and the Choice of the Reference Point. FOGA'09, January 9-11, 2009, Orlando, Florida, USA.

[21] Beume, N.; Fonseca, C.M.; Lopez-Ibanez, M.; Paquete, L.; Vahrenhold, J.; , "On the Complexity of Computing the Hypervolume Indicator," *Evolutionary Computation, IEEE Transactions on* , vol.13, no.5, pp.1075-1082, Oct. 2009

[22] Carlos M. Fonseca, Lu´ıs Paquete, and Manuel Lopez-Ibanez. An Improved Dimension-Sweep Algorithm for the Hypervolume Indicator. *2006 IEEE Congress on Evolutionary Computation Sheraton Vancouver Wall Centre Hotel, Vancouver, BC, Canada,* July 16-21, 2006

[23] Lyndon While, *Senior Member, IEEE*, Phil Hingston, *Member, IEEE*, Luigi Barone, *Member, IEEE*, and Simon Huband, *Member, IEEE.* A Faster Algorithm for Calculating Hypervolume. *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, VOL. 10, NO. 1*, FEBRUARY 2006

[24] Tobias Wagner , Tobias Wagner , Nicola Beume , Nicola Beume , Boris Naujoks , Boris Naujoks. Pareto-, Aggregation-, and Indicator-based Methods in Many-objective Optimization. *Proc. of EMO 2007, vol. 4403 of LNCS*

[25] Eckart Zitzler, Dimo Brockhoff, and Lothar Thiele. The Hypervolume Indicator Revisited: On The Design of Pareto-Compliant Indicators Via Weighted Integration. In *Evolutionary*

Multi-Criterion Optimization, Vol. 4403 (2007), pp. 862-876.

[26] Schaffer, J. D. (1984). *Multiple Objective Optimization with Vector Evaluated Genetic Algorithms.* Unpublished Ph.D. thesis, Vanderbilt University, Nashville, Tennessee.

[27] Fourman, M. P. (1985). Compaction of symbolic layout using genetic algorithms. In Grefenstette, J. J., editor, *Proceedings of an International Conference on Genetic Algorithms and Their Applications*, pages 141−153, sponsored by Texas Instruments and U.S. Navy Center for Applied Research in Artificial Intelligence (NCARAI).

[28] Kursawe, F. (1991). A variant of evolution strategies for vector optimization. In Schwefel, H.-P. and Manner, R., editors, *Parallel Problem Solving from Nature− Proceedings of the First Workshop PPSN*, pages 193−197, Springer, Berlin, Germany.

[29] Hajela, P. and Lin, C.-Y. (1992).Genetic search strategies in multicriterion optimal design. *Structural. Optimization*, 4:99−107.

[30] Fonseca, C. M. and Fleming, P. J. (1993). Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In Forrest, S., editor, *Proceedings of the Fifth International. Conference on Genetic Algorithms*, pages 416−423, Morgan Kaufmann, San Mateo, California.

[31] Horn, J., Nafpliotis, N. and Goldberg, D. E. (1994). A niched Pareto genetic algorithm for multiobjective optimization. In *Proceedings of the First IEEE Conference on Evolutionary Computation, IEEE. World Congress on Computational Computation,* Volume 1, pages 82−87, IEEE Press, Piscataway, New Jersey.

[32] Srinivas, N. and Deb, K. (1994).Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2(3):221−248.

[33] Van Veldhuizen, D. A. and Lamont, G. B. (1998a). Evolutionary computation and convergence to a Pareto front. In Koza, J. R., Banzhaf, W., Chellapilla, K., Deb, K., Dorigo, M., Fogel, D. B., Garzon, M. H., Goldberg, D. E., Iba, H. and Riolo, R., editors, *Genetic*

Programming 1998: Proceedings of the Third Annual Conference, pages 22−25, Morgan Kaufmann, San Francisco, California.

[34] Rudolph,G. (1998).On a multi-objective evolutionary algorithm and its convergence to the pareto set. In IEEE *International Conference on Evolutionary Computation (ICEC' 98)*, pages 511−516, IEEE. Press, Piscataway, New Jersey.

[35] Obayashi, S., Takahashi, S. and Takeguchi, Y. (1998).Niching and elitist models for mogas. In Eiben,A. E., B ¨ ack, T., Schoenauer, M. and Schwefel, H.-P., editors, *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, pages 260−269, Springer, Berlin, Germany.

[36] Parks, G. T. and Miller, I. (1998). Selective breeding in a multiobjective genetic algorithm. In Eiben,A. E., B ¨ ack, T., Schoenauer, M. and Schwefel, H.-P., editors, *Fifth International Conference on Parallel Problem Solving from Nature (PPSN-V)*, pages 250−259, Springer, Berlin, Germany.

[37] Laumanns, M., Rudolph, G. and Schwefel, H.-P. (1998). A spatial predator-prey approach to multiobjective optimization: A preliminary study. In Eiben, A. E., B ¨ ack, T., Schoenauer, M. and Schwefel, H.-P., editors, *Fifth International Conference on Parallel Problem Solving from Nature(PPSN-V)*, pages 241−249, Springer, Berlin, Germany.

[38] Zitzler, E. and Thiele, L. (1999). Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach. *IEEE Transactions on Evolutionary Computation*, 3(4):257−271.

[39] Kalyanmoy Deb. Multi-Objective Optimization Using Evolutionary Algorithms. *John Wiley & Sons, Inc*. New York, NY, USA

[40] M. Srinivas and L. Patnaik, “Adaptive probabilities of crossover and mutation in genetic algorithms,” *IEEE Trans. Syst., Man, and Cybernetics*, vol. 24, no. 4, pp. 656-667, Apr. 1994.

[41] William M. Spears. Adapting Crossover in Evolutionary Algorithms. *Proceedings of the Fourth Annual Conference on Evolutionary Programming*, 1995.

VITA

ZHENAN HE

Candidate for the Degree of

Master of Science

Thesis: PERFORMANCE METRICS ENSEMBLE FOR MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Major Field: Electrical Engineering

Biographical:

Education:

Completed the requirements for the Master of Science in Electrical Engineering at Oklahoma State University, Stillwater, Oklahoma in May, 2011.

Completed the requirements for the Bachelor of Engineering in Automation at University of Science and Technology Beijing, Beijing, China in 2008.

Experience:

Assistant Engineer, Function Group, China

Assistant, Rosenberger Asia Pacific Electronic Co., Ltd, China

Professional Memberships:

IEEE Computational Intelligence Society

Name: ZHENAN HE
Date of Degree: May, 2011.

Institution: Oklahoma State University
Location: Stillwater, Oklahoma

Title of Study: PERFORMANCE METRICS ENSEMBLE FOR MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS

Pages in Study: 95
Candidate for the Degree of Master of Science

Major Field: Electrical Engineering

Findings and Conclusions:

There are five types of unary performance metrics and two types of binary performance metrics. However, no single metric can faithfully measure MOEA performance. Moreover, every metric has its unique character; no metrics can substitute others completely. An ensemble method is introduced to compare EAs by combining a large number of single metrics using modified Double Tournament Selection. Double Tournament Selection can maximum protects the qualified individual from being lost by some stochastic factors in a comparison time. This ensures the final result is the really best one and the whole ensemble process is effective and precise. Therefore, performance metrics ensemble can overcome the lost information problem by the single metric which only provides some specific but limited information. Furthermore, ensemble method avoids the choosing process which is a heavy computational process and can be directly used to assessing EAs. Finally, from the experiment results by using performance metrics ensemble, Each MOEA's characteristic is summarized.

ADVISER'S APPROVAL: Gary G. Yen