Multi-Objective Differential Evolution with Multi-Noisy Random Vectors (mnv-MODE) for the Solution of Many-Objective Optimization Problems

Mohamed Abdulqader Bayazid

Submitted to the
Institute of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Engineering

Eastern Mediterranean University
September 2019
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Acting Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science in Computer Engineering.

Prof. Dr. Işık Aybay
Chair, Department of Computer Engineering

We certify that we have read this thesis and that in our opinion it is fully adequate in scope and quality as a thesis for the degree of Master of Science in Computer Engineering.

Asst. Prof. Dr. Ahmet Ünveren
Supervisor

Examining Committee

1. Assoc. Prof. Dr. Adnan Acan
2. Asst. Prof. Dr. Zehra Borataş Şensoy
3. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

The ubiquity of multi-objective optimization problems (MOOPs) in real life attracted the attention of many scientists during the last two decades and motivated them to do a large amount of research in multi-objective evolutionary algorithms (MOEAs) which are broadly used in solving MOOPs. However, no algorithm can be considered as the universal optimizer for MOOPs. In this dissertation, multi-objective differential evolution (MODE) is used to develop a new approach called mnv-MODE which aims to solve ZDT1-ZDT4, ZDT6, UF1-UF10 and MaOP1-MaOP10 benchmark problems with 2, 3 and 5 objectives. Four different versions of the proposed algorithm are introduced by modifying MODE and using a local search. Compared to other MOEAs, the results show that our proposed mnv-MODE versions (especially version 4) have the best IGD values on the majority of test instances. This means that mnv-MODE achieved better performance than some efficient algorithms such as SPEA2, MOEA/D and NSGA- II for the solved test Problems.

Keywords: Multi-objective optimization problems, multi-objective evolutionary algorithms, multi-objective differential evolution.

ÖZ

Gerçek hayattaki çok amaçlı optimizasyon problemlerinin (ÇAOP) yaygınlığı, son yirmi yıl boyunca birçok bilim adamının dikkatini çekti ve bunları çözmek için geniş çapta kullanılan çok amaçlı evrimsel algoritmalar (ÇAEA) konusunda büyük miktarda araştırma yapmaya teşvik etti. Bununla birlikte, hiçbir algoritma ÇAOP'ler için evrensel optimizer olarak kabul edilemez. Bu tezde, ZDT1-ZDT4, ZDT6, UF1-UF10 ve MaOP1-MaOP10 kıyaslama problemlerini 2, 3 ve 5 hedefleriyle çözmeyi amaçlayan mnv-MODE adlı yeni bir yaklaşım geliştirmek için çok amaçlı diferansiyel evrim algoritması (MODE) kullanılmıştır. Önerilen algoritmanın dört farklı sürümü MODE değiştirilerek ve yerel bir arama kullanılarak oluşturuldu. Diğer ÇAOP'larla karşılaştırıldığında, sonuçlar, önerilen mnv-MODE sürümlerimizin (özellikle sürüm 4), test örneklerinin çoğunda en iyi IGD değerlerine sahip olduğunu göstermektedir. mnv-MODE'nin, çok amaçlı test Problemlerinde SPEA2, MOEA / D ve NSGA-II gibi bazı etkili algoritmalardan daha iyi performans elde ettiği gösterilmiştir.

Anahatar Kelimeler: Çok amaçlı optimizasyon problemleri, çok amaçlı evrimsel algoritmalar, çok-amaçlı diferansiyel evrim.

To my parents with love

ACKNOWLEDGMENT

First and foremost, praises and thanks to the God, the Almighty, for His showers of blessings throughout my research work to complete the research successfully.

I would like to express my deep and sincere gratitude to my research supervisor Asst. Prof. Dr. Ahmet Ünveren for giving me the opportunity to do research and providing invaluable guidance throughout this research. His dynamism, vision, sincerity and motivation have deeply inspired me. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance. I am extremely grateful for what he has offered me. I would also like to thank him for his empathy, and great sense of humor.

I am extremely grateful to my parents for their love, prayers, caring and sacrifices for educating and preparing me for my future. I am very much thankful for continuing support to complete this research work. Also I express my thanks to my sisters and brother in law for their support and valuable prayers. My Special thanks goes to them for the keen interest shown to complete this thesis successfully.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

TABLE OF CONTENTS

LISTS OF TABLES

LISTS OF FIGURES

LIST OF ABBREVIATIONS

AMGA	Archive-based Micro Genetic Algorithm
CEC	Congress on Evolutionary Computation
COP	combinatorial optimization problem
DE	Differential evolution
DECMOSA-SQP	Differential Evolution with Self-adaptation and Local Search for Constrained Multiobjective Optimization
EA	Evolutionary algorithm
EMO	Evolutionary multi-objective optimization
ER	Error Ratio
GA	Genetic algorithm
GD	Generational Distance
GDE3	Generalized Differential Evolution 3
HV	Hypervolume
IGD	Inverted Generational Distance
MODE	Multi-objective differential evolution
MOEA	Multi-objective evolutionary algorithm
MOEA/D	MOEA based on decomposition
MOEA/D-GM	MOEA/D with Guided Mutation
MOEP	Multi-objective evolutionary programming (MOEP)
MOOP	Multi- objective optimization problem
MTS	Multiple Trajectory Search
NSGA	Non-dominated sorting genetic

OWMOSaDE	Multiobjective Self-adaptive Differential Evolution algorithm with objective wise learning strategies
PDE	Pareto-frontier Differential Evolution
PDEA	Pareto Differential Evolution Approach
PS	Pareto set
PSO	Particle swarm optimization
SOOP	Single objective optimization problem
SPEA	Strength Pareto Evolutionary Algorithm
TS	Tabu search

Chapter 1

INTRODUCTION

Optimization is a subfield of applied mathematics and numerical analysis which is important in numerous disciplines and domains such as Engineering, Science, Economics and many other fields. Almost every problem can be expressed as an optimization problem in which one (or more) objective function(s) is defined. An objective function is a function that is used as a measurement to determine the quality of extracted solutions. The goal of an optimization task is to discover the optimal solution (maximum or minimum value, subject to the problem) of an objective function [1, 2, 3].

Depending on the number of objective functions, optimization problems can be classified either as single objective optimization problem (SOOP) or as multi-objective optimization problem (**MOOP).** A single objective optimization problem has one objective function and a unique optimal solution. In contrast, MOOPs have two or more objective functions that are usually contradicting and instead of one optimum, tradeoffs (conflicting scenarios) and a set of alternative solutions with equivalent quality may be extracted. For instance, before buying a new car, the customer may have several desires which are clearly conflicting (competing) such as minimum price, maximum speed, minimum fuel consumption and maximum amount of luxury. Consequently, the solution which is optimal with regard to only one objective is not selected as an optimal solution for a MOOP [4, 5, 6].

One of the important principles that is frequently used to solve multi-objective optimization problems is domination (Pareto optimality) concept. To illustrate this concept, we can say that a vector $\overrightarrow{x_1}$ dominates a vector $\overrightarrow{x_2}$ only if $\overrightarrow{x_1}$ is at least as good as $\overrightarrow{x_2}$ for all objectives and $\overrightarrow{x_1}$ is strictly better than $\overrightarrow{x_2}$ for at least one objective. Using this concept, each solution is compared with the other solutions in the search space and the solutions can be divided into 3 categories as shown in Figure 1:

1) Solutions dominate others.
2) Solutions dominated by others.
3) Solutions indifferent to the rest of solution (neither dominated nor dominating).

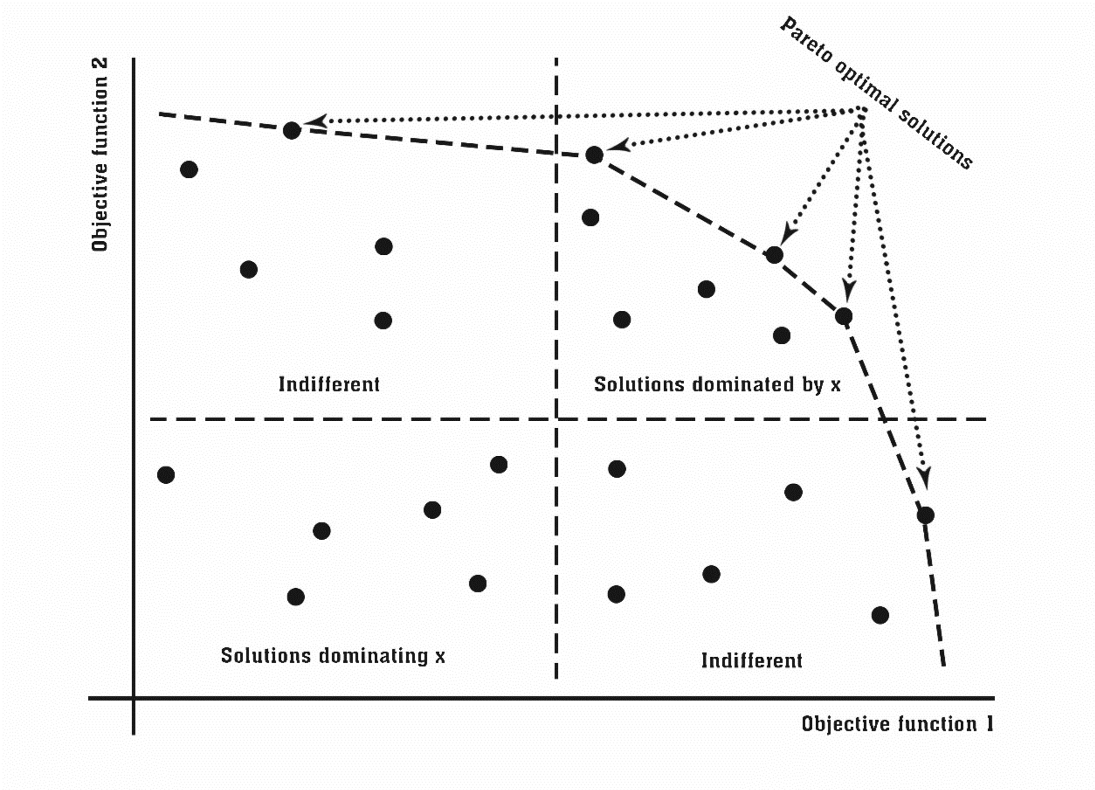


Figure 1: Pareto optimality concept [7].

Each solution which is not dominated by any other solution in the search space is called Pareto optimal solution(or non-dominated solution) and the whole set of such Pareto optimal solutions is known as Pareto optimal set (or non-dominated set) [2, 3, 7].

Unlike single-objective optimization problem, MOOP has two search spaces instead of one. In addition to decision variable space that exists in single objective optimization problem, there is another space called objective space in MOOP. Each solution in the decision variable space is represented by a point in the objective space [2]. The curve that is created by connecting the non-dominated points in the objective space is called Pareto-front [8].

Moreover, in place of one goal which is finding the optimum in SOOPs, the following two different goals must be achieved to solve a MOOP:

1) Convergence to the Pareto front.
2) Maintenance of the maximum diversity among solutions on the non-dominated front.[4]

Evolutionary algorithms (EAs) are stochastic search methods used to emulate the process of natural evolution of species [9]. EAs are characterized by the following three basic properties:

1) Population-based: use a set of solutions (a population) to generate a new population in each iteration [4, 10]. Each member (solution) in the population is known as an individual.
2) Fitness-oriented: each individual has its own fitness value and the fittest individuals are preferred by EAs.

3) Variation–driven: Many variation processes will be applied on individuals [10].

EAs are broadly used and preferred among optimization techniques due to the following reasons:

1) No derivative information is required by EA
2) They are flexible, easy to implement and have numerous applications [6].

Multi-objective evolutionary algorithms (MOEAs) which are first introduced by David Schaffer in 1984 are extended versions of EAs deal with MOOPs and this type of optimization is known as evolutionary multi-objective optimization (EMO) [9,10]. The difference between MOEA and single-objective EA can be considered in two main topics: selection mechanism and diversity maintenance mechanism. MOEAs aim to find a group of non-dominated solutions and avoid convergence between them as well [11].

Differential evolution (DE) is a population-based, parallel directed search algorithm (proposed by Rainer Storn and Kenneth Price) basically based on the idea of using the difference between vectors to mutate (perturbate) the vector population. It is considered as an enhanced model of Genetic algorithms (GA), another algorithm has been suggested before DE by Holland (1975), since both algorithms have mutation, crossover and selection. While simple GA uses binary numbers to represent problem parameters, DE utilizes real-valued parameters principally to solve continuous optimization problems. DE is also characterized by its simple structure, robustness, speed and its simplicity to implement. Ten different strategies of DE are suggested by Price and Storn. These strategies differ rely on 3 factors: number of difference vectors considered for perturbation, the type of crossover and the type of vector that will be perturbed. By extending DE and convert it from single –objective optimization to Multi-objective optimization, multi-objective differential evolution (MODE) approach is proposed. MODE approach consists of 3 essential parts: Reproduction, Pareto-based approach, and selection [12, 13]

Local search is a random search method integrated in several EAs in order to get the maximum overall efficiency of the algorithm [14]. Numerous algorithms can be classified as local search-based methods. These algorithms include iterated local search, simulated annealing, guided local search, tabu search and variable neighborhood search [1].Among several local search methods, tabu search (TS) is a popular one introduced by Fred Glover and it is characterized by using memories in order to get rid of local optima and to diversify the search [8].

1.1 Problem Statement

Because of their ubiquity in our real life, Multi-objective optimization problems have attracted the attention of many researchers around the world and a surge of researches have been done in this area in the last twenty years. The crucial role that Multi-objective evolutionary algorithms plays in solving different MOOPs has been becoming very clear over the years. Up to now, numerous MOEAs (including the most popular ones currently such as NSGA- II, SPEA2 and MOEA/D) have been introduced for dealing with various MOOPs [20, 21]. These MOEAs were developed by means of various population-based meta-heuristics like genetic algorithms, differential evolution and particle swarm optimization [15].

In spite of the success, to some extent, that achieved by MOEAs in solving various multi-objective optimization problems (MOOPs), the existing approaches in the literature vary a lot in regard to their solutions and the technique used to compare their best results with other current algorithms. To put in other words, no algorithm can be considered as the universal optimizer for those kind of problem till now [16]. This motivates researchers to devote much work, efforts and time in order to develop other good approaches to solve MOOPs.

1.2 Aim of the Study

This thesis seeks to define a new approach in order to solve several multi-objective benchmark optimization problems. During past years, Hybridizing MOEAs with local search have shown better performance than using MOEAs alone [17]. Several hybrid approaches in which local search is incorporated with MOEAs have been developed in order to obtain better convergence to the Pareto front [18]. In our proposed method, Multi-objective differential evolution (MODE) is combined with a local search algorithm called Tabu search (TS) to produce a new hybrid approach. The problems which this thesis aims to solve are ZDT1, ZDT2, ZDT3, ZDT4, ZDT6, UF1-UF10, and MaOP1-MaOP10. They will be solved using 2, 3 and 5 objectives and the results will be compared with the algorithm given in CEC2018.

1.3 Significance of the Study

Multi-objective optimization is beneficial in a wide variety of domains and fields such as medicine, computer science, electrical engineering, design, management, chemistry, physics and many other areas [19]. Multi-objective optimization gained its significance from its ubiquitous applications in our life since most of the real-world problems existing today require concurrent optimization of many competing objectives [2]. During the past few decades, MOEAs have shown its good performance in solving that kind of problems [20]. For this reason, a huge amount of research has been done in this scientific area [19]. Nevertheless, some interesting issues related with this subject remain open and much work is needed in this regard.

1.4 Thesis Structure

This thesis will be organized as follows: chapter 2 will provide general information about DE, MODE, TS and benchmark problems. Some of the techniques that used before to solve benchmark problems are also summarized in the literature review chapter. After that, our proposed method will be explained in details in chapter 3. Next chapter will discuss the experimental results. Finally, the conclusion will be given in chapter 5.

Chapter 2

LITERATURE REVIEW

2.1 Overview

Global search and optimization methods are divided into 3 groups: enumerative, deterministic and stochastic. Stochastic methods, such as Evolutionary algorithms (EAs) (like GA,DE and PSO),TS, and SA, were developed as an alternative approach for solving irregular problems(some NP-complete or high dimensional problems).This is due to the fact that several real world MOOPs in science and engineering disciplines are irregular and enumerative and deterministic search methods are unsuitable for solving these problems. Some measurement instruments called metrics are needed to evaluate your algorithm [8, 14]. General information about Differential evolution (DE), MODE (an enhanced version of DE), Tabu search (TS) and metrics will be provided in this chapter. This chapter will also refers to some benchmark problems and the previous methods used to solve them.

2.2 Differential Evolution

More than 2 decades ago, a simple, fast, robust and parallel direct search algorithm has emerged to deal with continuous optimization problems. The algorithm which introduced by Price and Storn is similar with previous EAs in that both of them are using a predefined set of operators (such as mutation and recombination) to imitate the evolution of a population (set of individuals). Nevertheless, the construction of the changing operator is the essential difference between previous EAs and DE. DE can be considered as an improved model of GAs. In comparison with GAs, DE uses realcoded values instead of binary numbers to represent problems. Moreover, while crossover is executed before mutation in GAs, the inverse order is applied in DE [1, 8, 12, 13, 22]. The working principle of DE is illustrated in Figure 2.

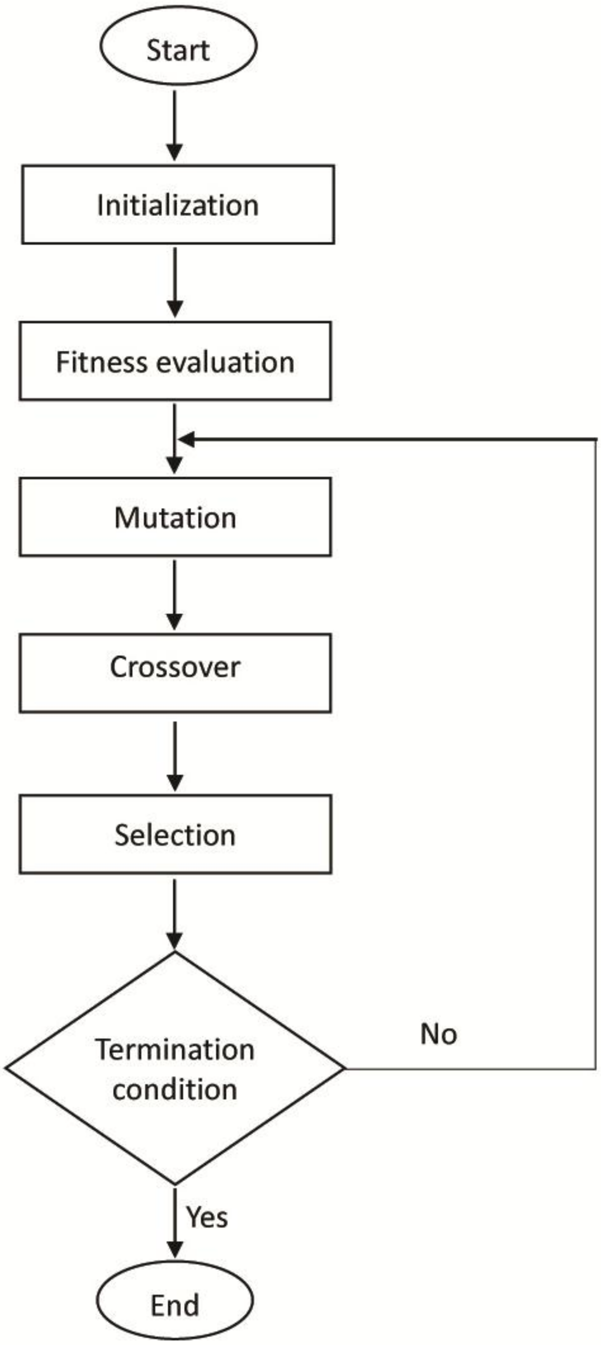


Figure 2: flowchart of DE [23]

2.2.1 Initialization

Similar to other EAs, number of **Individuals** $\boldsymbol{N_p}$ (vectors of solutions known also as **Agents** in DE) and the parameters of lower and upper bounds of each solution vector should be specified before executing the algorithm. Each agent in DE is initialized according to the following equation:

$$\boldsymbol{X_i} = X^L + \boldsymbol{p_i}\,(X^U - X^L)\text{ , i=1,2,…. }\boldsymbol{N_p} \tag{1}$$

Where $X^L\ and\ X^U$ denote the lower and upper bounds of parameters and $\boldsymbol{p_i}$ refer to a vector of numbers generated randomly in the range between 0 and 1 [24].

2.2.2 Mutation

Mutation is defined in numerous dictionaries as a change or alternation. Unlike other EAs, Mutation in DE rely on the idea of applying differences between parent vectors (target vectors) to obtain a mutant (donor) vector $\boldsymbol{V_{i,G}}$ corresponding to each agent $\boldsymbol{X_{i,G}}$**.** Various mutation strategies in DE has implemented but the following five are more frequently used than others:

1) DE/rand/1: $$\boldsymbol{V_{i,G}} = \boldsymbol{X_{r_1^i,G}} + \mathbf{F.}(\boldsymbol{X_{r_2^i,G}} - \boldsymbol{X_{r_3^i,G}}) \tag{2}$$
2) DE/best/1: $$\boldsymbol{V_{i,G}} = \boldsymbol{X_{best,G}} + \mathbf{F.}(\boldsymbol{X_{r_1^i,G}} - \boldsymbol{X_{r_2^i,G}}) \tag{3}$$
3) DE/rand-to-best/1:
$$\boldsymbol{V_{i,G}} = \boldsymbol{X_{i,G}} + \mathbf{F.}\,(\boldsymbol{X_{best,G}} - \boldsymbol{X_{i,G}}) + \mathbf{F.}(\boldsymbol{X_{r_1^i,G}} - \boldsymbol{X_{r_2^i,G}}) \tag{4}$$
4) DE/best/2: $$\boldsymbol{V_{i,G}} = \boldsymbol{X_{best,G}} + \mathbf{F.}(\boldsymbol{X_{r_1^i,G}} - \boldsymbol{X_{r_2^i,G}}) + \mathbf{F.}(\boldsymbol{X_{r_3^i,G}} - \boldsymbol{X_{r_4^i,G}}) \tag{5}$$
5) DE/rand/2: $$\boldsymbol{V_{i,G}} = \boldsymbol{X_{r_1^i,G}} + \mathbf{F.}(\boldsymbol{X_{r_2^i,G}} - \boldsymbol{X_{r_3^i,G}}) + \mathbf{F.}(\boldsymbol{X_{r_4^i,G}} - \boldsymbol{X_{r_5^i,G}}) \tag{6}$$

Where: $r_1^i, r_2^i, r_3^i, r_4^i, r_5^i$ are mutually exclusive integers selected randomly in the range between 1 and $\boldsymbol{N_p}$ and they are different from the index *i*, *F* is a positive control parameter called scaling factor and normally chosen between 0.2 and 1, and $X_{best,G}$ is the best agent vector with the best objective function in the population at generation G [25, 26, 27].

2.2.3 Crossover

After creating the mutant vector via mutation stage, a crossover operator is utilized in DE to produce an offspring $\boldsymbol{U_{i,G}}$ called **trail vector** by recombining the donor vector $\boldsymbol{V_{i,G}}$ with the target vector $\boldsymbol{X_{i,G}}$ in order to increase the diversity of the population. In DE, two types of crossover are broadly used: Binomial crossover and Exponential crossover [26]. Mathematical formulas of exponential and binomial crossover will be shown in figure 3.

In exponential crossover, two integer numbers n (which refer to the starting point in the target vector) and L (which indicates the number of components the target vector takes from the mutant vector) should be chosen randomly from the numbers in the interval [1,D] [26].

In binomial crossover, the trial vector component will equal to its corresponding component from the donor vector if a random number between 0 and 1 is less than Crossover rate CR or j=$\boldsymbol{j_{rand}}$**.** In different circumstances, the value of it will be identical to the value of its corresponding component value from target vector [25, 26].

2.2.4 Selection

In this step, each trial vector $\boldsymbol{U_{i,G}}$ compared with its corresponding target vector $\boldsymbol{X_{i,G}}$ according to their objective functions to determine which one of them will survive to the next generation G+1. If the fitness value of trial vector is the best, the corresponding target vector will be replaced by the trial vector. Otherwise, no change will happen and the trial vector will be eliminated. Equation (7) shows the formula of selection operation:[26, 27]

$$\boldsymbol{X_{i,G}} = \begin{cases} \boldsymbol{U_{i,G}} & \text{if } \mathrm{f}(U_{i,G}) \leq \mathrm{f}(X_{i,G}) \\ \boldsymbol{X_{i,G}} & \text{otherwise} \end{cases} \tag{7}$$

Step 1: set the generation number $G = 0$, and randomly initialize a population of *NP* individuals $P_G = \{X_{1,G,}, \ldots, X_{NP,G}\}$
with $X_{i,G} = \{x_{i,G}^1, \ldots, x_{i,G}^D\}, i = 1, \ldots, NP$ uniformly distributed in the range $[X_{min}, X_{max}]$, where
$X_{min} = \{x_{min}^1, \ldots, x_{min}^D\}$ and $X_{max} = \{x_{max}^1, \ldots, x_{max}^D\}$

Step 2: WHILE stopping criterion is not satisfied
DO
Step 2.1 Mutation step
/* Generate a mutated vector $V_{i,G} = \{v_{i,G}^1, \ldots, v_{i,G}^D\}$ for each target vector $X_{i,G}$ */
FOR *i*=1 to *NP*
Generate a mutated vector $V_{i,G} = \{v_{i,G}^1, \ldots, v_{i,G}^D\}$ corresponding to $X_{i,G}$ via one of the equations (2)-(6).
END FOR
Step 2.2 Crossover step
/* Generate a trial vector $U_{i,G} = \{u_{i,G}^1, \ldots, u_{i,G}^D\}$ for each target vector $X_{i,G}$ */
a) Binomial crossover
FOR *i*=1 to *NP*
$j_{rand} = rand[1, D]$
FOR *j*=1 to *D*
$$u_{i,G}^j = \begin{cases} v_{i,G}^i, & if\ (rand[0,1] \leq CR)\ or\ (j = j_{rand}) \\ x_{i,G}^i, & otherwise \end{cases}$$
END FOR
END FOR
b) Exponential crossover
FOR *i*=1 to *NP*
$j_{rand} = rand[1, D], L = 0$
$U_{i,G} = X_{i,G}$
DO
$u_{i,G}^j = v_{i,G}^i$
$j = (j+1)_{D*}$
L=*L*+1
WHILE $(rand[0,1] < CR \,\&\, L < D)$
END FOR
Step 2.3 Selection step
FOR *i*=1 to *NP*
Evaluate the trial vector $U_{i,G}$
IF $f(U_{i,G}) \leq f(X_{i,G})$, THEN $X_{i,G+1} = U_{i,G}$, $f(X_{i,G+1}) = f(U_{i,G})$
IF $f(U_{i,G}) \leq f(X_{best,G})$, THEN $X_{best,G} = U_{i,G}$, $f(X_{best,G}) = f(U_{i,G})$
END IF
END IF
END FOR
Step 2.4 Increment the generation count *G*=*G*+1

Step3 END WHILE

*Acute brackets denote modulo function with modulus D

Figure 3: Pseudo code of DE [25].

2.2.5 Strategies of DE

Although many variants of differential evolution are existed, all of them are following the convention DE/x/y/z where x represents the target vector, y refers to how many difference vectors should be considered for mutation, and z which indicates the kind of crossover that is used. The name of DE strategy will be modified if one of the parameters x, y and z is changed. According to that notation, Price and Storn introduce the following DE strategies:

1) DE/rand/1/bin
2) DE/rand/1/exp
3) DE/best//1/bin
4) DE/best//1/exp
5) DE/rand//2/bin
6) DE/rand//2/exp
7) DE/best//2/bin
8) DE/best//2/exp
9) DE/rand-to-best//1/bin
10) DE/rand-to-best//1/exp

Where the first strategy mentioned above is considered as the standard one in DE method. [28, 29]

2.3 Multi-Objective Differential Evolution

In order to extend DE to solve multi-objective optimization problems, two main issues should be taken in consideration:

1) Like any MOEA, the method should be used to increase the diversity between non-dominated solutions is very important aspect. Two types of metrics are mostly used to promote diversity: Crowding distance and Fitness sharing.

2) The case in which the parent will be replaced by the candidate solution [22].

Up to now, several MOEAs such as PDE, PDEA and MODE have been developed by extending DE to handle MOOPs. The first one was Pareto-frontier Differential Evolution (PDE) which introduced by Abbass et al in 2002. In this approach, DE is used to generate new individuals but only the non-dominated ones are kept to be used in the next generation (dominant individuals are eliminated iteratively). PDE algorithm has shown better performance than SPEA. After that, Madavan propose another algorithm known as Pareto Differential Evolution Approach (PDEA). In this approach, DE is also applied to generate new solutions. Once the new population is generated, both new and current populations are combined and subsequently the non-dominated rank and diversity rank for each agent are calculated. Depending on these 2 ranks, the best individuals are chosen in the efficient variant of PDEA (since there are two variants of PDEA: one of them is inefficient) [30].

Subsequently, Xue et al proposed another algorithm in their paper “Pareto-based multi-objective differential evolution” and referred to that algorithm as MODE. PDEA and MODE are similar in that both of them utilize crowding distance metric and Pareto-based ranking assignment. However, the manner in which Pareto-based ranking and crowding distance are applied in MODE is different than the manner used in PDEA. In MODE, the fitness value of each individual is firstly calculated (by employing pareto-based approach) and then decreased with regard to the crowding distance of the individual. According to these fitness values, the best agents are selected to be used in the next generation [31]. In 2005, Babu et al developed another algorithm which referred also as MODE in the paper “Multi-objective Optimization Using Differential evolution” [32].

2.4 Local Search

There are local search algorithms which are used widely to solve hard optimization problems. By searching within the solution space, LS algorithms try to detect high-quality solutions. Beginning with a starting solution, a new solution, which is near the current one, is generated iteratively. The solutions which are near the current one are defined by a neighborhood function. LS method is the basis of some successful heuristics such as simulating annealing and tabu search.[33,34]

2.4.1 Tabu Search

Tabu Search (TS), which can be considered as an extension of hill climbing techniques, is a single-objective local search algorithm proposed initially to solve large COPs and extended later to solve continuous optimization problems [1]. Nowadays, a wide variety of TS applications are existing in several branches of knowledge such as design, telecommunications, technology, routing, scheduling, and artificial intelligence [35].

In comparison with conventional hill climbing methods, TS utilizes memories to dispose of being stuck in local optima by allowing non-improving moves. Short-term, Intermediate term and long term memories are different kinds of memories used in TS. While diversification of the search is the job of the long-term memory and intensification of the search is the task of the intermediate-term memory, the short-term memory (called Tabu list) is employed to prohibit cycling (going back to visited solutions).[8]

In addition to the termination criteria, tabu search has two significant criteria: Tabu criteria and aspiration criteria. Tabu criteria is responsible for classifying the solutions which are visited by the algorithm once as tabu (i.e. banned) and prevent visiting those solutions again for a predetermined number of iterations (this number is known as tabu tenure). Nonetheless, tabu criteria are sometimes too powerful and may forbid visiting good solutions which are unvisited before. For this reason, a criterion which allow solutions better than current one to be selected even if they are prevented by tabu criterion is emerged. This criterion is called Aspiration criterion and it is considered as the simplest criterion that is usually used in many TS implementation. Other complicated aspiration criteria which are rarely used are also available. The task of any aspiration criteria is to enable the algorithm to break the tabu restrictions only if a certain condition is satisfied [1, 35, 36]. Figure 4 shows the pseudo code of tabu search.

1. Set $t = 0$.
2. Generate an initial solution x.
3. Initialize the tabu lists $T \leftarrow \emptyset$ and the size of tabu list L.
4. **Repeat**:
    a. Set the candidate set $A\,(x,t) = \{\, x' \in N(x)\ \backslash T\,(x,t) \cup \check{T}(x,t)\}$.
    b. Find the best x from $A\,(x,t)$: Set $x' = \arg min_{y \in A(x,t)} f(y)$.
    c. If $f(x')$ is better than $f(x)$, $x \leftarrow x'$.
    d. Update the tabu lists and the aspiration criteria.
    e. If the tabu list T is full, then old features from T are replaced.
    f. Set $t = t + 1$.

    Until termination criterion is satisfied.

Figure 4: Pseudo code of Tabu search [1].

2.5 Benchmark Problems

Benchmark problems are broadly-used test functions designed to evaluate single-objective and multi-objective EAs [37]. They play a significant role in identifying the weakness and strength points of EAs [38]. Nowadays, several benchmark MOOP suites are existing. Depending on what you want to test, the suitable benchmark test suite is chosen. Efficacy (quality), reliability (success rate) and efficiency (speed) are three basic factors which are normally considered [10]. This thesis aims to solve 25 benchmark problems. These problems belong to 3 different test suites: ZDT, UF and MaOP. Nonetheless, ZDT, UF and MaOP problems are similar in that the aim in all of them is minimizing all of the objective functions.

2.5.1 ZDT Benchmarks

This suite, which constructed by Zitzler et al., consists of six different test problems frequently used to evaluate MOEAs. All these problems except ZDT5, which uses binary encoding, are real-valued problems. In contrast to the rest of ZDT problems in which the Pareto front is continuous, ZDT3 possess disconnected Pareto front. All ZDT problems are bi-objective and they cannot deal with more than 2 objectives [39, 40, 41].

The main characteristics and the search space of ZDT problems except the omitted one (ZDT5) are provided in table 1 where n refers to the number of decision variables. Among the problems in table 1, it is obvious that only ZDT4 has a distinct search space. While the domain of the first parameter in ZDT4 is [0, 1], the rest of the parameters of the same problem have the domain [-5, 5].

Table 1: ZDT problems [40, 41].

Problem name	Parameter domains	characteristics
ZDT1	[0,1]	Convex, uni-modal
ZDT2	[0,1]	Concave, multi-modal
ZDT3	[0,1]	Disconnected, multimodal/unimodal
ZDT4	$[0,1][-5,5]^{n-1}$	Convex, multimodal/unimodal
ZDT6	[0,1]	Concave, multimodal

Two arguments in favor of using ZDT test functions are:

1) They have well defined Pareto-optimal front
2) Numerous research papers which contains test results of these problems are widespread [41].

2.5.2 UF Benchmarks

This test suite consist of ten different problems were involved in the CEC 2009 competition and called unconstrained (bound constrained) problems. These problems, which are characterized by their complicated Pareto set and referred to as UF1-UF10, were built by Zhang et al. Depending on the number of objective functions, UF test suite can be classified into 2 categories. The first group contains the first seven problem UF1-UF7 which are bi-objective problems. The other group is composed of the rest of UF problems (UF8-UF10) and each member in this group has three objectives [42, 16].

UF problems can be also categorized into four groups on the basis of the search space of the problem. The first group contains the problems UF1, UF2, UF5, UF6 and UF7. The domain of first parameter of these problems is [0, 1] while the other parameters of each one of these problem possess the domain [-1, 1]. UF8, UF9 and UF10 are the members of the second group. The first two parameters of each of these problems have the domain [0, 1] whereas the domain of the remaining parameters is [-2, 2]. The third group have only one problem which is UF4. The domain of UF4 parameters except the first parameter (whose domain is [0, 1]) is [-2, 2]. UF3, in which each parameter has the domain [0, 1], is the only member in the fourth group. Table 2 presents the search space and main characteristics of UF problems [42].

Table 2: UF problems [42]

Problem	Search space	characteristics
UF1	$[0,1][-1,1]^{n-1}$	Convex, Multimodal
UF2	$[0,1][-1,1]^{n-1}$	Convex, Multimodal
UF3	$[0,1]^{n}$	Convex, Multimodal
UF4	$[0,1][-2,2]^{n-1}$	Concave, Multimodal
UF5	$[0,1][-1,1]^{n-1}$	Linear, Multimodal
UF6	$[0,1][-1,1]^{n-1}$	Linear, Disconnected, Multimodal
UF7	$[0,1][-1,1]^{n-1}$	Linear, Multimodal
UF8	$[0,1]^{2}[-2,2]^{n-2}$	Concave, Multimodal
UF9	$[0,1]^{2}[-2,2]^{n-2}$	Linear, Disconnected, Multimodal
UF10	$[0,1]^{2}[-2,2]^{n-2}$	Concave, Multimodal

2.5.3 MaOP Benchmarks

Ten various test problems are the members of this set of benchmark problems. MaOP problems are many objective problems which means that they can be applied with two or more objectives. In contrast to ZDT and UF problems which can only deal with two or three objectives, MaOPs can possess more than three objectives. If the MaOP problem comprises difficult features like complicated Pareto set, disconnection, degeneracy, or bias, it will be more challenging. Each parameter of MaOP problems has the domain [0, 1] [43]. In addition to the search space of MaOPs, features of MaOP problems are elaborated in table 3.

Table 3: MaOPs characteristics [44]

problem	Search space	characteristics
MaOP1	$[0,1]^n$	inverse of simplex, multimodality ,objective scales
MaOP2	$[0,1]^n$	complicated PS
MaOP3- MaOP4	$[0,1]^n$	complicated PS, bias
MaOP5- MaOP6	$[0,1]^n$	degeneracy, complicated PS
MaOP7- MaOP10	$[0,1]^n$	local degeneracy, complicated PS

2.5.4 Related work

Several algorithms were used to solve these problems. Some of them are mentioned in this section and the result of these algorithms will be compared with the result of our proposed method in the experimental results chapter.

The idea of joining Pareto ranking approach (originally proposed by Goldberg) and fitness sharing to produce a new algorithm known as non-dominated sorting genetic algorithm NSGA was introduced by Sirnivas and Deb in 1994 [10]. According to the concept of domination, individuals in the population are ranked using pareto ranking and classified into several categories (members of a category have the same rank). All non-dominated individuals will be on the first front. After that, the same dummy fitness value is assigned to individuals that have the same rank. To maintain the diversity inside the population, Individuals in each category share their dummy fitness values using a sharing function approach [45, 46]. As time passed, it is found that NSGA suffer from the following three basic criticisms:

1) The non-dominated sorting in NSGA characterized by its high computational complexity.
2) It is non-elitist algorithm which means that some good solutions may be lost during the execution of the optimization process.
3) The sharing parameter involved in the sharing function method must be specified.

All the issues mentioned above were considered in a study named “A fast and Elitist Multiobjective Genetic Algorithm: NSGA- II”. NSGA- II is an enhanced version of NSGA proposed by Deb et al in 2002 [47]. In NSGA- II, the non-dominated sorting (Pareto ranking) procedure is improved in order to reduce the computational complexity [1]. Moreover, an elitism mechanism is added to NSGA- II so that the performance and convergence of the algorithm is enhanced. Furthermore, the sharing function method is substituted by a new method called crowded-comparison approach. Using crowded-comparison method to preserve diversity among individuals in the population, there is no need to specify any additional parameter [47].

Local Search Based Evolutionary Multi-objective Optimization Algorithm for Constrained and Unconstrained Problems, which use an extension of NSGA- II called NSGA-II-LS [18], was one of the accepted papers in CEC2009 MOEA competition.

NSGA- II was extended by Deb and Jain to introduce a new algorithm called NSGA -III. The new algorithm is efficient in solving MOOPs which possess 3 to 15 objectives. Unlike NSGA- II, the diversity between individuals in NSGA -III is preserved by adding some reference points which are well-spread and updated adaptively [48].

SPEA2 [49] is an elitist algorithm developed by Zitzler et al to avoid the weakness points of a former version called Strength Pareto Evolutionary Algorithm (SPEA). In addition to the regular population, SPEA [50] utilizes an external set (archive). In each iteration, all non-dominated individuals in the population are stored in the archive and the archive get rid of any dominated solutions. A clustering technique is used in SPEA to keep the archive size below a predefined value without changing the characteristics of the non-dominated front. A strength value is determined for each member of the archive. The strength value of an individual i is calculated by the formula $\frac{n}{N+1}$ where n represents the number of individuals dominated by i and N indicates the archive size. The summation of the strength values of all archive members that dominate an individual give us the fitness value of that individual in the regular population [49, 50].

SPEA and SPEA2 are similar in that both of them are elitist algorithms. SPEA and SPEA2 differs mainly in three issues:

1) The fitness assignment strategy in SPEA2 is improved.
2) A nearest neighbor density estimation technique is added in SPEA2 to guide the search more precisely.
3) The preservation of outer solutions is guaranteed in SPEA2 by an improved archive truncation technique. This is due to the fact that the clustering technique in SPEA may not be able to maintain the boundary solutions [49].

In spite of the fact that decomposition is one of the essential strategies in classical multi-objective optimization, most of the modern MOEAs (such as NSGA- II and SPEA2) doesn’t use this technique. Various approaches (like weighted sum approach, Tchebycheff approach and Boundary Intersection approach) are existing for the purpose of decomposition. In 2007, MOEA/D (MOEA based on decomposition) algorithm was suggested by Zhang and Li [51]. Using MOEA/D, the MOOP is decomposed to many scalar optimization subproblems and then solve them simultaneously [11]. This explains why diversity maintenance and fitness assignment can be easily handled in MOEA/D [51]. No subproblem in MOEA/D can be optimized without using information from its neighboring subproblems [11]. MOEA/D can deal with disparate scaled objectives by using objective normalization techniques. The performance of MOEA/D is either better than or similar to NSGA- II performance in relation to solution quality. While MOEA/D performs similarly to or better than NSGA- II with respect to solution quality, its computational complexity is lower than that of NSGA- II in each iteration [51].

Nowadays, numerous versions of MOEA/D, such as MOEA/D-DE and MOEA/D-GM, are available. By employing DE and polynomial mutation operators, MOEA/D was extended to MOEA/D-DE which developed by Li and Zhang [52]. Another extension of MOEA/D which won CEC2009 MOEA contest is MOEA/D-GM [53].

In addition to MOEA/D algorithm which was pointed out above, several MOEAs won the CEC2009 MOEA competition which aims to solve UF problems. The winner algorithms, which are mentioned in the CEC2009 MOEA contest final report [54], comprises GDE3 [55], DMOEA/DD [56], MTS [57], LiuLi Algorithm [58], OWMOSaDE [59], ClusteringMOEA [60], AMGA [61], MOEP [62] and DECMOSA-SQP [63]

2.6 Metrics

The performance of any multi-objective optimization method is measured using some measurement instruments called metrics [3]. Various kinds of metrics are available for that purpose and some of them will be described here. Before using any of these metrics, the standard pareto front (PF-true) should be known [64].

2.6.1 Error Ratio (ER)

To indicate how many vectors in the pareto front generated by the algorithm (PF-known) whose fitness functions are not elements of PF-true, ER is used [8]. According to the following formula, ER is calculated

$$\text{ER} \triangleq \frac{\sum_{i=1}^{n} e_i}{n}$$

Where n refers to how many vectors are represented in PF-known and e_i =1 unless the objective functions of vector i belong to PF-true [3].

2.6.2 Generational Distance (GD)

This metric utilized to indicate how much is the average Euclidean distance between elements of PF-known and the nearest members of PF-true [65]. GD is determined using the following mathematical expression:

$$\text{GD} = \frac{(\sum_{i=1}^{n} d_i^p)^{\frac{1}{p}}}{n}$$

Where p=2, d represents the Euclidean distance existed between each solution i and the nearest solution of PF-true, and n indicates how many elements are in PF-known [66, 3].

2.6.3 Inverted Generational Distance

This metric utilized to indicate how much is the minimal Euclidean distance between elements of PF-known and the nearest members of PF-true [65]. IGD is calculated using the following formula:

$$\mathrm{IGD} \triangleq \frac{(\sum_{i=1}^{m} \bar{d}_i^{p})^{\frac{1}{p}}}{m}$$

Where $\bar{d}_i^{p}$ denotes the minimum Euclidean distance from each element i in PF-known and the nearest member of PF-true [66,67].

2.6.4 Hyperarea and Hyperarea Ratio

Objective space volume between the members of known-PF and true-PF is determined by using a measurement instrument called Hypervolume (HV). The best value among several Hypervolume values is the largest one. Hypervolume value is computed by the formula [1]:

$$HV_k = \frac{HV_k^*}{max_{i\,=\,1,2,...,N}\, HV_i^*}$$

where k=1,2,….., N, HV_k is the normalized value of HV_k^* and HV_k^* is the kth hypervolume value for a test problem.

Chapter 3

METHODOLOGY

In this study, Multi-Objective Differential Evolution (MODE) is employed to define a new approach which aims to solve ZDT,UF and MaOP problems. Four distinct variants of our algorithm were developed and introduced in this thesis. Each one of these variants will be discussed in detail in this chapter.

3.1 Software Used

As mentioned in chapter 2, several MOEAs which are based on DE are available currently and some of them are referred as MODE. The source code for Multi-objective differential evolution algorithm MODE, which utilized in this thesis, was developed by Meza G. M. The flowchart of this algorithm is illustrated on figure 5.

Before running the algorithm, all required parameters should be set. These parameters include number of objectives, number of decision variables, number of evaluation functions, number of generations, and population size. The algorithm starts working by creating the population using random techniques and then evaluating the objective functions of each individual in the population.

The next two steps are mutation stage and crossover stage respectively. This algorithm uses the DE/rand/1/ mutation scheme (in which 3 agents are selected randomly to produce the donor (mutant) vector) and bionomial crossover (the working principle of this type of crossover is illustrated in figure 3). This means that DE/rand/1/bin strategy (the standard DE strategy) is the only strategy that employed in this algorithm to create the trial vector. Then the cost functions of the trial vector is calculated and subsequently compared their corresponding cost functions of the corresponding target vector.

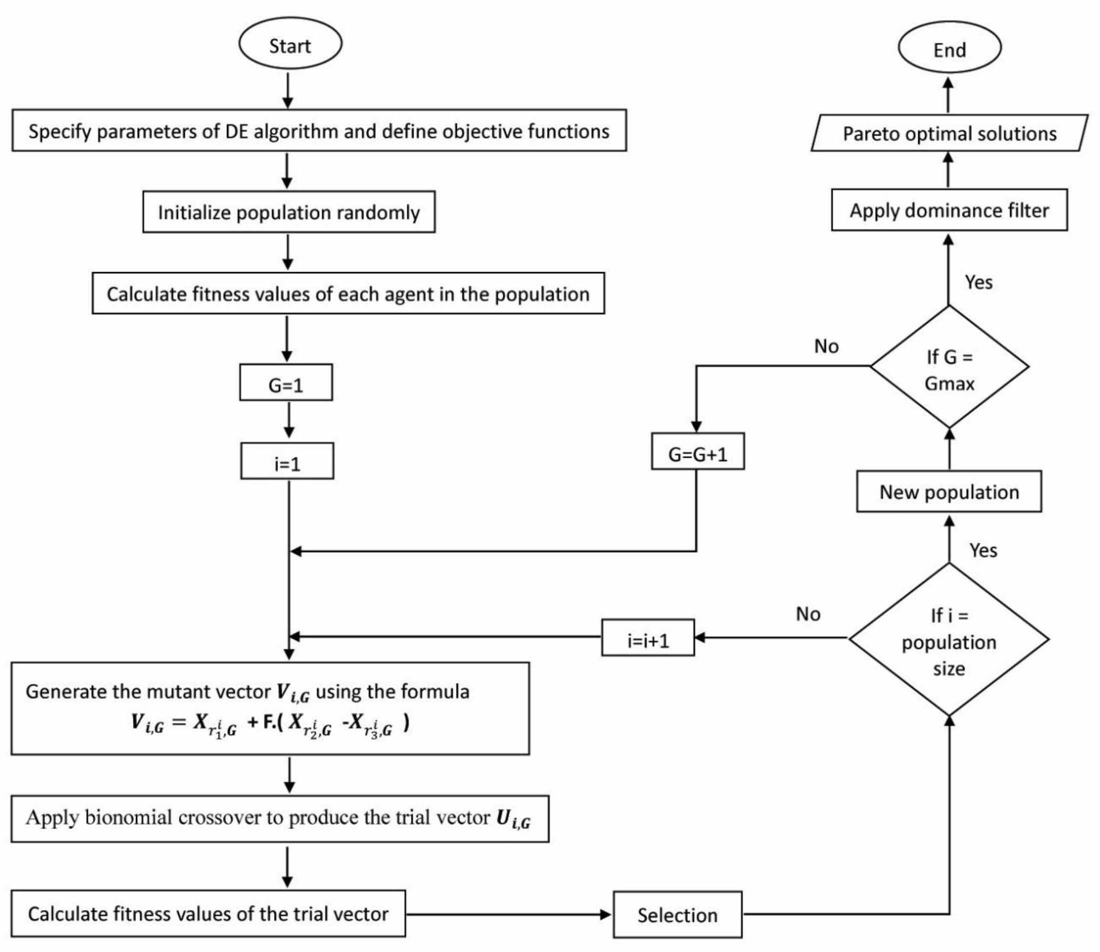


Figure 5: Flowchart of MODE algorithm

After that, the target vector is replaced by the trial vector only if the trial vector objective functions are better than their corresponding objective functions of the corresponding target vector. This step is called selection stage. After doing this process (mutation, crossover and selection stages) for all agents in the population, a new generation (population) will be formed and this process will be applied on the new generation. This will last until the stopping criteria (reaching the maximum number of generations) is satisfied. Afterward, the fitness values of each two individuals in the population will be compared on the basis of Pareto dominance relation to determine the non-dominated individuals which are also known as Pareto optimal solutions.

3.2 Version 1: MODE using 10 Different DE Strategies (mnv-MODE V1)

After setting all relevant parameters and defining objective functions, this version start running by creating DE population randomly and subsequently evaluating the objective function of each member of the parent population. Then the individuals in the population are ranked on the basis of the number of individuals that each individual dominated by. The non-dominated solutions will get the first rank and subsequently added to an external population known as archive.

Afterward, the parent population is divided into ten subpopulations where each subpopulation consists of the same number of agents. Instead of using only one DE strategy, 10 different DE strategies will be utilized to create the trial vector. Each subpopulation will utilizes a DE variant (which is not used in the other nine subpopulations) to do this task. The DE strategies which used in this thesis are mentioned before in chapter 2. Six of these strategies utilizes the best agent vector X_{best} in addition to the randomly selected vectors to produce the mutant vector. X_{best} is selected randomly from the non-dominated solutions in the archive. Then the objective functions of the trial vector are evaluated. The next stage is selection which is same as selection step in MODE. Whenever Mutation, Crossover and selection stages are done for all population members, a new population is emerged. Then the new population individuals are ranked (using the same way in which the old population members were ranked) and the non-dominated solutions are added to the archive.

Afterward, the archive is updated by removing any dominated solution from the archive. Then the same progress from dividing population step until this step is performed using the new population. This cycle remains active unless the termination criterion (number of generation will be equal to G_{max}) is satisfied. If the stopping criterion is satisfied and pareto-dominance relation is applied on the last population, this algorithm will yield Pareto optimal solutions. Figure 6 shows the flowchart of the algorithm of version 1.

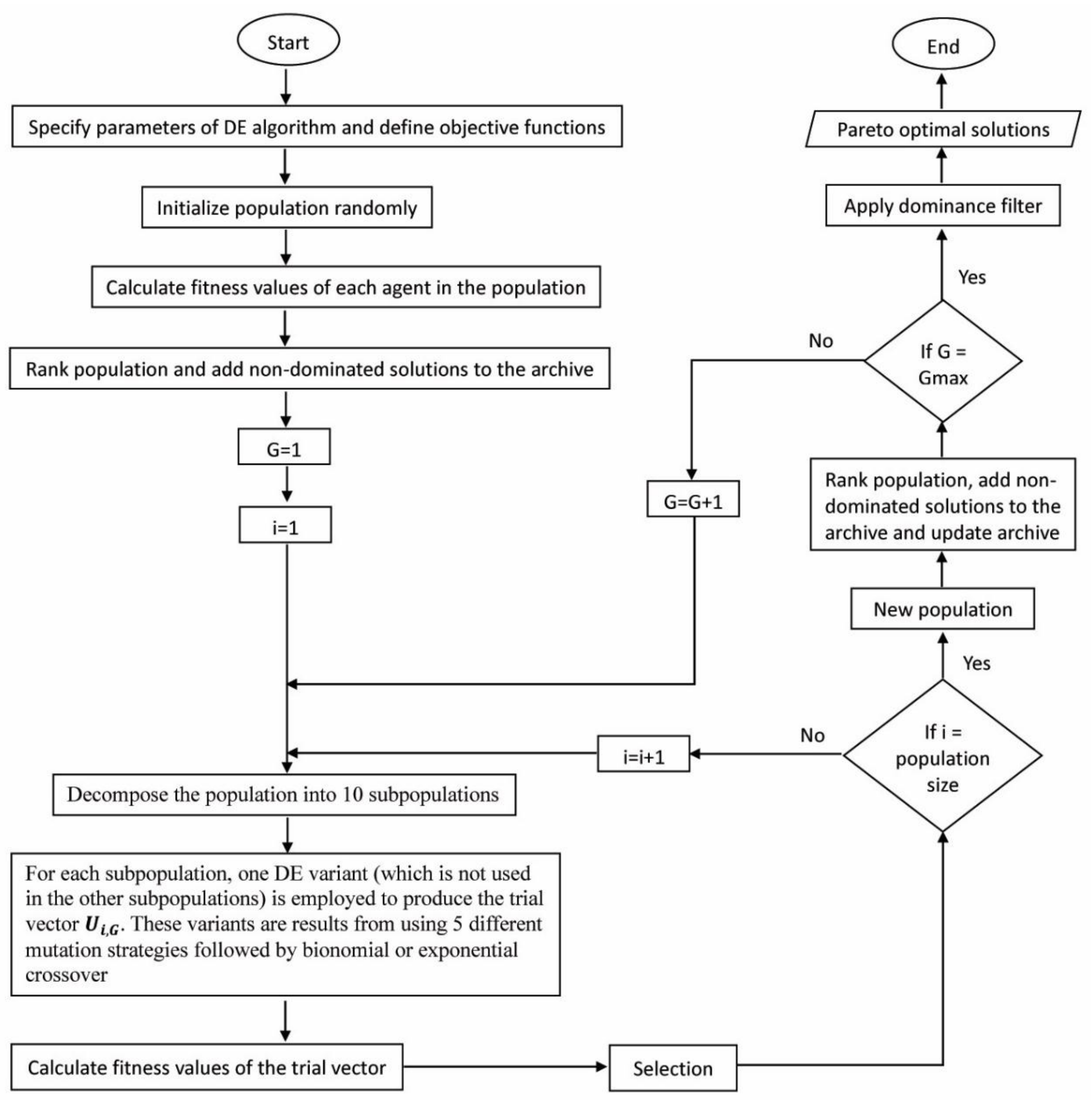


Figure 6: Flowchart of version 1 algorithm

3.3 Version 2: mnv-MODE V2

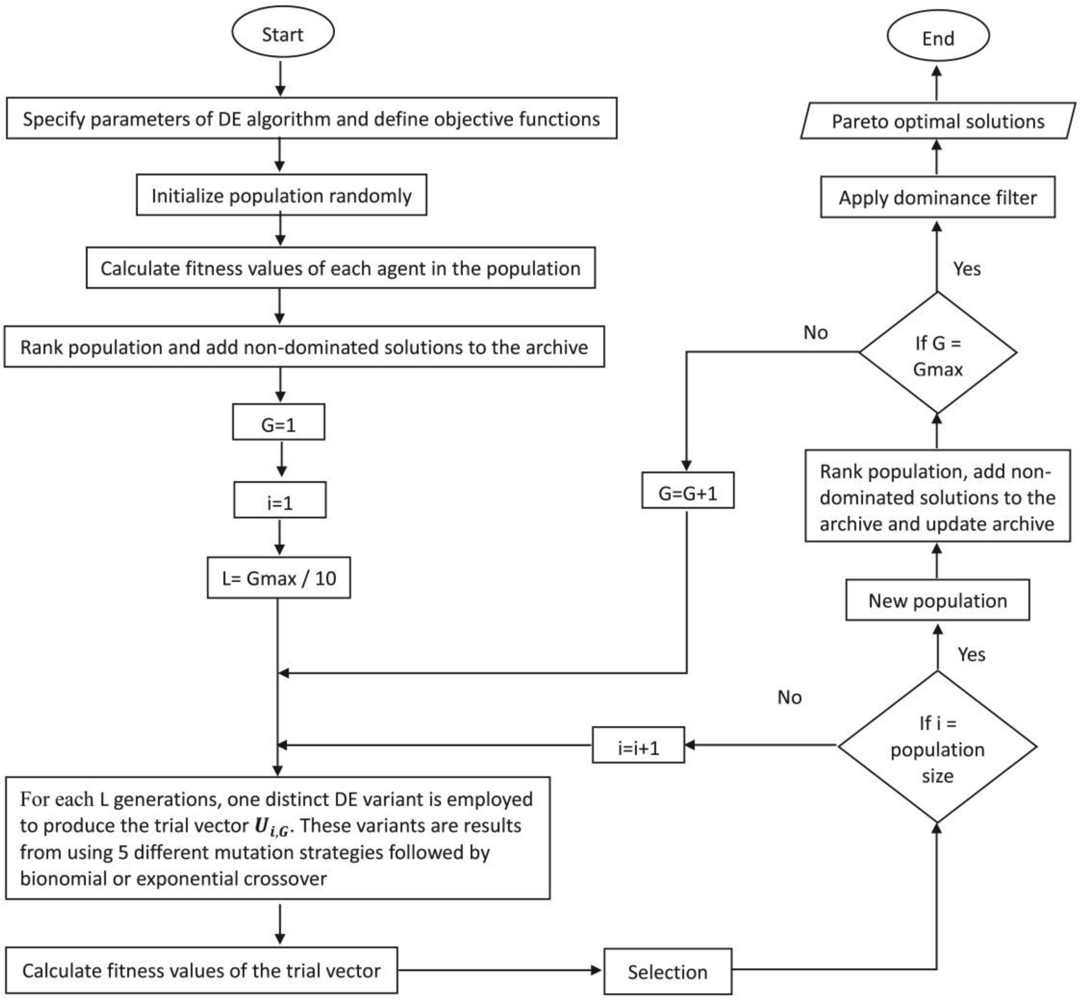


Figure 7: Flowchart of version 2 algorithm

As shown in figure 7, most of the steps of the algorithm in this version is same as the steps of version 1 algorithm. The differences between version 1 and version 2 are concentrated in the following points:

- Instead of dividing the parent population into 10 subpopulation, we will define an integer L. The value of L equal the quotient of dividing maximum number of generations G_{max} by 10.

- In each number of generations equal to L, a DE strategy which is not used in another L generations is employed to generate the trial vector corresponding to each agent in the population.

After obtaining the trial vector, the remaining steps of this version is exactly the same as the steps of version 1 algorithm after creating the trial vector.

3.4 Version 3: mnv-MODE V3

Similar to the previous versions, this version beginning with initialize population randomly after defining all required parameters and objective functions. Then the algorithm follow the same steps of version 2 until the stage in which an integer L is defined. Like version 2, the value of this integer L equal the result of the division process when dividing G_{max} by 10. We referred to the first L generations as G1, the second L generations as G2, the third L generations as G3 and so on. Afterward, either procedure A or procedure B will be performed on the current population.

In the generations which belong to G1, G3, G5, G7 or G9, the procedure A which consists of following steps will be done:

1) Divide the population into 10 subpopulations (groups) in which each subpopulation contain the same number of individuals.
2) A DE strategy is employed in each subpopulation to obtain the trial vector and subsequently the objective functions of this trial vector are evaluated. The DE strategy which utilized in one subpopulation will not be used in the other nine subpopulations.
3) Each subpopulation is ranked on the basis of the answer of the following question: how many individuals are dominated an individual?

The non-dominated solutions, which have the first rank, in each subpopulation are stored in an additional archive related with that subpopulation. After each generation, all the archives will be updated and all dominated solutions will be eliminated.

After the last generation in G1, the inverted generational distance IGD of each subpopulation will be computed. Subsequently, the minimum IGD value among the ten IGD values is selected as the best IGD value IGD_{best}. IGD_{best} is updated once after the last generation in each one of the following groups: G3, G5, G7 and G9. In procedure B which executed in G2, G4, G6, G8 and G10 generations, the DE strategy which utilized in the subpopulation whose IGD value is the best will be performed in the whole population.

After applying procedure A or B, the algorithm move to the selection stage to produce the new population. As in version1 and 2, the new population is then ranked and the non-dominated solutions will be added to the archive. The archive will be updated after each generation. If the maximum number of generations G_{max} is reached, Pareto-dominance relation is performed on the current population to get the pareto optimal solutions. Figure 8 illustrate the working principle of this version.

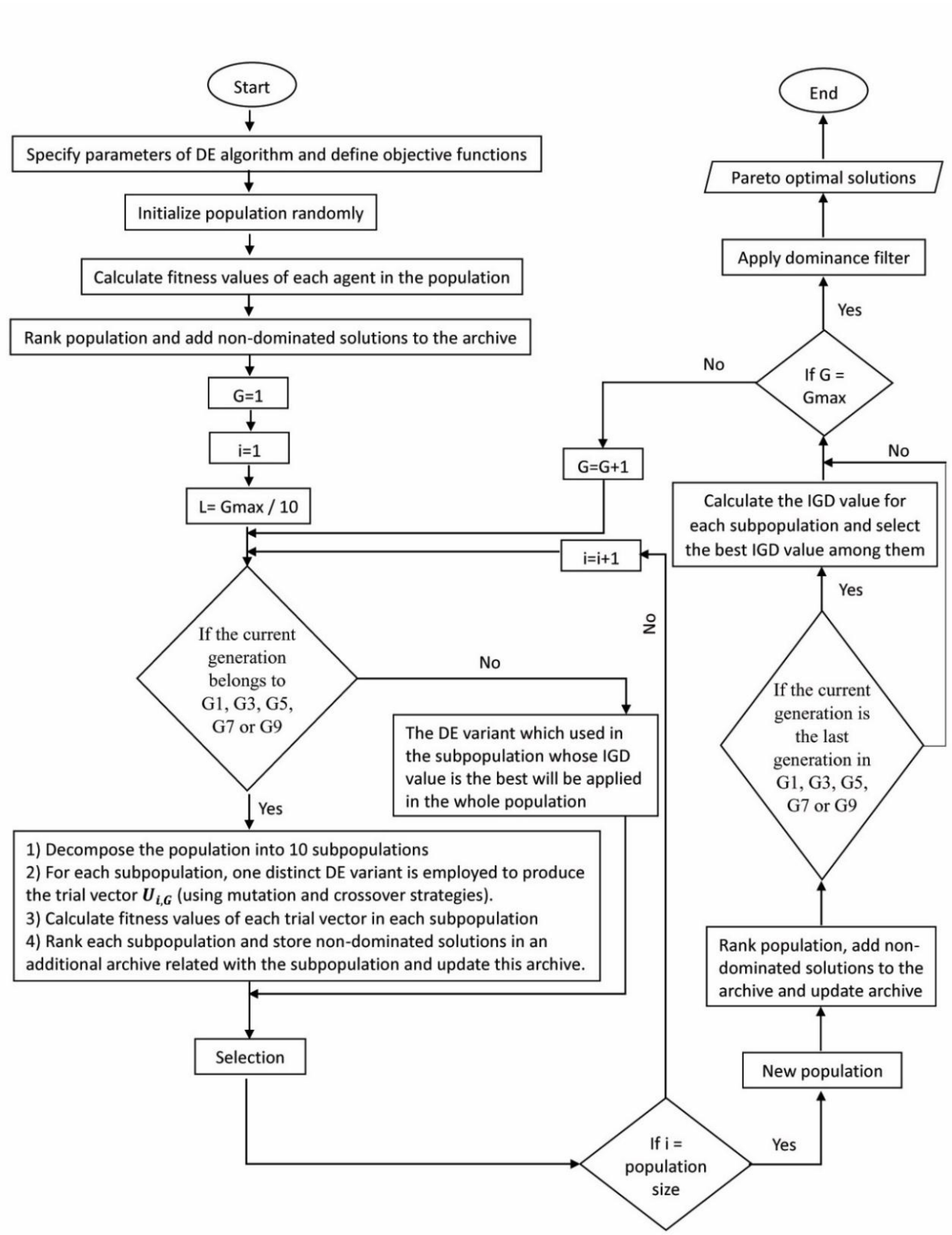


Figure 8: Flowchart of version 3 algorithm.

3.5 Version 4: mnv-MODE V4

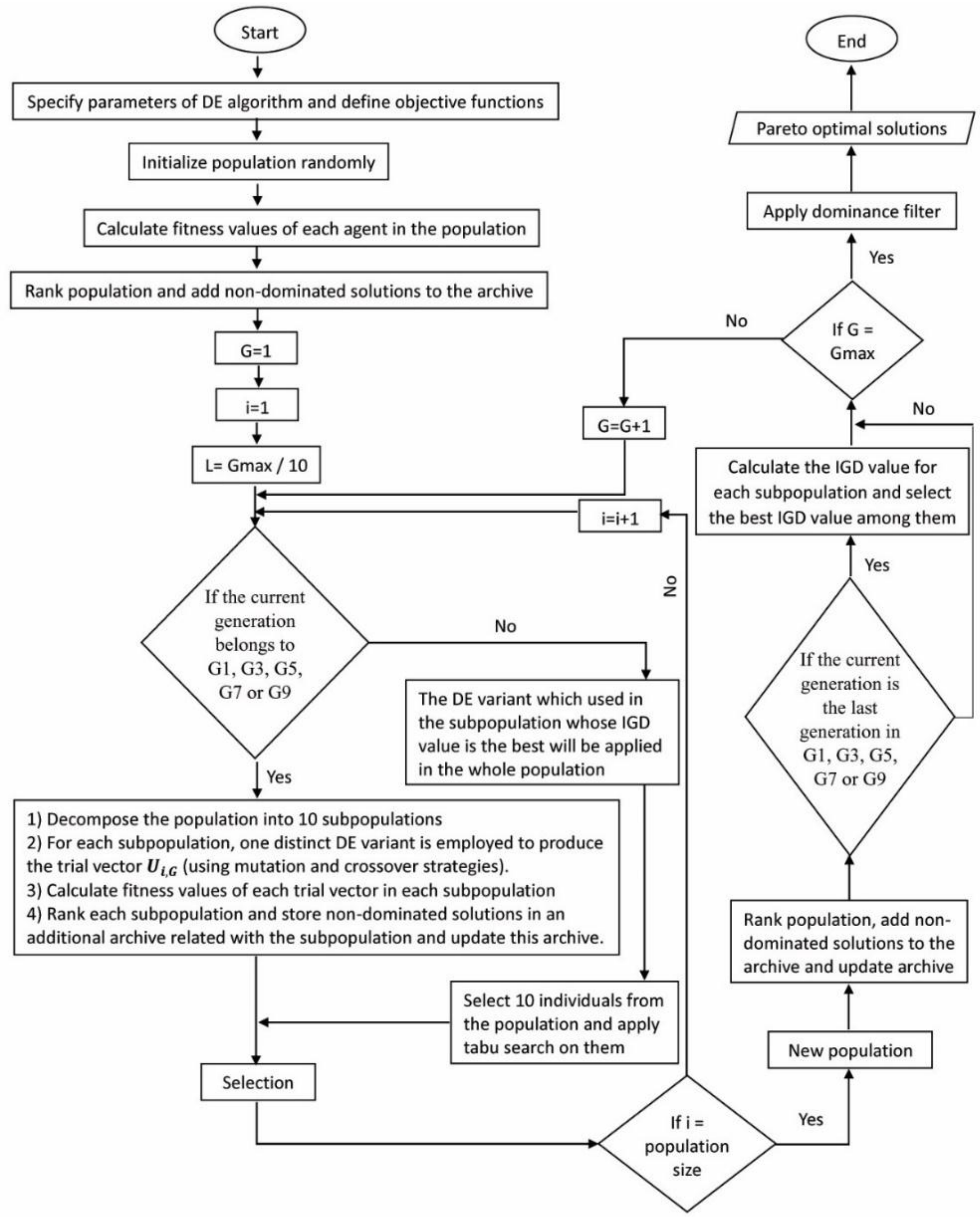


Figure 9: Working principle of version 4

In this version, only one stage is added to the algorithm of the previous version. This stage is the usage of tabu search to increase the efficiency of the algorithm. This stage follows the step in which the best DE variant (the strategy which possess IGD_{best}) is applied in the population in each generation belongs to G2, G4, G6, G8 and G10. All the steps which lies before and after this stage is exactly the same as the steps in version 3 and performed in the same way. In this stage, ten agents are chosen randomly from the population and tabu search will be applied on each one of these agents for 15 iteration. Figure 9 demonstrate the flowchart of this version.

Chapter 4

EXPERIMENTAL RESULTS

This chapter highlights the experimental results of this study and can be organized as follows: firstly, the results of our four proposed methods are presented and then compared with their corresponding results of another algorithms. Lastly, Fredman test which was applied to rank the algorithms is explained and the algorithms are sorted according to that ranking procedure.

To check the robustness, effectiveness and efficiency of the proposed algorithms, various test problems are needed. Twenty-five different benchmark problems were employed in this study as test problems to obtain the results and evaluate our proposed algorithms. The detailed information of these problems was provided/presented in chapter 2. This information include the number of objective (N_{obj}) of each problem. In contrast to ZDT and UF problems in which N_{obj} is a fixed number and cannot be changed, the number of objective in MaOP problems can be set to 3, 5, 8 or 10 which means that we can easily change N_{obj} value if the test problem belongs to MaOP1-MaOP10 problems. In this thesis, each one of MaOP problems was applied twice: once with three objectives and the second time with five objectives.

4.1 Experimental settings and simulation results

Each algorithm aims to solve a ZDT or UF problem was executed 30 independent times. On the other hand, only 20 runs were performed independently by each algorithm seeks to solve one of the MaOP problems. We will refer to this number of independent runs as $\boldsymbol{N_{runs}}$ in this thesis. Table 4 illustrates the parameters settings used for all algorithms in this study. The algorithm is terminated when the maximum number of function evaluations **MaxFunEvals** or the maximum number of generations **MaxGen** is reached (taken in consideration that **MaxFunEvals =. MaxGen *** Population size**).**

Table 4: parameter settings utilized in this study

	Number of decision variables	**Population size (pop)**	**Maximum number of evaluation functions (Maxfunevals)**
ZDT1-ZDT3	30	100	30000
ZDT4 and ZDT6	10	100	30000
UF1-UF7	30	100	300000
UF8-UF10	30	150	300000
MaOP1-MaOP10 for 3 objectives	20	300	150000
MaOP1-MaOP10 for 5 objectives	20	500	250000

The metric which utilized in this thesis to measure the performance of the algorithms is IGD metric. Among several obtained solutions of a test problem, the solution whose IGD value is the smallest is considered as the best solution. The simulation results of version1 to version 4 are reported respectively in the tables from table 5 to table 8.

Each test problem is expressed by 4 IGD values (the best (Min), worst (Max), mean and standard deviation (std)) in these tables.

Table 5: IGD values for all problems using version 1.

	best	worst	mean	std
ZDT1	2.74E-04	5.00E-04	3.78E-04	6.31E-05
ZDT2	5.89E-04	1.34E-03	8.41E-04	1.80E-04
ZDT3	6.24E-04	1.41E-03	9.76E-04	2.29E-04
ZDT4	1.57E-02	6.30E-02	4.13E-02	1.09E-02
ZDT6	2.88E-04	6.48E-04	4.45E-04	1.03E-04
UF1	5.34E-04	1.10E-03	6.84E-04	1.46E-04
UF2	4.54E-04	8.45E-04	5.96E-04	1.03E-04
UF3	5.57E-03	7.39E-03	6.34E-03	3.96E-04
UF4	1.17E-03	1.38E-03	1.22E-03	3.93E-05
UF5	5.71E-02	1.05E-01	8.77E-02	1.16E-02
UF6	9.15E-03	1.43E-02	1.23E-02	1.19E-03
UF7	1.96E-03	1.28E-02	5.21E-03	3.21E-03
UF8	1.59E-03	3.08E-03	2.04E-03	3.11E-04
UF9	1.18E-03	2.09E-03	1.64E-03	2.52E-04
UF10	5.91E-03	8.76E-03	6.88E-03	6.23E-04
MaOP1*	5.09E-01	5.23E-01	5.16E-01	3.65E-03
MaOP2*	3.59E-03	5.07E-03	4.18E-03	4.39E-04
MaOP3*	2.65E+00	2.96E+00	2.82E+00	8.27E-02
MaOP4*	2.09E-02	2.09E-02	2.09E-02	1.37E-06
MaOP5*	1.57E-02	1.73E-02	1.63E-02	5.04E-04
MaOP6*	1.17E-02	1.71E-02	1.36E-02	1.22E-03
MaOP7*	7.94E-03	1.00E-02	9.01E-03	6.47E-04
MaOP8*	7.36E-03	1.16E-02	9.14E-03	1.19E-03
MaOP9*	1.20E-02	2.29E-02	1.66E-02	2.74E-03
MaOP10*	9.28E-03	1.25E-02	1.04E-02	8.42E-04
MaOP1**	4.03E-01	4.12E-01	4.07E-01	2.52E-03
MaOP2**	2.43E-03	2.95E-03	2.65E-03	1.29E-04
MaOP3**	1.52E+00	1.67E+00	1.60E+00	3.57E-02
MaOP4**	9.75E-03	9.95E-03	9.83E-03	5.14E-05
MaOP5**	7.98E-03	9.38E-03	8.78E-03	3.91E-04
MaOP6**	7.96E-03	1.10E-02	9.63E-03	9.12E-04
MaOP7**	4.05E-03	5.12E-03	4.59E-03	2.79E-04
MaOP8**	4.27E-03	5.56E-03	4.89E-03	3.51E-04
MaOP9**	5.23E-03	8.34E-03	6.09E-03	8.99E-04
MaOP10**	4.34E-03	4.96E-03	4.61E-03	1.85E-04

*3 objectives
**5 objectives

Table 6: IGD values for all problems using version 2.

	best	worst	mean	std
ZDT1	3.48E-04	1.40E-03	6.12E-04	2.42E-04
ZDT2	5.62E-03	2.31E-02	1.72E-02	7.09E-03
ZDT3	5.26E-04	1.52E-03	9.58E-04	2.86E-04
ZDT4	5.30E-02	1.12E-01	7.59E-02	1.54E-02
ZDT6	7.15E-04	1.50E-02	2.87E-03	3.24E-03
UF1	5.59E-04	1.43E-03	7.42E-04	2.05E-04
UF2	6.66E-04	1.20E-03	9.12E-04	1.67E-04
UF3	5.21E-03	9.19E-03	7.31E-03	1.03E-03
UF4	1.21E-03	1.33E-03	1.25E-03	3.10E-05
UF5	5.61E-02	1.19E-01	8.11E-02	1.41E-02
UF6	4.16E-03	1.29E-02	8.25E-03	2.77E-03
UF7	1.39E-03	1.08E-02	4.26E-03	2.06E-03
UF8	4.49E-03	5.68E-03	5.51E-03	2.91E-04
UF9	2.95E-03	5.34E-03	4.44E-03	6.12E-04
UF10	4.51E-03	5.57E-03	5.11E-03	2.67E-04
MaOP1*	5.15E-01	5.23E-01	5.19E-01	1.87E-03
MaOP2*	3.90E-03	5.02E-03	4.51E-03	3.30E-04
MaOP3*	2.19E+00	2.49E+00	2.32E+00	8.87E-02
MaOP4*	2.09E-02	2.09E-02	2.09E-02	1.00E-06
MaOP5*	1.75E-02	2.75E-02	1.95E-02	2.28E-03
MaOP6*	1.22E-02	2.41E-02	1.69E-02	3.71E-03
MaOP7*	7.66E-03	1.70E-02	1.06E-02	2.27E-03
MaOP8*	8.62E-03	1.29E-02	1.06E-02	1.17E-03
MaOP9*	1.02E-02	2.72E-02	1.89E-02	3.93E-03
MaOP10*	9.34E-03	1.75E-02	1.12E-02	1.95E-03
MaOP1**	4.05E-01	4.15E-01	4.10E-01	2.73E-03
MaOP2**	2.33E-03	2.73E-03	2.53E-03	1.29E-04
MaOP3**	1.09E+00	1.24E+00	1.19E+00	3.22E-02
MaOP4**	9.83E-03	1.00E-02	9.91E-03	5.60E-05
MaOP5**	9.24E-03	9.65E-03	9.59E-03	8.74E-05
MaOP6**	7.78E-03	1.44E-02	1.14E-02	1.52E-03
MaOP7**	5.17E-03	6.21E-03	5.71E-03	2.85E-04
MaOP8**	5.79E-03	7.35E-03	6.50E-03	4.12E-04
MaOP9**	5.10E-03	7.64E-03	5.71E-03	6.55E-04
MaOP10**	4.27E-03	5.19E-03	4.62E-03	2.44E-04

*3 objectives

**5 objectives

Table 7: IGD values for all problems using version 3.

	best	worst	mean	std
ZDT1	2.68E-04	6.40E-04	3.99E-04	7.71E-05
ZDT2	6.38E-04	4.28E-03	1.82E-03	1.11E-03
ZDT3	5.43E-04	1.65E-03	9.16E-04	2.83E-04
ZDT4	3.46E-03	5.55E-02	1.96E-02	1.27E-02
ZDT6	2.72E-04	1.35E-03	5.44E-04	2.48E-04
UF1	5.28E-04	9.86E-04	7.03E-04	1.24E-04
UF2	4.72E-04	8.01E-04	5.86E-04	7.43E-05
UF3	5.33E-03	7.10E-03	6.02E-03	4.05E-04
UF4	1.21E-03	1.34E-03	1.24E-03	3.34E-05
UF5	3.94E-02	1.07E-01	7.19E-02	2.20E-02
UF6	4.16E-03	1.43E-02	9.74E-03	3.53E-03
UF7	1.32E-03	8.37E-03	3.53E-03	1.55E-03
UF8	1.58E-03	4.49E-03	2.77E-03	8.47E-04
UF9	1.07E-03	3.08E-03	1.71E-03	4.16E-04
UF10	5.72E-03	7.76E-03	6.56E-03	5.05E-04
MaOP1*	5.00E-01	5.10E-01	5.05E-01	2.21E-03
MaOP2*	2.58E-03	4.65E-03	3.76E-03	5.30E-04
MaOP3*	1.94E+00	2.37E+00	2.16E+00	1.27E-01
MaOP4*	2.09E-02	2.09E-02	2.09E-02	5.48E-07
MaOP5*	1.70E-02	1.78E-02	1.74E-02	1.94E-04
MaOP6*	1.30E-02	2.61E-02	1.58E-02	3.07E-03
MaOP7*	6.80E-03	1.21E-02	8.82E-03	1.59E-03
MaOP8*	6.70E-03	1.01E-02	8.27E-03	9.42E-04
MaOP9*	1.08E-02	2.51E-02	1.61E-02	4.35E-03
MaOP10*	7.84E-03	1.27E-02	9.42E-03	1.04E-03
MaOP1**	3.90E-01	4.01E-01	3.95E-01	3.22E-03
MaOP2**	2.17E-03	2.81E-03	2.48E-03	1.67E-04
MaOP3**	1.18E+00	1.49E+00	1.32E+00	8.92E-02
MaOP4**	1.92E-02	2.20E-02	2.07E-02	9.73E-04
MaOP5**	9.31E-03	9.57E-03	9.43E-03	7.46E-05
MaOP6**	6.19E-03	1.21E-02	9.18E-03	1.58E-03
MaOP7**	3.60E-03	5.16E-03	4.05E-03	4.98E-04
MaOP8**	3.68E-03	5.95E-03	4.72E-03	8.18E-04
MaOP9**	4.43E-03	7.44E-03	5.13E-03	6.85E-04
MaOP10**	3.84E-03	4.61E-03	4.23E-03	1.97E-04

*3objectives
**5objectives

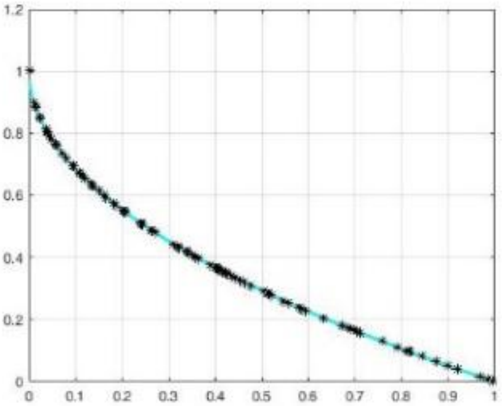
Table 8: IGD values for all problems using version 4.

	best	worst	mean	std
ZDT1	3.08E-04	4.51E-04	3.76E-04	4.12E-05
ZDT2	6.38E-04	4.32E-03	1.91E-03	9.43E-04
ZDT3	6.30E-04	1.57E-03	9.96E-04	2.59E-04
ZDT4	1.72E-03	1.70E-02	5.78E-03	4.45E-03
ZDT6	2.91E-04	2.10E-03	6.22E-04	4.55E-04
UF1	4.37E-04	1.35E-03	7.15E-04	2.79E-04
UF2	4.56E-04	9.94E-04	6.26E-04	1.22E-04
UF3	3.08E-03	4.64E-03	3.70E-03	4.07E-04
UF4	1.20E-03	1.32E-03	1.25E-03	3.23E-05
UF5	2.78E-02	4.95E-02	3.85E-02	5.98E-03
UF6	3.92E-03	7.97E-03	5.30E-03	1.20E-03
UF7	1.65E-03	4.51E-03	3.05E-03	8.05E-04
UF8	1.35E-03	3.85E-03	2.66E-03	6.96E-04
UF9	9.62E-04	2.74E-03	1.57E-03	3.78E-04
UF10	3.38E-03	5.55E-03	4.46E-03	5.18E-04
MaOP1*	5.02E-01	5.16E-01	5.07E-01	3.73E-03
MaOP2*	2.76E-03	4.67E-03	3.49E-03	5.11E-04
MaOP3*	1.91E+00	2.85E+00	2.23E+00	2.07E-01
MaOP4*	2.09E-02	2.09E-02	2.09E-02	1.04E-06
MaOP5*	1.69E-02	1.81E-02	1.75E-02	3.53E-04
MaOP6*	1.13E-02	2.07E-02	1.41E-02	2.40E-03
MaOP7*	6.39E-03	1.05E-02	7.53E-03	1.13E-03
MaOP8*	6.06E-03	1.21E-02	8.64E-03	1.68E-03
MaOP9*	1.03E-02	2.66E-02	1.57E-02	5.48E-03
MaOP10*	8.30E-03	1.11E-02	9.19E-03	7.55E-04
MaOP1**	3.91E-01	3.99E-01	3.95E-01	2.45E-03
MaOP2**	2.07E-03	3.06E-03	2.43E-03	2.15E-04
MaOP3**	1.17E+00	1.42E+00	1.32E+00	7.77E-02
MaOP4**	9.73E-03	9.92E-03	9.82E-03	5.27E-05
MaOP5**	8.46E-03	9.56E-03	9.32E-03	2.57E-04
MaOP6**	6.89E-03	1.12E-02	8.61E-03	1.08E-03
MaOP7**	3.36E-03	4.97E-03	4.18E-03	5.50E-04
MaOP8**	3.59E-03	5.94E-03	4.31E-03	5.98E-04
MaOP9**	4.42E-03	6.71E-03	5.06E-03	5.58E-04
MaOP10**	3.87E-03	4.55E-03	4.22E-03	1.77E-04

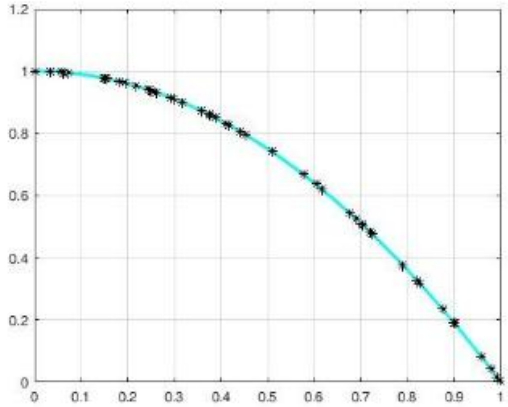
*3objectives
**5objectives

As shown in tables 5-8, MaOP3 and then MaOP1 are the hardest problem to solve. By setting the number of objective in MaOP problems to five, it is observed that the obtained IGD values are better than when it was set as three. For UF problems and all two objective problems, it can be seen that the IGD values obtained by UF5 are the largest and the worst. The tables shows also that ZDT4 has the worst IGD value among ZDT problems. For 3-objective UF problems, the performance of the four proposed methods on UF9 is better than the other two problems (UF8 and UF10).

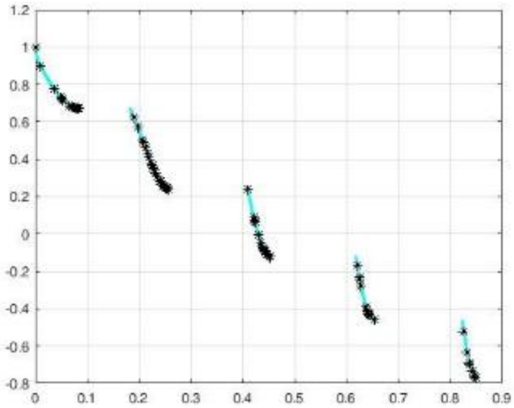
The final approximation set with the smallest IGD values for all test problems solved by mnv-MODE v4 are plotted in the figures 10-14. Figures 10–14 illustrates PF-true and PF-known for ZDT, UF and 3 objective MaOP problems solved by version 4. While the black stars in the figures 10-12 indicates/represents the pareto-front obtained by version 4, the true pareto front is represented by cyan dots in these figures. These figures shows that version 4 achieved good IGD results/ values on the majority of the test problems. Figures 10-14 shows that most of the problems has good convergence towards the true pareto front in most test instance. The divergence of the solutions in the obtained pareto front can be noticed easily by looking at the Figures 10-14.



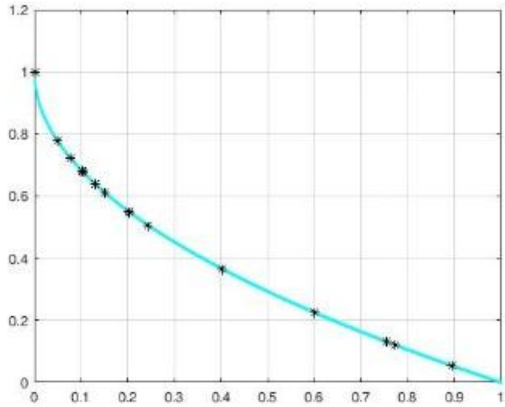
ZDT1



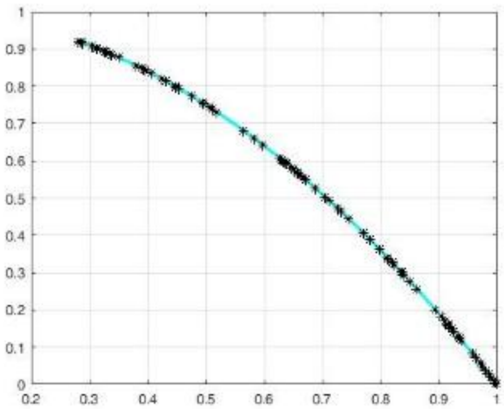
ZDT2



ZDT3

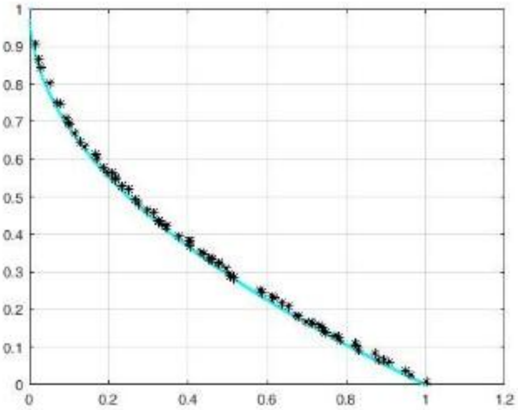


ZDT4

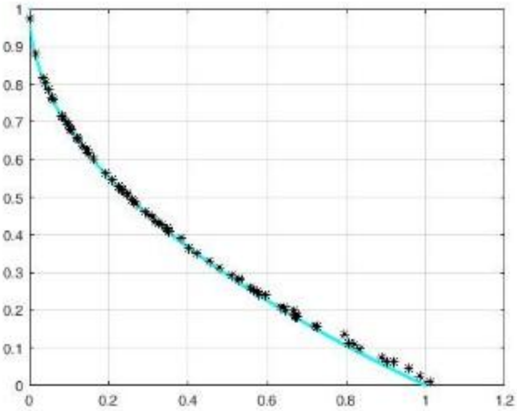


ZDT6

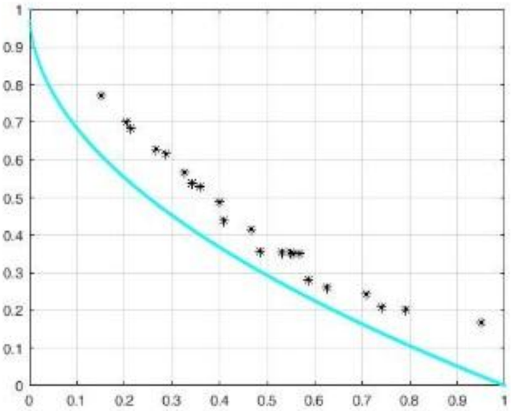
Figure 10: Best Pareto-fronts of ZDT1-ZDT4 and ZDT6 in mnv-MODE v4



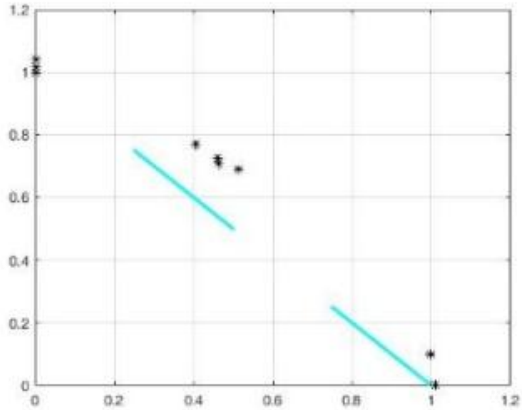
UF1



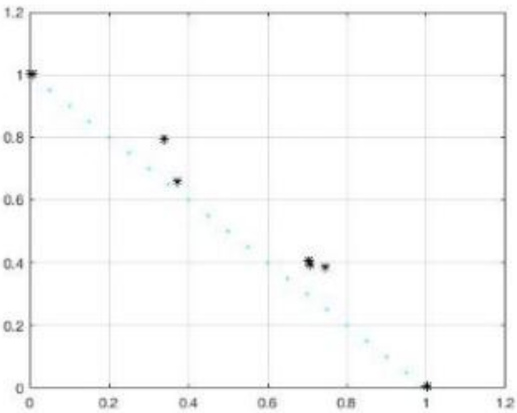
UF2



UF**3**

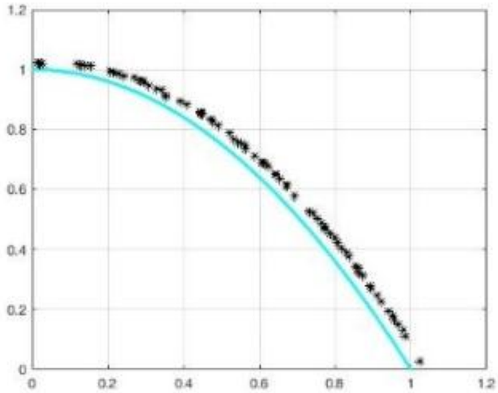


UF**4**

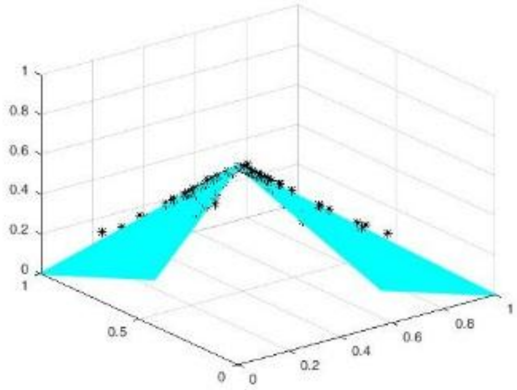


UF5

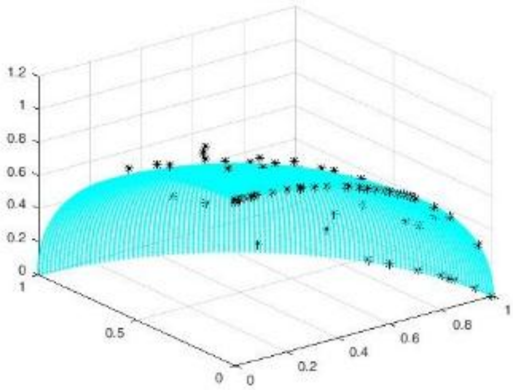
Figure 11: Best Pareto-fronts of UF1-UF5 in mnv-MODE v4



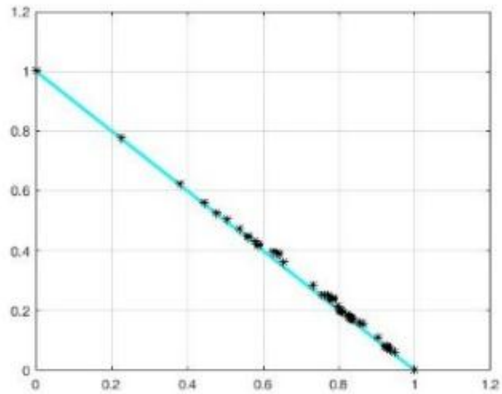
UF**6**



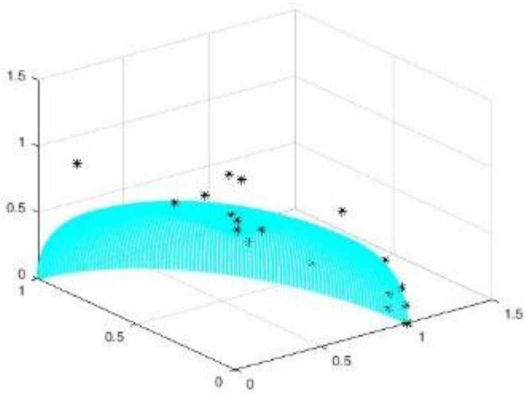
UF7



UF8

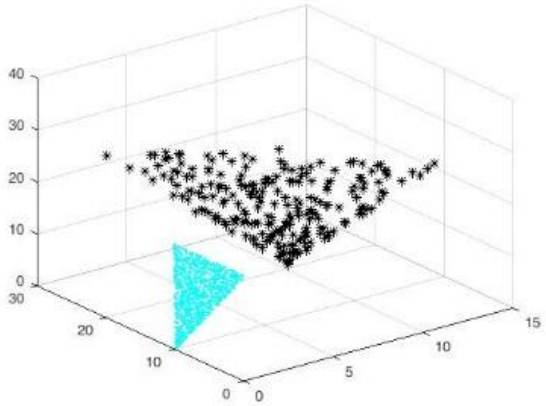


UF9

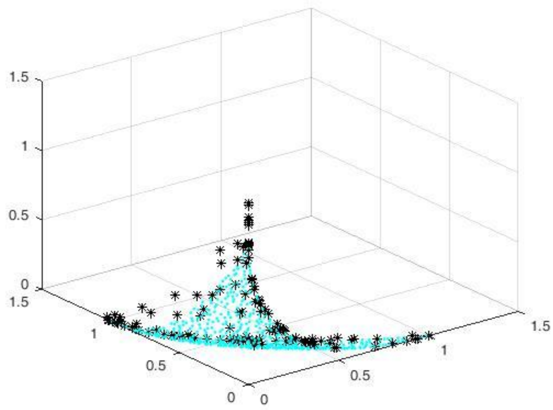


UF10

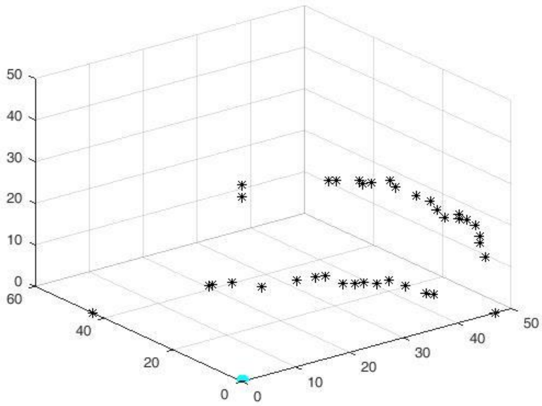
Figure 12: Best Pareto-fronts of UF6-UF10 in mnv-MODE v4



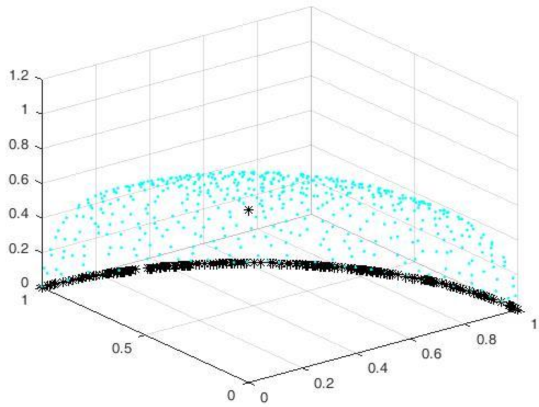
MaOP1_3obj



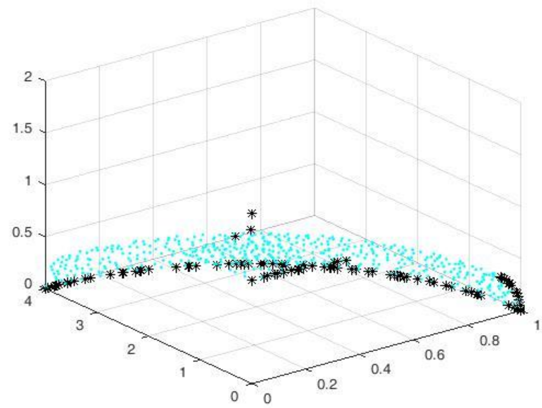
MaOP2_3obj



MaOP3_obj

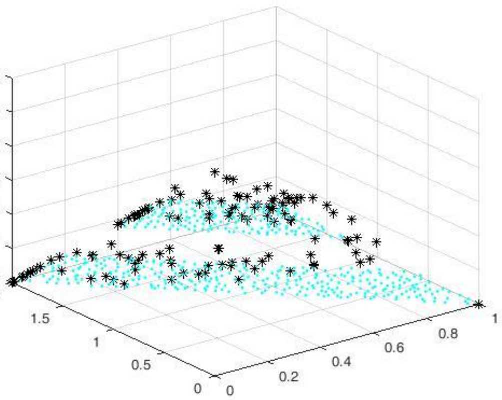


MaOP4 _3obj

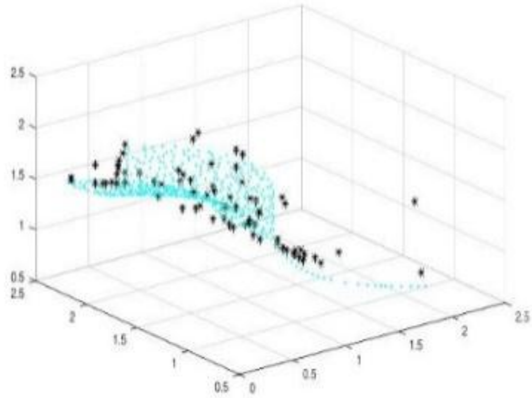


MaOP5_3obj

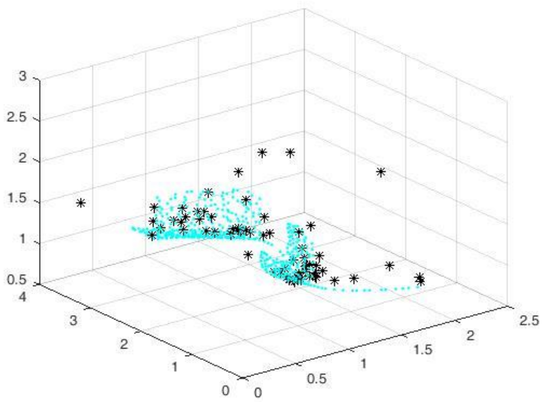
Figure 13: Best Pareto-fronts of MaOP1-MaOP5 in mnv-MODE v4



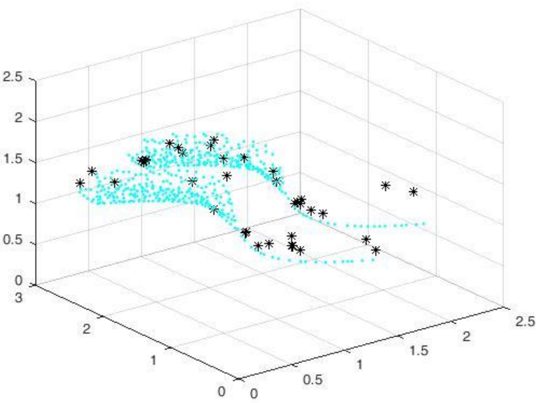
MaOP6_3obj



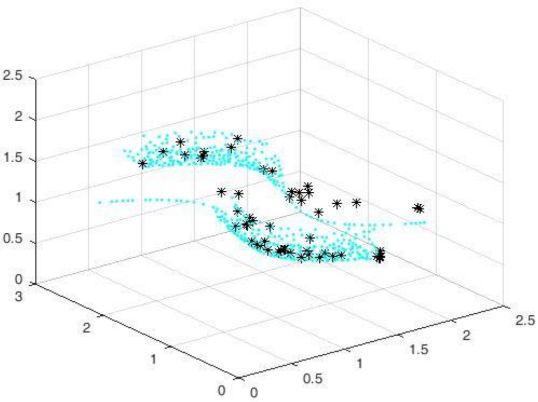
MaOP7_3obj



MaOP8_3obj



MaOP9_3obj



MaOP10_3obj

Figure 14: Best Pareto-fronts of MaOP6-MaOP10 in mnv-MODE v4

4.2 Result of comparing MODE with the four proposed methods.

Table 9: Results Comparisons between MODE and the proposed methods.

	MODE	mnv-MODE v1	mnv-MODE v2	mnv-MODE v3	mnv-MODE v4
ZDT1	6.244E-04	3.778E-04	6.119E-04	3.993E-04	3.760E-04
ZDT2	1.555E-02	8.408E-04	1.715E-02	1.824E-03	1.915E-03
ZDT3	9.874E-04	9.760E-04	9.576E-04	9.164E-04	9.959E-04
ZDT4	2.081E-01	4.132E-02	7.595E-02	1.961E-02	5.780E-03
ZDT6	1.756E-03	4.453E-04	2.866E-03	5.440E-04	6.220E-04
UF1	2.817E-03	6.837E-04	7.422E-04	7.025E-04	7.155E-04
UF2	1.187E-03	5.960E-04	9.115E-04	5.858E-04	6.262E-04
UF3	8.508E-03	6.336E-03	7.313E-03	6.018E-03	3.698E-03
UF4	2.190E-03	1.223E-03	1.255E-03	1.244E-03	1.249E-03
UF5	2.251E-01	8.769E-02	8.112E-02	7.188E-02	3.850E-02
UF6	1.463E-02	1.233E-02	8.249E-03	9.737E-03	5.299E-03
UF7	1.348E-02	5.205E-03	4.262E-03	3.530E-03	3.049E-03
UF8	5.600E-03	2.041E-03	5.508E-03	2.774E-03	2.662E-03
UF9	5.050E-03	1.639E-03	4.435E-03	1.707E-03	1.572E-03
UF10	2.068E-02	6.881E-03	5.112E-03	6.560E-03	4.458E-03
MaOP1*	5.886E-01	5.157E-01	5.188E-01	5.048E-01	5.071E-01
MaOP2*	2.128E-02	4.184E-03	4.513E-03	3.755E-03	3.494E-03
MaOP3*	3.159E+00	2.819E+00	2.320E+00	2.164E+00	2.232E+00
MaOP4*	2.288E-02	2.090E-02	2.090E-02	2.090E-02	2.090E-02
MaOP5*	1.985E-02	1.633E-02	1.950E-02	1.735E-02	1.746E-02
MaOP6*	2.567E-02	1.356E-02	1.687E-02	1.578E-02	1.408E-02
MaOP7*	3.177E-02	9.005E-03	1.063E-02	8.817E-03	7.530E-03
MaOP8*	3.142E-02	9.142E-03	1.064E-02	8.270E-03	8.640E-03
MaOP9*	5.366E-02	1.660E-02	1.892E-02	1.606E-02	1.567E-02
MaOP10*	4.425E-02	1.043E-02	1.120E-02	9.422E-03	9.185E-03
MaOP1**	4.881E-01	4.066E-01	4.097E-01	3.951E-01	3.946E-01
MaOP2**	1.831E-02	2.652E-03	2.534E-03	2.482E-03	2.430E-03
MaOP3**	2.099E+00	1.604E+00	1.195E+00	1.321E+00	1.323E+00
MaOP4**	9.912E-03	9.835E-03	9.908E-03	2.066E-02	9.824E-03
MaOP5**	9.552E-03	8.780E-03	9.588E-03	9.433E-03	9.320E-03
MaOP6**	2.066E-02	9.627E-03	1.145E-02	9.176E-03	8.606E-03
MaOP7**	5.782E-03	4.594E-03	5.706E-03	4.050E-03	4.181E-03
MaOP8**	6.233E-03	4.887E-03	6.505E-03	4.718E-03	4.312E-03
MaOP9**	5.482E-03	6.086E-03	5.711E-03	5.125E-03	5.057E-03
MaOP10**	4.584E-03	4.613E-03	4.618E-03	4.233E-03	4.221E-03

*3objectives
**5objectives

The mean IGD values of MODE and the four proposed methods are presented in table 9 in which the best mean IGD value is highlighted. None of these proposed methods has better IGD values than the rest on all test problems. From table 9, it is observed that version 4 outperforms the other algorithms in 19 test instance and version 3 outperforms other MOEAs only in 7 test problems. It can be seen also that version 1 is better than other methods in 8 test instances and version 2 is better than others only in the five objective MaOP3. It is also noticed that version 4 achieved the best performance in the 5-objective MaOPs.

4.3 Result of comparing our four proposed methods with other MOEAs.

After running each algorithm $\boldsymbol{N_{runs}}$ times, the mean IGD values for each algorithm on different problems is computed. In tables 10-21, each one of our four proposed methods is compared with other MOEAs on the basis of the average IGD value of each algorithm on several test instances. The highlighted value in each row indicates the best mean IGD value of the corresponding test problem and refers to the algorithm which achieved/has the best performance on this problem.

Table 10: Average IGD results of version 1 and other MOEAs on ZDT problems

	mnv-MODE v1	MODE	NSGA- II	MOEA/D	SPEA2
ZDT1	3.78E-04	6.24E-04	4.55E-03	8.18E-03	3.88E-03
ZDT2	8.41E-04	1.55E-02	4.74E-03	9.09E-03	3.89E-03
ZDT3	9.76E-04	9.87E-04	3.46E-02	1.73E-02	7.83E-03
ZDT4	4.13E-02	2.08E-01	5.93E-03	2.63E-02	5.07E-03
ZDT6	4.45E-04	1.76E-03	3.68E-03	6.47E-03	3.15E-03

Table 11: Average IGD results of version 1 and other MOEAs on UF problems

	mnv-MODE v1	MODE	MOEA/D	NSGA-II LS	MOEA/D-GM	GDE3	DMOEA-DD	MTS	LiuLi algorithm	DECMOSA-SQP	AMGA	clustering MOEA	MOEP	OWMOSaDE
UF1	6.84E-04	2.82E-03	4.35E-03	1.15E-02	6.20E-03	5.34E-03	1.04E-02	6.47E-03	7.85E-03	7.70E-02	3.59E-02	2.99E-02	5.96E-02	1.22E-02
UF2	5.96E-04	1.19E-03	6.79E-03	1.24E-02	6.40E-03	1.20E-02	6.79E-03	6.16E-03	1.23E-02	2.83E-02	1.62E-02	2.28E-02	1.89E-02	8.10E-03
UF3	6.34E-03	8.51E-03	7.42E-03	1.06E-01	4.29E-02	1.06E-01	3.34E-02	5.31E-02	1.50E-02	9.35E-02	7.00E-02	5.49E-02	9.90E-02	1.03E-01
UF4	1.22E-03	2.19E-03	6.39E-02	5.84E-02	4.76E-02	2.65E-02	4.27E-02	2.36E-02	4.35E-02	3.39E-02	4.06E-02	5.85E-02	4.27E-02	5.13E-02
UF5	8.77E-02	2.25E-01	1.81E-01	5.66E-01	1.79E+00	3.93E-02	3.15E-01	1.49E-02	1.62E-01	1.67E-01	9.41E-02	2.47E-01	2.25E-01	4.30E-01
UF6	1.23E-02	1.46E-02	5.87E-03	3.10E-01	5.56E-01	2.51E-01	6.67E-02	5.92E-02	1.76E-01	1.26E-01	1.29E-01	8.71E-02	1.03E-01	1.92E-01
UF7	5.21E-03	1.35E-02	4.44E-03	2.13E-02	7.60E-03	2.52E-02	1.03E-02	4.08E-02	7.30E-03	2.42E-02	5.71E-02	2.23E-02	1.97E-02	5.85E-02
UF8	2.04E-03	5.60E-03	5.84E-02	8.63E-02	2.45E-01	2.49E-01	6.84E-02	1.13E-01	8.24E-02	2.16E-01	1.71E-01	2.38E-01	4.23E-01	9.45E-02
UF9	1.64E-03	5.05E-03	7.90E-02	7.19E-02	1.88E-01	8.25E-02	4.90E-02	1.14E-01	9.39E-02	1.41E-01	1.89E-01	2.93E-01	3.42E-01	9.83E-02
UF10	6.88E-03	2.07E-02	4.74E-01	8.45E-01	5.65E-01	4.33E-01	3.22E-01	1.53E-01	4.47E-01	3.70E-01	3.24E-01	4.11E-01	3.62E-01	7.43E-01

Table 12: Average IGD results of version 1 and other MOEAs on MaOP problems

	mnv-MODE v1	MODE	MOEA/D	NSGA-III
MaOP1*	5.16E-01	5.89E-01	5.47E-01	5.12E-01
MaOP2*	4.18E-03	2.13E-02	1.14E-02	5.38E-03
MaOP3*	2.82E+00	3.16E+00	4.67E-01	2.62E+00
MaOP4*	2.09E-02	2.29E-02	2.09E-02	2.13E-02
MaOP5*	1.63E-02	1.98E-02	8.07E-02	1.27E-02
MaOP6*	1.36E-02	2.57E-02	5.71E-02	1.16E-02
MaOP7*	9.01E-03	3.18E-02	4.20E-02	2.27E-02
MaOP8*	9.14E-03	3.14E-02	3.68E-02	2.34E-02
MaOP9*	1.66E-02	5.37E-02	6.51E-02	3.36E-02
MaOP10*	1.04E-02	4.42E-02	3.91E-02	3.42E-02
MaOP1**	4.07E-01	4.88E-01	4.49E-01	4.89E-01
MaOP2**	2.65E-03	1.83E-02	4.56E-03	6.72E-03
MaOP3**	1.60E+00	2.10E+00	1.16E-02	2.44E+00
MaOP4**	9.83E-03	9.91E-03	1.02E-02	1.14E-02
MaOP5**	8.78E-03	9.55E-03	4.26E-02	2.23E-02
MaOP6**	9.63E-03	2.07E-02	3.14E-02	9.52E-02
MaOP7**	4.59E-03	5.78E-03	1.28E-02	1.19E-02
MaOP8**	4.89E-03	6.23E-03	1.14E-02	1.59E-02
MaOP9**	6.09E-03	5.48E-03	1.93E-02	2.38E-02
MaOP10**	4.61E-03	4.58E-03	1.69E-02	3.82E-02

* 3objectives
**5objectives

Table 13: Average IGD results of version 2 and other MOEAs on ZDT problems

	mnv-MODE v2	MODE	NSGA- II	MOEA/D	SPEA2
ZDT1	6.12E-04	6.24E-04	4.55E-03	8.18E-03	3.88E-03
ZDT2	1.72E-02	1.55E-02	4.74E-03	9.09E-03	3.89E-03
ZDT3	9.58E-04	9.87E-04	3.46E-02	1.73E-02	7.83E-03
ZDT4	7.59E-02	2.08E-01	5.93E-03	2.63E-02	5.07E-03
ZDT6	2.87E-03	1.76E-03	3.68E-03	6.47E-03	3.15E-03

Table 14: Average IGD results of version 2 and other MOEAs on UF problems

	mnv-MODE v2	MODE	MOEA/D	NSGA-II LS	MOEA/D -GM	GDE3	DMOEA-DD	MTS	LiuLi algorithm	DECMOSA -SQP	AMGA	clustering MOEA	MOEP	OWMOSaDE
UF1	7.42E-04	2.82E-03	4.35E-03	1.15E-02	6.20E-03	5.34E-03	1.04E-02	6.47E-03	7.85E-03	7.70E-02	3.59E-02	2.99E-02	5.96E-02	1.22E-02
UF2	9.12E-04	1.19E-03	6.79E-03	1.24E-02	6.40E-03	1.20E-02	6.79E-03	6.16E-03	1.23E-02	2.83E-02	1.62E-02	2.28E-02	1.89E-02	8.10E-03
UF3	7.31E-03	8.51E-03	7.42E-03	1.06E-01	4.29E-02	1.06E-01	3.34E-02	5.31E-02	1.50E-02	9.35E-02	7.00E-02	5.49E-02	9.90E-02	1.03E-01
UF4	1.25E-03	2.19E-03	6.39E-02	5.84E-02	4.76E-02	2.65E-02	4.27E-02	2.36E-02	4.35E-02	3.39E-02	4.06E-02	5.85E-02	4.27E-02	5.13E-02
UF5	8.11E-02	2.25E-01	1.81E-01	5.66E-01	1.79E+00	3.93E-02	3.15E-01	1.49E-02	1.62E-01	1.67E-01	9.41E-02	2.47E-01	2.25E-01	4.30E-01
UF6	8.25E-03	1.46E-02	5.87E-03	3.10E-01	5.56E-01	2.51E-01	6.67E-02	5.92E-02	1.76E-01	1.26E-01	1.29E-01	8.71E-02	1.03E-01	1.92E-01
UF7	4.26E-03	1.35E-02	4.44E-03	2.13E-02	7.60E-03	2.52E-02	1.03E-02	4.08E-02	7.30E-03	2.42E-02	5.71E-02	2.23E-02	1.97E-02	5.85E-02
UF8	5.51E-03	5.60E-03	5.84E-02	8.63E-02	2.45E-01	2.49E-01	6.84E-02	1.13E-01	8.24E-02	2.16E-01	1.71E-01	2.38E-01	4.23E-01	9.45E-02
UF9	4.44E-03	5.05E-03	7.90E-02	7.19E-02	1.88E-01	8.25E-02	4.90E-02	1.14E-01	9.39E-02	1.41E-01	1.89E-01	2.93E-01	3.42E-01	9.83E-02
UF10	5.11E-03	2.07E-02	4.74E-01	8.45E-01	5.65E-01	4.33E-01	3.22E-01	1.53E-01	4.47E-01	3.70E-01	3.24E-01	4.11E-01	3.62E-01	7.43E-01

Table 15: Average IGD results of version 2 and other MOEAs on MaOP problems

	mnv-MODE v2	MODE	MOEA/D	NSGA-III
MaOP1*	5.19E-01	5.89E-01	5.47E-01	5.12E-01
MaOP2*	4.51E-03	2.13E-02	1.14E-02	5.38E-03
MaOP3*	2.32E+00	3.16E+00	4.67E-01	2.62E+00
MaOP4*	2.09E-02	2.29E-02	2.09E-02	2.13E-02
MaOP5*	1.95E-02	1.98E-02	8.07E-02	1.27E-02
MaOP6*	1.69E-02	2.57E-02	5.71E-02	1.16E-02
MaOP7*	1.06E-02	3.18E-02	4.20E-02	2.27E-02
MaOP8*	1.06E-02	3.14E-02	3.68E-02	2.34E-02
MaOP9*	1.89E-02	5.37E-02	6.51E-02	3.36E-02
MaOP10*	1.12E-02	4.42E-02	3.91E-02	3.42E-02
MaOP1**	4.10E-01	4.88E-01	4.49E-01	4.89E-01
MaOP2**	2.53E-03	1.83E-02	4.56E-03	6.72E-03
MaOP3**	1.19E+00	2.10E+00	1.16E-02	2.44E+00
MaOP4**	9.91E-03	9.91E-03	1.02E-02	1.14E-02
MaOP5**	9.59E-03	9.55E-03	4.26E-02	2.23E-02
MaOP6**	1.14E-02	2.07E-02	3.14E-02	9.52E-02
MaOP7**	5.71E-03	5.78E-03	1.28E-02	1.19E-02
MaOP8**	6.50E-03	6.23E-03	1.14E-02	1.59E-02
MaOP9**	5.71E-03	5.48E-03	1.93E-02	2.38E-02
MaOP10**	4.62E-03	4.58E-03	1.69E-02	3.82E-02

* 3objectives
**5objectives

Table 16: Average IGD results of version 3 and other MOEAs on ZDT problems

	mnv-MODE v3	MODE	NSGA- II	MOEA/D	SPEA2
ZDT1	3.99E-04	6.24E-04	4.55E-03	8.18E-03	3.88E-03
ZDT2	1.82E-03	1.55E-02	4.74E-03	9.09E-03	3.89E-03
ZDT3	9.16E-04	9.87E-04	3.46E-02	1.73E-02	7.83E-03
ZDT4	1.96E-02	2.08E-01	5.93E-03	2.63E-02	5.07E-03
ZDT6	5.44E-04	1.76E-03	3.68E-03	6.47E-03	3.15E-03

Table 17: Average IGD results of version 3 and other MOEAs on UF problems

	mnv-MODE v3	MODE	MOEA/D	NSGA-II LS	MOEA/D-GM	GDE3	DMOEA-DD	MTS	LiuLi algorithm	DECMOSA-SQP	AMGA	clustering MOEA	MOEP	OWMOSaDE
UF1	7.03E-04	2.82E-03	4.35E-03	1.15E-02	6.20E-03	5.34E-03	1.04E-02	6.47E-03	7.85E-03	7.70E-02	3.59E-02	2.99E-02	5.96E-02	1.22E-02
UF2	5.86E-04	1.19E-03	6.79E-03	1.24E-02	6.40E-03	1.20E-02	6.79E-03	6.16E-03	1.23E-02	2.83E-02	1.62E-02	2.28E-02	1.89E-02	8.10E-03
UF3	6.02E-03	8.51E-03	7.42E-03	1.06E-01	4.29E-02	1.06E-01	3.34E-02	5.31E-02	1.50E-02	9.35E-02	7.00E-02	5.49E-02	9.90E-02	1.03E-01
UF4	1.24E-03	2.19E-03	6.39E-02	5.84E-02	4.76E-02	2.65E-02	4.27E-02	2.36E-02	4.35E-02	3.39E-02	4.06E-02	5.85E-02	4.27E-02	5.13E-02
UF5	7.19E-02	2.25E-01	1.81E-01	5.66E-01	1.79E+00	3.93E-02	3.15E-01	1.49E-02	1.62E-01	1.67E-01	9.41E-02	2.47E-01	2.25E-01	4.30E-01
UF6	9.74E-03	1.46E-02	5.87E-03	3.10E-01	5.56E-01	2.51E-01	6.67E-02	5.92E-02	1.76E-01	1.26E-01	1.29E-01	8.71E-02	1.03E-01	1.92E-01
UF7	3.53E-03	1.35E-02	4.44E-03	2.13E-02	7.60E-03	2.52E-02	1.03E-02	4.08E-02	7.30E-03	2.42E-02	5.71E-02	2.23E-02	1.97E-02	5.85E-02
UF8	2.77E-03	5.60E-03	5.84E-02	8.63E-02	2.45E-01	2.49E-01	6.84E-02	1.13E-01	8.24E-02	2.16E-01	1.71E-01	2.38E-01	4.23E-01	9.45E-02
UF9	1.71E-03	5.05E-03	7.90E-02	7.19E-02	1.88E-01	8.25E-02	4.90E-02	1.14E-01	9.39E-02	1.41E-01	1.89E-01	2.93E-01	3.42E-01	9.83E-02
UF10	6.56E-03	2.07E-02	4.74E-01	8.45E-01	5.65E-01	4.33E-01	3.22E-01	1.53E-01	4.47E-01	3.70E-01	3.24E-01	4.11E-01	3.62E-01	7.43E-01

Table 18: Average IGD results of version 3 and other MOEAs on MaOP problems

	mnv-MODE v3	MODE	MOEA/D	NSGA-III
MaOP1*	5.05E-01	5.89E-01	5.47E-01	5.12E-01
MaOP2*	3.76E-03	2.13E-02	1.14E-02	5.38E-03
MaOP3*	2.16E+00	3.16E+00	4.67E-01	2.62E+00
MaOP4*	2.09E-02	2.29E-02	2.09E-02	2.13E-02
MaOP5*	1.74E-02	1.98E-02	8.07E-02	1.27E-02
MaOP6*	1.58E-02	2.57E-02	5.71E-02	1.16E-02
MaOP7*	8.82E-03	3.18E-02	4.20E-02	2.27E-02
MaOP8*	8.27E-03	3.14E-02	3.68E-02	2.34E-02
MaOP9*	1.61E-02	5.37E-02	6.51E-02	3.36E-02
MaOP10*	9.42E-03	4.42E-02	3.91E-02	3.42E-02
MaOP1**	3.95E-01	4.88E-01	4.49E-01	4.89E-01
MaOP2**	2.48E-03	1.83E-02	4.56E-03	6.72E-03
MaOP3**	1.32E+00	2.10E+00	1.16E-02	2.44E+00
MaOP4**	2.07E-02	9.91E-03	1.02E-02	1.14E-02
MaOP5**	9.43E-03	9.55E-03	4.26E-02	2.23E-02
MaOP6**	9.18E-03	2.07E-02	3.14E-02	9.52E-02
MaOP7**	4.05E-03	5.78E-03	1.28E-02	1.19E-02
MaOP8**	4.72E-03	6.23E-03	1.14E-02	1.59E-02
MaOP9**	5.13E-03	5.48E-03	1.93E-02	2.38E-02
MaOP10**	4.23E-03	4.58E-03	1.69E-02	3.82E-02

* 3objectives
**5objectives

Table 19: Average IGD results of version 4 and other MOEAs on ZDT problems

	mnv-MODE v4	MODE	NSGA- II	MOEA/D	SPEA2
ZDT1	3.760E-04	6.244E-04	4.55E-03	8.18E-03	3.88E-03
ZDT2	1.915E-03	1.555E-02	4.74E-03	9.09E-03	3.89E-03
ZDT3	9.959E-04	9.874E-04	3.46E-02	1.73E-02	7.83E-03
ZDT4	5.780E-03	2.081E-01	5.93E-03	2.63E-02	5.07E-03
ZDT6	6.220E-04	1.756E-03	3.68E-03	6.47E-03	3.15E-03

Table 20: Average IGD results of version 4 and other MOEAs on UF problems

	mnv-MODE v4	MODE	MOEA/D	NSGA-II LS	MOEA/D -GM	GDE3	DMOEA-DD	MTS	LiuLi algorithm	DECMOSA -SQP	AMGA	clustering MOEA	MOEP	OWMOSaDE
UF1	7.155E-04	2.817E-03	4.35E-03	1.15E-02	6.20E-03	5.34E-03	1.04E-02	6.47E-03	7.85E-03	7.70E-02	3.59E-02	2.99E-02	5.96E-02	1.22E-02
UF2	6.262E-04	1.187E-03	6.79E-03	1.24E-02	6.40E-03	1.20E-02	6.79E-03	6.16E-03	1.23E-02	2.83E-02	1.62E-02	2.28E-02	1.89E-02	8.10E-03
UF3	3.698E-03	8.508E-03	7.42E-03	1.06E-01	4.29E-02	1.06E-01	3.34E-02	5.31E-02	1.50E-02	9.35E-02	7.00E-02	5.49E-02	9.90E-02	1.03E-01
UF4	1.249E-03	2.190E-03	6.39E-02	5.84E-02	4.76E-02	2.65E-02	4.27E-02	2.36E-02	4.35E-02	3.39E-02	4.06E-02	5.85E-02	4.27E-02	5.13E-02
UF5	3.850E-02	2.251E-01	1.81E-01	5.66E-01	1.79E+00	3.93E-02	3.15E-01	1.49E-02	1.62E-01	1.67E-01	9.41E-02	2.47E-01	2.25E-01	4.30E-01
UF6	5.299E-03	1.463E-02	5.87E-03	3.10E-01	5.56E-01	2.51E-01	6.67E-02	5.92E-02	1.76E-01	1.26E-01	1.29E-01	8.71E-02	1.03E-01	1.92E-01
UF7	3.049E-03	1.348E-02	4.44E-03	2.13E-02	7.60E-03	2.52E-02	1.03E-02	4.08E-02	7.30E-03	2.42E-02	5.71E-02	2.23E-02	1.97E-02	5.85E-02
UF8	2.662E-03	5.600E-03	5.84E-02	8.63E-02	2.45E-01	2.49E-01	6.84E-02	1.13E-01	8.24E-02	2.16E-01	1.71E-01	2.38E-01	4.23E-01	9.45E-02
UF9	1.572E-03	5.050E-03	7.90E-02	7.19E-02	1.88E-01	8.25E-02	4.90E-02	1.14E-01	9.39E-02	1.41E-01	1.89E-01	2.93E-01	3.42E-01	9.83E-02
UF10	4.458E-03	2.068E-02	4.74E-01	8.45E-01	5.65E-01	4.33E-01	3.22E-01	1.53E-01	4.47E-01	3.70E-01	3.24E-01	4.11E-01	3.62E-01	7.43E-01

Table 21: Average IGD results of version 4 and other MOEAs on MaOP problems

	mnv-MODE v4	MODE	MOEA/D	NSGA-III
MaOP1*	5.071E-01	5.886E-01	5.47E-01	5.12E-01
MaOP2*	3.494E-03	2.128E-02	1.14E-02	5.38E-03
MaOP3*	2.232E+00	3.159E+00	4.67E-01	2.62E+00
MaOP4*	2.090E-02	2.288E-02	2.09E-02	2.13E-02
MaOP5*	1.746E-02	1.985E-02	8.07E-02	1.27E-02
MaOP6*	1.408E-02	2.567E-02	5.71E-02	1.16E-02
MaOP7*	7.530E-03	3.177E-02	4.20E-02	2.27E-02
MaOP8*	8.640E-03	3.142E-02	3.68E-02	2.34E-02
MaOP9*	1.567E-02	5.366E-02	6.51E-02	3.36E-02
MaOP10*	9.185E-03	4.425E-02	3.91E-02	3.42E-02
MaOP1**	3.946E-01	4.881E-01	4.49E-01	4.89E-01
MaOP2**	2.430E-03	1.831E-02	4.56E-03	6.72E-03
MaOP3**	1.323E+00	2.099E+00	1.16E-02	2.44E+00
MaOP4**	9.824E-03	9.912E-03	1.02E-02	1.14E-02
MaOP5**	9.320E-03	9.552E-03	4.26E-02	2.23E-02
MaOP6**	8.606E-03	2.066E-02	3.14E-02	9.52E-02
MaOP7**	4.181E-03	5.782E-03	1.28E-02	1.19E-02
MaOP8**	4.312E-03	6.233E-03	1.14E-02	1.59E-02
MaOP9**	5.057E-03	5.482E-03	1.93E-02	2.38E-02
MaOP10**	4.221E-03	4.584E-03	1.69E-02	3.82E-02

* 3objectives
**5objectives

Considering the ZDT problems in Table 10 and Table 16, mnv-MODE v1 (in table 10) and mnv-MODE v3 (in table 16) have shown the best performance on all ZDT test instances except ZDT 4 in which SPEA2 is the winner competitor. Table 11 demonstrate that mnv-MODE v1 performs best on UF1-UF4 and all tri-objective UF problems while MOEA/D performs best on UF6-UF7, and MTS on UF5. It can be seen in table 12 that mnv-MODE v1 has the best performance on 13 test instances: 6 of them are tri-objective MaOP problems and the rest are 5-objective MaOP problems. NSGA-III can be considered as the second best algorithm in table 12 since it has the best IGD results on the tri-objective MaOP1, MaOP5 and MaOP6. Moreover, each one of MODE and MOEA/D performs best only in 2 MaOP test instances. Table 13 demonstrates that each of mnv-MODE v2 and SPEA2 algorithms performs best on two ZDT problems while MODE gets the smallest IGD value on ZDT6. It is clearly shown in table 14 and table 17 that all MOEAs are beaten either by mnv-MODE v2 (in table 14) or by mnv-MODE v3 (in table 17) on all UF test instances except UF5 and UF6. It was observed in table 15 that mnv-MODE v2 has the best performance on 11 test instances and NSGA-III was the winner competitor only on 3 test instances while MODE is the best on 4 test instances. Tables 18 shows that mnv-MODE v3 has the first rank among the four competitors on 14 test instances. Moreover, it is observed in tables 19-21 that mnv-MODE v4 yields the smallest IGD values on 16 MaOP test instances, all ZDT problems except ZDT3 and ZDT4, and on all UF problems except UF5.

It was also noticed in tables 12, 15, 18 and 21 that MOEA/D has the smallest IGD values on both 3-objective and 5-objective MaOP3 while NSGA-III achieved the best performance on the tri-objective MaOP5-MaOP6. From the tables 11, 14, 17 and 20, we can observe that all our proposed methods are beaten by MTS algorithm on UF5. Tables 10,13,15 and 19 illustrates that SPEA2 outperform the other competitors only on ZDT4.

4.4 Friedman Rank testing

Friedman test is a non-parametric statistical multiple comparisons test utilized to rank the compared algorithms on the basis of their average fitness for each benchmark problem. Afterward, the mean rank of each algorithm is determined using the Friedman test [68, 69]. Table 22 shows that the calculated probability P-value of this test is less than the significance level (0.05) which indicates that the existence of significant statistical difference among the results of the compared methods [68, 70, 71]. The average (mean) rank of MODE and the four proposed algorithms is illustrated also in table 22. Figure 15 and table 22 shows that method mnv-MODE v4 is significantly different from all the other methods and it has minimum rank.

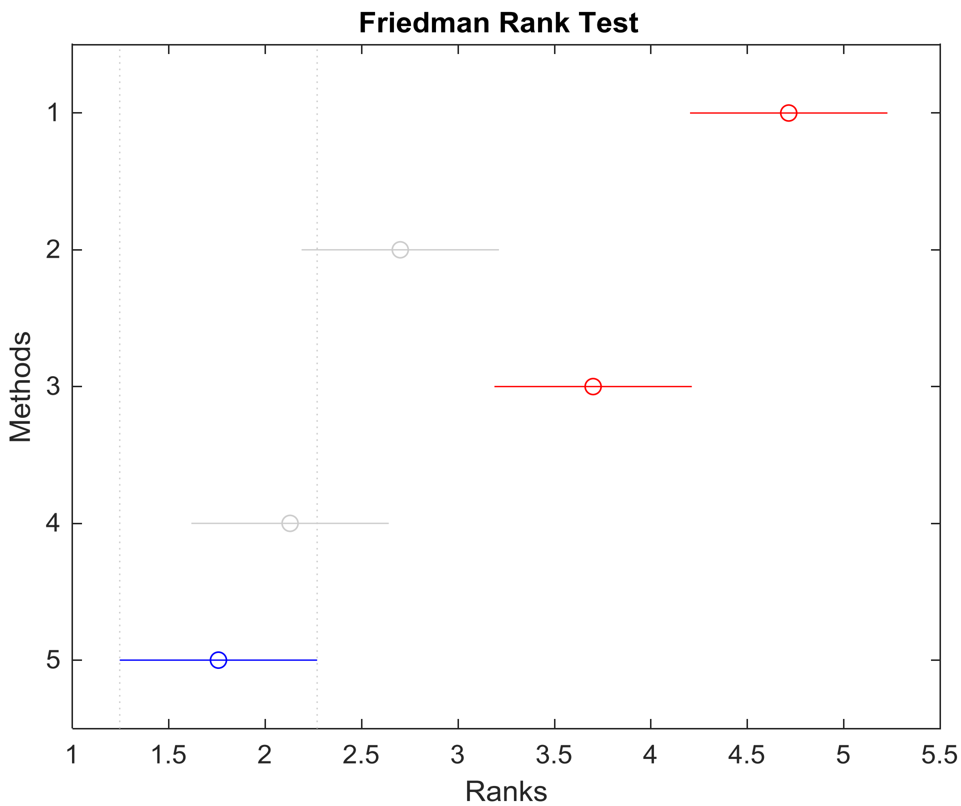


Figure 15: Friedman Rank test results for methods MODE(1), mnv-MODE v1 (2), mnv-MODE v2 (3), mnv-MODE v3 (4) and mnv-MODE v4 (5) (p-value = 4.6609e-17).

Table 22: Friedman Rank test results for MODE and our proposed methods

Algorithm	**mean rank**
MODE	4.7143
mnv-MODE v1	2.7
mnvMODE v2	3.7
mnv-MODE v3	2.1286
mnv-MODE v4	1.7571
p-value = 4.6609e-17	

Chapter 5

CONCLUSION

This work introduced a new approach called mnv-MODE which targets to solve/ for the aim of solving 25 multi-objective benchmark problems with 2, 3 and 5 objectives. These problems are ZDT1-ZDT4, ZDT6, UF1-UF10 and MaOP1-MaOP10. Four different versions of mnv-MODE were developed. MODE and tabu search are utilized in these versions. The finding of mnv-MODE versions and the previous algorithms that solved these problems are compared according to their IGD values. Friedman ranking test is then used to select the best 5 methods and arrange them.

The results illustrates that mnv-MODE v4 has the better performance in the majority of the test instances. Obtained results for mnv-MODE v4 also shows good convergence towards the true pareto front. Moreover, the divergence of the solutions in the pareto-front obtained by mnv-MODE v4 is obvious. However, mnv-MODE v4 was beaten by other MOEAs on some test instances. This means that much work and time should be devoted to improve this algorithm.

REFERENCES

[1] Du, K. L., & Swamy, M. N. S. (2016). Search and optimization by metaheuristics. Techniques and Algorithms Inspired by Nature; Birkhauser: Basel, Switzerland.

[2] Burke, E. K., & Kendall, G. (Eds.). (2013). Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques. Springer Science & Business Media.

[3] Yann Collette, Professor Patrick Siarry (2004).Multiobjective Optimization_ Principles and Case Studies. Springer-Verlag Berlin Heidelberg.

[4] Deb, K. (2001). Multi-objective optimization using evolutionary algorithms (Vol. 16). John Wiley & Sons.

[5] Sbalzarini, I. F., Müller, S., & Koumoutsakos, P. (2000, November). Multiobjective optimization using evolutionary algorithms. In Proceedings of the summer Program (Vol. 2000, pp. 63-74).

[6] Deb, K. (2011). Multi-objective optimisation using evolutionary algorithms: an introduction. In Multi-objective evolutionary optimisation for product design and manufacturing (pp. 3-34). Springer, London.

[7] Abraham, A., & Jain, L. (2005). Evolutionary multiobjective optimization. In Evolutionary Multiobjective Optimization (pp. 1-6). Springer, London.

[8] Coello, C. C., Lamont, G. B., & van Veldhuizen, D. A. (2007). Evolutionary Algorithms for Solving Multi-Objective Problems. Springer Science & Business Media.

[9] Jaimes, A. L., Martınez, S. Z., & Coello, C. A. C. (2009). An introduction to multiobjective optimization techniques. Optimization in Polymer Processing, 29-57.

[10] Yu, X., & Gen, M. (2010). Introduction to evolutionary algorithms. Springer Science & Business Media.

[11] Coello-Coello, C. A. (2014, November). An Introduction to Evolutionary Multi-Objective Optimization with Some Applications in Pattern Recognition. In Iberoamerican Congress on Pattern Recognition (pp. 1-13). Springer, Cham.

[12] Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.

[13] Babu, B. V., Mubeen, J. S., & Chakole, P. G. (2005). Multiobjective optimization using differential evolution. TechGenesis-The Journal of Information Technology, 2(2), 4-12.

[14] Gandibleux, X., Sevaux, M., Sörensen, K., & T'kindt, V. (Eds.). (2004). Metaheuristics for multiobjective optimisation (Vol. 535). Springer Science & Business Media

[15] Zhang, X., Zheng, X., Cheng, R., Qiu, J., & Jin, Y. (2018). A competitive mechanism based multi-objective particle swarm optimizer with fast convergence. Information Sciences, 427, 63-76.

[16] Shim, V. A., Tan, K. C., & Tang, H. (2015). Adaptive Memetic Computing for Evolutionary Multiobjective Optimization. IEEE Transactions on Cybernetics, 4(45), 610-621.

[17] Sindhya, K., Miettinen, K., & Deb, K. (2012). A hybrid framework for evolutionary multi-objective optimization. IEEE Transactions on Evolutionary Computation, 17(4), 495-511.

[18] Sindhya, K., Sinha, A., Deb, K., & Miettinen, K. (2009, May). Local search based evolutionary multi-objective optimization algorithm for constrained and unconstrained problems. In 2009 IEEE Congress on Evolutionary Computation (pp. 2919-2926). IEEE.

[19] Wilamowski, B. M., & Irwin, J. D. (Eds.). (2011). The industrial electronics handbook: Intelligent Systems. Boca Raton, Florida, United States: CRC Press.

[20] Tian, Y., Zhang, X., Cheng, R., & Jin, Y. (2016, July). A multi-objective evolutionary algorithm based on an enhanced inverted generational distance metric. In 2016 IEEE Congress on Evolutionary Computation (CEC) (pp. 5222-5229). IEEE.

[21] Lin, Q., Zhu, Q., Wang, N., Huang, P., Wang, W., Chen, J., & Ming, Z. (2018). A multi-objective immune algorithm with dynamic population strategy. Swarm and Evolutionary Computation.

[22] Mezura-Montes, E., Reyes-Sierra, M., & Coello, C. A. C. (2008). Multi-objective optimization using differential evolution: a survey of the state-of-the-art. In Advances in differential evolution (pp. 173-196). Springer, Berlin, Heidelberg.

[23] Huang, Z., & Chen, Y. (2013). An improved differential evolution algorithm based on adaptive parameter. Journal of Control Science and Engineering, 2013, 3.

[24] Menon, P. P., Bates, D. G., Postlethwaite, I., Marcos, A., Fernandez, V., & Bennani, S. (2008). Worst case analysis of control law for re-entry vehicles using hybrid differential evolution. In Advances in Differential Evolution (pp. 319-333). Springer, Berlin, Heidelberg.

[25] Qin, A. K., Huang, V. L., & Suganthan, P. N. (2008). Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE transactions on Evolutionary Computation, 13(2), 398-417.

[26] Suganthan, P. N. (2012, October). Differential evolution algorithm: recent advances. In International Conference on Theory and Practice of Natural Computing (pp. 30-46). Springer, Berlin, Heidelberg.

[27] Price, K., Storn, R. M., & Lampinen, J. A. (2006). Differential evolution: a practical approach to global optimization. Springer Science & Business Media.

[28] Babu, B. V., & Jehan, M. M. L. (2003, December). Differential evolution for multi-objective optimization. In The 2003 Congress on Evolutionary Computation, 2003. CEC'03. (Vol. 4, pp. 2696-2703). IEEE.

[29] Xiang, W. L., Meng, X. L., An, M. Q., Li, Y. Z., & Gao, M. X. (2015). An enhanced differential evolution algorithm based on multiple mutation strategies. Computational intelligence and neuroscience, 2015, 2.

[30] Robič, T., & Filipič, B. (2005, March). Differential evolution for multiobjective optimization. In International Conference on Evolutionary Multi-Criterion Optimization (pp. 520-533). Springer, Berlin, Heidelberg.

[31] Saha, I., Maullik, U., Łukasik, M., & Plewczynski, D. (2014). Multiobjective differential evolution: a comparative study on benchmark problems. In Man-Machine Interactions 3 (pp. 529-536). Springer, Cham.

[32] Babu, B. V., Mubeen, J. S., & Chakole, P. G. (2005). Multiobjective optimization using differential evolution. TechGenesis-The Journal of Information Technology, 2(2), 4-12.

[33] Hromkovič, J. (2013). Algorithmics for hard problems: introduction to combinatorial optimization, randomization, approximation, and heuristics. Springer Science & Business Media.

[34] Michiels, W., Aarts, E., & Korst, J. (2007). Theoretical aspects of local search. Springer Science & Business Media.

[35] Du, D. Z. (2005). Handbook of Combinatorial Optimization: Supplement Volume B.

[36] Glover, F. W., & Kochenberger, G. A. (Eds.). (2003). Handbook of Metaheuristics (Vol. 57). Springer Science & Business Media.

[37] Okabe, T., Jin, Y., Olhofer, M., & Sendhoff, B. (2004, September). On test functions for evolutionary multi-objective optimization. In International Conference on Parallel Problem Solving from Nature (pp. 792-802). Springer, Berlin, Heidelberg.

[38] Cheng, R., Li, M., Tian, Y., Zhang, X., Yang, S., Jin, Y., & Yao, X. (2017). A benchmark test suite for evolutionary many-objective optimization. Complex & Intelligent Systems, 3(1), 67-81.

[39] Chase, N., Rademacher, M., Goodman, E., Averill, R., & Sidhu, R. (2009). A benchmark study of multi-objective optimization methods. BMK-3021, Rev, 6, 1-24.

[40] Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. Evolutionary computation, 8(2), 173-195.

[41] Huband, S., Hingston, P., Barone, L., & While, L. (2006). A review of multiobjective test problems and a scalable test problem toolkit. IEEE Transactions on Evolutionary Computation, 10(5), 477-506.

[42] Zhang, Q., Zhou, A., Zhao, S., Suganthan, P. N., Liu, W., & Tiwari, S. (2008). Multiobjective optimization test instances for the CEC 2009 special session and competition. University of Essex, Colchester, UK and Nanyang technological University, Singapore, special session on performance assessment of multi-objective optimization algorithms, technical report, 264.

[43] Li, H., Deb, K., Zhang, Q., & Suganthan, P. N. (2018). Challenging Novel Many and Multi-Objective Bound Constrained Benchmark Problems (pp. 1-10). Technical Report, updated 11 Jan.

[44] Li, H., Deb, K., Zhang, Q., Suganthan, P. N., & Chen, L. (2019). Comparison between MOEA/D and NSGA-III on a set of novel many and multi-objective benchmark problems with challenging difficulties. Swarm and Evolutionary Computation, 46, 104-117.

[45] Coello, C. A. C. (2003). Evolutionary multi-objective optimization: A critical review. In Evolutionary optimization (pp. 117-146). Springer, Boston, MA.

[46] Chang Wook, A. (1860). Advances in Evolutionary Algorithms: Theory, Design and Practice.

[47] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. A. M. T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE transactions on evolutionary computation, 6(2), 182-197.

[48] DEB, K., & JAIN, H. (2014). An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. IEEE transactions on evolutionary computation, 18(4), 577-601.

[49] Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength Pareto evolutionary algorithm. TIK-report, 103.

[50] Zitzler, E., & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. IEEE transactions on Evolutionary Computation, 3(4), 257-271.

[51] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on evolutionary computation, 11(6), 712-731.

[52] Li, H., & Zhang, Q. (2009). Multiobjective optimization problems with complicated Pareto sets, MOEA/D and NSGA-II. IEEE transactions on evolutionary computation, 13(2), 284.

[53] Chen, C. M., Chen, Y. P., & Zhang, Q. (2009, May). Enhancing MOEA/D with guided mutation and priority update for multi-objective optimization. In 2009 IEEE Congress on Evolutionary Computation (pp. 209-216). IEEE.

[54] Zhang, Q., & Suganthan, P. N. (2009). Final report on CEC'09 MOEA competition. In Congress on evolutionary computation (CEC 2009).

[55] Kukkonen, S., & Lampinen, J. (2009, May). Performance assessment of generalized differential evolution 3 with a given set of constrained multi-objective test problems. In 2009 IEEE Congress on Evolutionary Computation (pp. 1943-1950). IEEE.

[56] Liu, M., Zou, X., Chen, Y., & Wu, Z. (2009, May). Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances. In 2009 IEEE Congress on Evolutionary Computation (pp. 2913-2918). IEEE.

[57] Tseng, L. Y., & Chen, C. (2009, May). Multiple trajectory search for unconstrained/constrained multi-objective optimization. In 2009 IEEE Congress on Evolutionary Computation (pp. 1951-1958). IEEE.

[58] Liu, H. L., & Li, X. (2009, May). The multiobjective evolutionary algorithm based on determined weight and sub-regional search. In 2009 IEEE Congress on Evolutionary Computation(pp. 1928-1934). IEEE.

[59] Huang, V. L., Zhao, S. Z., Mallipeddi, R., & Suganthan, P. N. (2009, May). Multi-objective optimization using self-adaptive differential evolution algorithm. In 2009 IEEE Congress on Evolutionary Computation (pp. 190-194). IEEE.

[60] Wang, Y., Dang, C., Li, H., Han, L., & Wei, J. (2009, May). A clustering multi-objective evolutionary algorithm based on orthogonal and uniform design. In 2009 IEEE congress on evolutionary computation (pp. 2927-2933). IEEE.

[61] Tiwari, S., Fadel, G., Koch, P., & Deb, K. (2009, May). Performance assessment of the hybrid archive-based micro genetic algorithm (AMGA) on the CEC09 test problems. In 2009 IEEE Congress on Evolutionary Computation (pp. 1935-1942). IEEE.

[62] Qu, B. Y., & Suganthan, P. N. (2009, May). Multi-objective evolutionary programming without non-domination sorting is up to twenty times faster. In 2009 IEEE Congress on Evolutionary Computation (pp. 2934-2939). IEEE.

[63] Zamuda, A., Brest, J., Boskovic, B., & Zumer, V. (2009, May). Differential evolution with self-adaptation and local search for constrained multiobjective optimization. In 2009 IEEE Congress on Evolutionary Computation (pp. 195-202). IEEE.

[64] Yen, G. G., & He, Z. (2013). Performance metric ensemble for multiobjective evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 18(1), 131-144.

[65] Riquelme, N., Von Lücken, C., & Baran, B. (2015, October). Performance metrics in multi-objective optimization. In 2015 Latin American Computing Conference (CLEI) (pp. 1-11). IEEE.

[66] Schutze, O., Esquivel, X., Lara, A., & Coello, C. A. C. (2012). Using the averaged Hausdorff distance as a performance measure in evolutionary multiobjective optimization. IEEE Transactions on Evolutionary Computation, 16(4), 504-522.

[67] Ishibuchi, H., Masuda, H., Tanigaki, Y., & Nojima, Y. (2015, March). Modified distance calculation in generational distance and inverted generational distance. In International Conference on Evolutionary Multi-Criterion Optimization (pp. 110-125). Springer, Cham.

[68] Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1(1), 3-18.

[69] Ibrahim, Z., Aziz, N. A., Aziz, N. A. A., Razali, S., Shapiai, M. I., Nawawi, S. W., & Mohamad, M. S. (2015). A Kalman filter approach for solving unimodal optimization problems. ICIC Express Lett, 9(12), 3415-3422.

[70] Abdelkhalik, O., & Darani, S. (2016, July). Hidden genes genetic algorithms for systems architecture optimization. In Proceedings of the Genetic and Evolutionary Computation Conference 2016 (pp. 629-636). ACM.

[71] Sherinov, Z., & Ünveren, A. (2018). Multi-objective imperialistic competitive algorithm with multiple non-dominated sets for the solution of global optimization problems. Soft Computing, 22(24), 8273-8288.