Cooperative Multi-Agent Ensembles for Multi-Objective Optimization

Jamshid Tamouk

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

Doctor of Philosophy
in
Computer Engineering

Eastern Mediterranean University
July 2021
Gazimağusa, North Cyprus

Approval of the Institute of Graduate Studies and Research

Prof. Dr. Ali Hakan Ulusoy
Director

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

Prof. Dr. Hadi 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 Doctor of Philosophy in Computer Engineering.

Assoc. Prof. Dr. Adnan Acan
Supervisor

Examining Committee

1. Prof. Dr. Ercan Öztemel
2. Assoc. Prof. Dr. Adnan Acan
3. Assoc. Prof. Dr. Mehmet Bodur
4. Assoc. Prof. Dr. Mohammed Salamah
5. Asst. Prof. Dr. Ahmet Ünveren

ABSTRACT

Multi-objective optimization (MOO) using metaheuristics is a hot research area that is receiving great interest of algorithm designers. In literature, use of multiple metaheuristic algorithms within a multi-agent framework or within an ensemble system were studied and significant performance improvements compared to single algorithm's successes is achieved. The first proposed architecture in this thesis study is a multi-metric and multi-deme multi-agent system which comprises several MOO method agents namely: MOGA, SPEA2, MODE, MOSA and MOPSO. The agents cooperate in consecutive sessions to discover and extract a feasible and high-quality Pareto fronts. The system divides the population to sub-populations and assigns them to metaheuristic agents in the beginning of sessions. Once after running metaheuristics, they return the optimized sub-populations to be used in next session and to update the global archive. Four metrics are used within the system in which three of them are multi-objective assessment metrics and one of them is metaheuristic performance measurement metric. The performance of metaheuristics is used by system to adjust their associated number of fitness evaluations. Also, it is used to accept or reject the improved sub-populations. The sub-populations are mixed to get the common population to be used in next session. Meanwhile, the non-dominated solutions of each sub-population are used to form the global archive. The global archive keeps all Non-Dominated Solutions discovered by all metaheuristics.

In the second architecture a dynamic metaheuristic network is proposed based on a layered platform. The network represents the MOO algorithms with nodes and flow of sub-populations by edges. The system operates in consecutive epochs in which each epoch begins with assigning sub-population to nodes continues with running each metaheuristic within its algorithmic framework. Afterwards the enhanced sub-populations are transferred to all forward linked nodes. At the end of each epoch the metaheuristic agents at layers are changed by a rotation operator. The proposed method contains seven different MOO metaheuristic algorithms formed as 3-3-1 network configuration with three layers. Enhanced sub-populations are transferred to the next layer nodes (one or more layers) when an epoch or session terminates. At the end of all sessions, all local and global Non-dominated individuals are merged. The evaluation of the proposed methods are carried out using a set of well-known benchmarks and the obtained results are compared to state-of-the-art methods. Likewise, it is noticed that the proposed methods outperform the existent state-of-the-arts methods in the majority of test problems.

Keywords: Multi-Agent Systems, Metaheuristics, Multi-Objective Optimization, Ensemble Systems, Multi-Objective Metrics, Dynamic Network Systems.

ÖZ

Üstsezgisel yöntemlere dayalı çok amaçlı eniyileme, algoritma tasarımcıları ve uygulamalı problemler üzerinde çalışanlar için büyük ilgi çeken aktif bir araştırma alanıdır. Birden çok üstsezgisel yöntemin bir çoklu ajan yapısı veya bir yöntemler birliği çerçevesinde kullanılması literatürde yapılan yayınlarda çalışılmış ve tek başına kullanılan yöntemlerin başarımı ile karşılaştırıldığında önemli oranda verimlilik artışına ulaşılmıştır. Bu doktora tezi çalışmasında önerilen birinci mimari, çok ölçülü ve çok topluluklu bir çoklu ajan sistemidir. Bu çoklu ajan sistemi MOGA, SPEA2, MODE, MOSA ve MOPSO üstsezgisel ajanlarından oluşur. Bu sistemde ajanlar ardışık seanlarda yardımlaşarak belirli sayıda en iyi veya en iyiye yakın Pareto cepheleri çıkarmaya çalışırlar. Her bir otorumun başlangıcında, toplam potansiyel çözüm nüfusu alt nüfuslara bölünerek çok amaçlı çözüm ajanlarına dağıtılır. Ajanlar kendilerine dağıtılan alt nüfuslar üzerinde çalıştıktan sonra bir sonraki seansta kullanılacak eniyilenmiş alt nüfularlarla birlikte hükmedilmemiş çözüm kümelerini geri iade ederler ve tümel arşivi yenilerler. Üçü çok amaçlı değerlendirme ölçevi olmak üzere sistemde dört ölçev kullanılmıştır, bir ölçev sezgisel ajanların başarımına yöneliktir. Ajanların belli bir ölçeve göre başarımları her ajan için tayin edilen amaç işlevi değerleme sayısını ayarlamak için kullanılır. Aynı zamanda, bir ajanın eniyilediği alt nüfus ajanın başarım seviyesine bağlı olarak reddedilebilir. Her seansın sonunda, eniyilenmiş alt nüfuslar birleştirilerek tümel nüfus oluşturulur ve hükmedilmemiş çözüm alt kümelerileri de birleştirilerek hükmedilmemiş çözümler kümesi (tümel Pareto cephesi) elde edilir.

İkinci mimari katmanlar halinde birbirlerine bağlanmış çok amaçlı eniyileme yöntemlerininden oluşan bir dinamik üstsezgisel ağ önerisidir. Ağın her düğümü bir

çok amaçlı üstsezgisel yönteme karşılık gelir ve düğümler arası bağlantılar alt nüfus elemanlarının ileri yönlü besleme ile akışını temsil ederler. Önerilen sitem ardışık seanslar halinde çalışır öyle ki, bir seans alt nüfusların ağ düğümlerine atanmasıyla başlar ve her üstsezgisel yöntemin kendi algorithmic çerçevesi içinde çalışmasıyla devam eder. Sonrasında geliştirilmiş alt nüfus elemanlarının ileri yönde bağlanmış komşu düğümlere aktarılır. Dinamik ağ mimarisi her seans sonunda katmanlardaki üstsezgisel ajanlar bir kaydırma işlemi ile değiştirilirler. Mevcut ugulama durumunda, önerilen yöntem yedi farklı çok amaçlı üstsezgisel algoritmanın üç katmanlı bir ağ üzerinde 3-3-1 topolojisi ile sıralanmasını içerir. Her seans sonunda eniyilenmiş alt nufüs elemanları ileri yönde bağlanmış komşu düğümlere aktarılırlar. Bütün seanslar tamamlandığında, yerel hükmedilmemiş çözüm kümeleri birleştirilir. Önerilen sistemlerin sınanması literatürde yayınlanan gerçel değerli çok amaçlı test problemleri kullanılarak sınanmıştır. Elde edilen sonuçlar geniş bir modern algoritmalar kümesindekilerle karşılaştırılmış ve önerilen yöntemlerin kullanılan test problemlerinin çoğunluğu mevcut algoritmalardan daha iyi olduğu gösterilmiştir.

Anahtar Kelimeler: Çok Ajanlı Sistemler, Üstsezgiseller, Çok Amaçlı Eniyileme, Takım Sistemleri, Çok Amaçlı Değerlendirme Ölçütleri, Dinamic Ağ Sistemleri.

DEDICATION

To my beloved brother who has been in a coma for seven years.
I hope he wakes up one day.

ACKNOWLEDGMENT

I would like to express my special thanks to my supervisor Assoc. Prof. Dr. Adnan Acan for providing great and valuable supervision throughout my study and research.

I greatly appreciate my committee members Assoc. Prof. Dr. Mehmet Bodur and Asst. Prof. Dr. Ahmet Ünveren for their valued suggestions and constructive criticism for the improvement of my research.

I really appreciate my external jury member, Prof. Dr. Ercan Öztemel and my substituted jury member Assoc. Prof. Dr. Mohammed Salamah for their presence and valuable suggestions.

Finally, I would like to deeply thank my parents and my wife for their great support and unconditional love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

$I_{\varepsilon+}$	ε–indicator
Δ*	Genetic Algorithm
AMOSA	A Multi-Objective Simulated Annealing
CEC	Congress on Evolutionary Computation
DE	Differential Evolution
DMN	Dynamic Metaheuristic Network
FAR	Friedman Aligned Ranks
GA	Genetic Algorithm
IGD	Inverted Generational Distance Metric
M^3D/MAS	Multi-Metric and Multi-Deme Multi-Agent System
MAS	Multi-Agent System
MOABC	Multi-Objective Artificial Bee Colony
MODE	Multi-Objective Differential Evolution
MOGA	Multi-Objective Genetic Algorithm
MOO	Multi-Objective Optimization
MOP	Multi-objective Optimization Problem
MOPSO	Multi-Objective Particle Swarm Optimization
NDS	Non-dominated solutions
NSGAII	Non-dominated Sorting Genetic Algorithm
POF	Pareto-Optimal Front
PSO	Particle Swarm Optimization
SPA	Solution Pool Agent
SPEA2	Strength Pareto Evolutionary Algorithm

Chapter 1

INTRODUCTION

1.1 Introduction

Most often the objectives within the real-world optimization problems are inconsistent. Hence, MOO approaches extract solution set which indicates trade-off solutions related to the problem [1]. Solving these types of problems are still considered as a hot and challenging topic, especially for engineering applications. Definition of multi-objective optimization problem (MOP) for real-valued optimization with m objectives on R^n is given as:

$$\begin{aligned}
&\min_{X \in D} F(X) \\
&\text{such that. } F(X) = (f_1(X), \dots, f_m(X)) \\
&X = (x_1, \dots, x_n) \in R^n \\
&x_i \in D_i \text{ and } D = D_1 \times \dots \times D_n
\end{aligned} \qquad (1)$$

On this point the Pareto-optimal is a set of all extracted non-dominated solutions based on problem objectives. Figure 1.1 depicts Pareto-optimal front and two estimated solution sets A and B. In this graph A and B are not comparable as far as Pareto dominance, but A is much preferable in comparison to B because the following reasons: 1- A is closer to the Pareto-optimal front (POF), 2- It is more extended, more populated and 3- It has better been distributed [2].

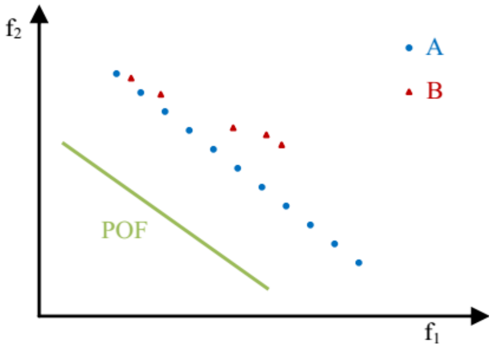


Figure 1.1: Pareto optimal front (POF) and estimated solutions (A and B) [2]

All non-dominated solutions belong to Pareto-optimal set. The Pareto-optimal set is declared as following. Assuming u and v as problem solutions, if solution u is better than solution v for minimum of one objective and it is equal to v for other objectives, u dominates v [1]. In other words, for a minimization purpose u dominates v if $f_i(u) \leq f_i(v)$, $\forall i$ and $\exists$ j for which $f_j(u) < f_j(v)$. Therefore, it can be said that u and v are non-dominated if none of them dominates another one. The meaning of solving an optimization problem is to extract a pareto-optimal set.

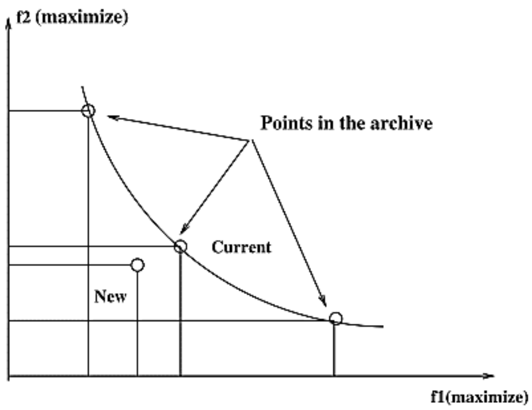


Figure 1.2: Dominated and non-dominated solutions [3]

This set consists of possible solutions (feasible/candidate solution is an optimal solution that the objective function reaches its best value) [3] in a way that all selected individuals in the set are non-dominated (If in comparing two solutions neither solution dominates another one) [3]. The Pareto Front is the set of non-dominated objective values [1]. In Figure 1.2 it is shown that the ‘Current’ (current solution) is dominated by ‘New’ (new solution). Therefore, ‘New’ is non-dominating with respect to the solutions of Archive.

Methods of finding solutions for MOO problems could be grouped in two categories, namely: exact methods (with the purpose of computing the complete optimal Pareto front), and, approximation methods (with the goal of extracting good solutions without guaranteeing the optimality) [4]. Deterministic approaches like dynamic programming are used to solve the problems with small size [4]; although, approximation methods became more popular due to having lower and much feasible computational time requirement in compared to exact methods. Metaheuristic based methods are popular for solving MOO problems due to their low computational complexity and their ability in discovering very good Pareto fronts when dealing with NP-hard problems. Some of these hard problems are knapsack problem, job shop scheduling problem, travelling salesman problem, Flight Path Optimization, Water Distribution Systems, Electrical Distribution Systems and so on [5]. A survey on multi-objective metaheuristics and their applications are given in [6] with detail. Some of the MOO metaheuristics that are well-known by their success (which all are also implemented and used in this research) are: Multi-Objective Differential Evolution (MODE) [11], Multi-Objective Genetic Algorithm (MOGA) [8, 9], Multi-Objective Simulated Annealing (AMOSA) [12], Strength Pareto Evolutionary Algorithm (SPEA 2) [10], Non-Dominated Sorting Genetic Algorithm (NSGA II) [7, 8], Multi-Objective Particle Swarm Optimization (MOPSO) [13, 14] and Multi-Objective Artificial Bee Colony [15]. The two novel methods proposed in this thesis introduce and implement architectures in which each MOO metaheuristics act as either an individual agent of a MAS or individual node of 3-layer network respectively in Chapter 2 and 3. Finally by utilizing the well-known benchmark problem set, the proposed methods are experimentally evaluated and results are compared to some state-of-the-art competitors.

The remaining of this research is categorized as follows: a novel architecture of MAS called M^3D/MAS (A Multi-Metric and Multi-Deme Multi-Agent System for Multi-Objective Optimization) is illustrated in Chapter 2. Chapter 3 presents a novel architecture called DMN (A Dynamic Metaheuristic Network for Numerical Multi-Objective Optimization) and finally Chapter 4 presents conclusions and some future research directions.

1.2 Challenges in Real Word Examples

There is no doubt that in real-life problems, the dataset/solution pool is not as clean as the benchmark problems used in this research. Therefore, it requires some preprocessing steps to deal with missing values (either by eliminating those samples or by replacing them with some acceptable values such as the mean or median value of the columns under consideration), spars datasets including many zeros, invalid type of values (e.g., having string value instead of a number), outliers (either by eliminating or replacing them with some acceptable). In such cases the data may require normalization, feature reduction, feature engineering/generating, feature selection, selecting a suitable technique for creating random or pre-designed initial population/solution pool, and many other preprocessing steps. Depending on dataset and scenarios, some of the mentioned methods can be carried out on dataset.

The preprocessing step can improve the performance of the optimizer either by increasing accuracy (lowering error/cost function) or by decreasing the time complexity (by speeding up of the optimization's process).

Chapter 2

A MULTI-METRIC AND MULTI-DEME MULTI-AGENT SYSTEM FOR MULTI-OBJECTIVE OPTIMIZATION

2.1 Introduction

A multi-agent system is a platform consisting of several agents in which the agents collaborate to fulfill the required tasks and reach the predefined goals. Hence, multi-agent systems for metaheuristics are designed in a way that each agent acts as a metaheuristic for solving complicated problems. Fundamentally, each agent collects perceptions from the environment, do processing on them by using its collected data, to discover the search space using the operators to achieve goals. Multi-agent systems can be classified into two classes: homogeneous and heterogeneous. In homogeneous systems all agents have same abilities and characteristics while in heterogeneous systems each agent can have specific characteristics [16]. In these two system types, all agents cooperate and compete together to carry out a particular task [17]. During the interactions, the system needs some agents to realize communication capabilities in order to transfer information between agents. Some related works regarding to multiagent systems are illustrated in following sections.

This chapter of thesis introduces an innovative multi-agent system for solving real-valued multi-objective optimization problems. The suggested architecture implements a MAS, so that all aforementioned multi-objective optimization algorithms (namely: NSGA II, MOGA, SPEA 2, MODE, AMOSA, MOPSO and MOABC) act as system agents. These agents cooperate together on related sub-populations in order to discover the solutions in optimal PF. Agent characteristics consist of data structures and sets of actions for the agents that are in charge. There is only one global population in the system which is divided into sub-populations. This is done by uniform random sampling to keep the same cardinality. The suggested MAS system operates in iterated sessions including two steps. The first step divides the global population to sub-populations so that one solution can be located in different sub-populations. Meanwhile, there is no repeated solution in a sub-population. Thereafter in the second step, MOO agents work on their own sub-populations to improve it and extract the Pareto-front. Afterwards, the system uses some measurement metrics to evaluate the discovered Pareto-fronts. These measurements manipulate the number of fitness evaluations based on the achieved performance. Also, the modified sub-populations can be rejected according to the results. In proposed MAS, all MOO agents have their own local archives. These archives keep all non-dominated individuals found in a session. As well as a global archive is used within the system to keep all non-dominated individuals found up to now by all MOO agents. All verified sub-populations are mixed to provide a new global population and local archives are applied to update the shared global archive. This way the MOO agents cooperate together to improve the global population and Pareto-front kept in global archive. Figure 2.1 shows the architecture of the proposed multi-agent system.

The IEEE CEC2009 which includes 10 test problems [18] is used to evaluate the proposed MAS system. According to the test results presented in Section 6, the suggested MAS finds better Pareto-fronts in compared to most of its competitors. Meanwhile, the ZDT and DTLZ benchmark sets are applied to evaluate the suggested MAS [8, 9]. The obtained results are again better than most of the competitors. Section 5 describes the suggested MAS in great details.

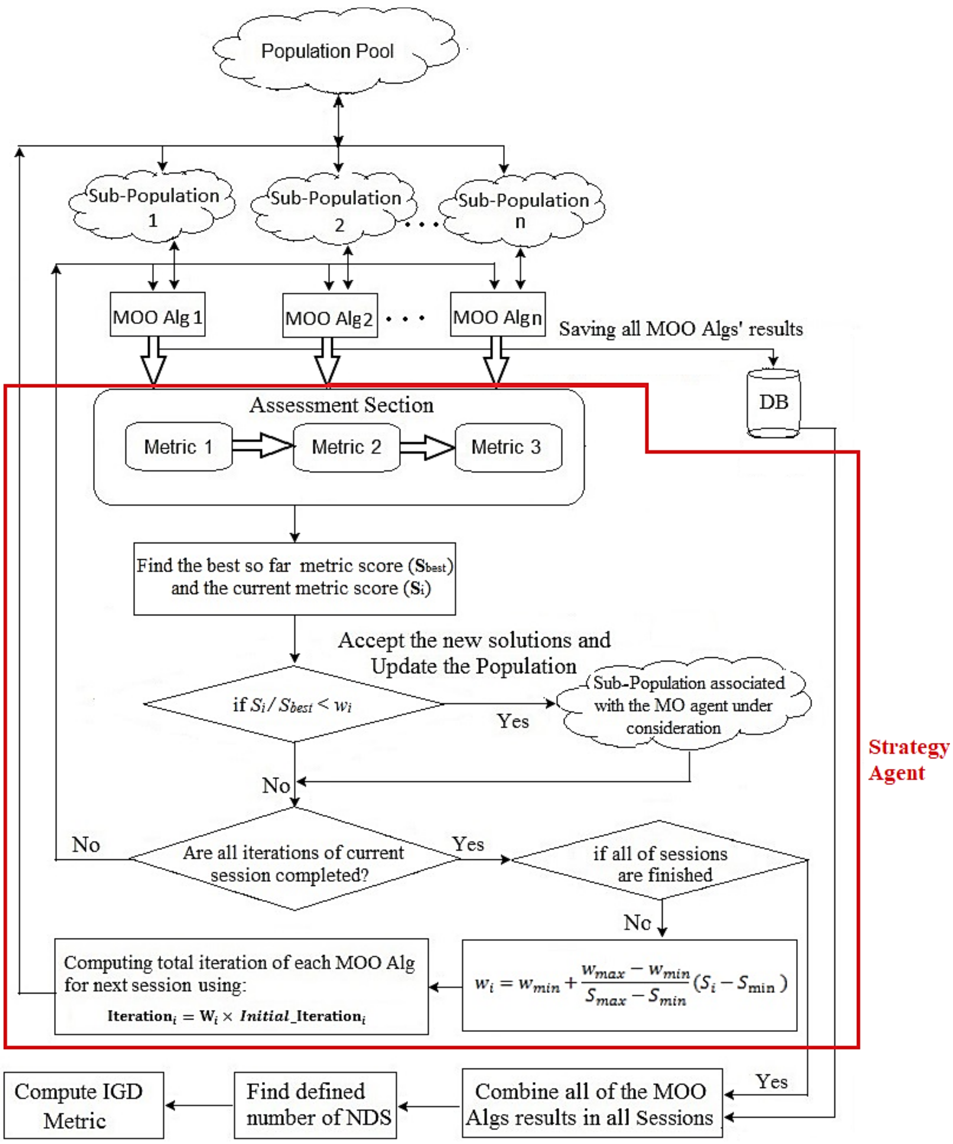


Figure 2.1: Flowchart of the process in the proposed MAS

The following paragraph shows the organization of the rest of this chapter: Section 2.2 expresses the basic components and main subjects of the proposed MAS. General description of Multi-Agent systems is given in Section 2.3. Meanwhile, the state-of-the-art in MAS's is presented in Section 2.4 which explains the related works on using Multi-agent systems for solving single and MOO problems. Section 2.5 illustrates the multi-objective assessment metrics applied in suggested MAS. Section 2.6 expresses the proposed heterogeneous, multi-deme multi-metric MAS for solving real-parameter MOOP. Finally, the description of benchmark problems, parameter values, evaluation results and comparisons are indicated in Section 2.7. The comparison is carried out in terms of quality and statistical analysis.

2.2 Multi-objective Optimization Methods Used in Proposed System

2.2.1 Archived Multi-Objective Simulated Annealing (AMOSA)

The simulated annealing method was extended and improved by Bandyopadhyay et al. (2008) to cover the multi-objective optimization task. The obtained algorithm was named as Archived Multi-Objective Simulated Annealing (AMOSA). The probability of accepting a solution in AMOSA is calculated based on dominance-rank. To apply this idea, the dominance ranks for current and new solutions against the solutions in archive are computed. Two kinds of constraints namely hard limit (HL) and soft limit (SL) are applied on the archive in which HL determines the MAX size for archive and SL indicates the amount of extra space for HL. By using a clustering method, the size of archive is decreased to HL at the end of algorithm. AMOSA applies the single linkage clustering algorithm for this objective. The evaluation results of AMOSA exhibited that AMOSA outperforms NSGA-II and MOSA in majority of benchmark problems [12].

2.2.2 Multi-Objective Differential Evolution (MODE)

Xue et al. (2003) extended and improved the DE method to work as an algorithm for multi-objective optimization. This method was called as Multi-Objective Differential Evolution (MODE). Similar to majority of optimization methods, MODE uses the crossover and mutation operators to construct and discover new solutions.

In MODE algorithm the selection strategy is carried out based on domination in which if a new solution appears in PF it is chosen as best, if not another solution on PF is chosen as best by coincidence.

To form an appropriate new population, the (μ+λ)-selection strategy alongside crowding distance is applied. It can be noticed from evaluation results that MODE algorithm outperforms SPEA method in terms of efficiency and solution quality [11].

2.2.3 Multi-Objective Genetic Algorithm (MOGA)

Fonseca et al (1993) introduced a population-based multi-objective genetic algorithm (MOGA) which follows all fundamentals and aspects of GA. MOGA uses dominance-rank value to measure the quality of solutions. Dominance rank value for a solution x is the count of all solutions dominated by solution x. In MOGA algorithm, the non-dominated solutions are assigned as highest quality solutions but dominated ones are assigned a penalty according to their degree of dominancy. Since MOGA is robust and powerful in finding optimal solutions, it is great of importance in many engineering applications.

2.2.4 Multi-Objective Particle Swarm Optimization (MOPSO)

Moore et al (1999) suggested a population based MOO method called MOPSO for multi-objective optimization.

MOPSO algorithm begins with generating the population of solutions (particles) and keeping the non-dominated individuals in an archive. The global best particles of the population are all particles in the archive. Thereafter the algorithms generate the next generation by changing the position of particles based on the position of their global best particles. Then the personal archives of the particles are updated if the new particle dominates its best position found so far. This way, the best positions of all particles are kept inside the personal archives. As soon as all particles are updated the global archive is also updated to keep all non-dominated solutions extracted up until now (Reyes-Sierra et al, 2006) [13, 14].

2.2.5 Strength Pareto Evolutionary Algorithm (SPEA2)

Zitzler et al. (2001) introduced a MOO method named by SPEA2 which uses an archive to keep all non-dominated solutions found so far. Instead of using dominance rank value, SPEA2 calculates the strength value to be used within the algorithm. Strength value of solution x is the number of solutions dominated by solution x. To increase the performance, they applied both dominance rank and strength values to specify the fitness of solutions located in archive. Moreover, the NNDS (Nearest Neighbor Density Estimation) was used to increase the extraction performance. SPEA2 merges the archive and current population in each generation to keep the archive updated. Later on, the parents are selected using tournament selection method to apply the crossover and mutation operators [10].

2.2.6 Non-dominated Sorting Genetic Algorithm (NSGAII)

Deb et al. (2002) suggested a novel multi-objective evolutionary algorithm called by NSGAII. The proposed method applies crowding and elitism operators in order to have a well spread Pareto-front and storing the good-quality individuals.

NSGAII generates the first population randomly and calculates the dominance-rank value for each individual in the population. Dominance-rank of a solution x is the number of individuals dominating solution x and it is taken into account when the fitness value is calculated. Thereafter, NSGAII applies the GA operators for creating the new population. The operators are Selection, Crossover and Mutation operators. In the next step, the current and new populations are merged to obtain an updated population. When the size of obtained population exceeds the maximum allowed size, the additional elements are removed based on the crowding distance metric. NSGAII method iterates these steps and ends up when the termination criterion is satisfied.

Since NSGAII is simple, robust and capable of finding solutions close to PF, it is well-known as an efficient algorithm for multi-objective optimization [7, 8].

2.2.7 Multi-Objective Artificial Bee Colony (MOABC)

Artificial Bee Colony (ABC) optimization algorithm was found and invented by Karaboga and Basturk (2005) that is inspired by honeybee's behavior in the nature. Thereafter, the researchers extended and improved ABC to solve the problems with more than one objective. The obtained method was called as MOABC. Since MOABC method uses archive to keep the non-dominated solutions found so far, all high-quality solutions are protected during the algorithm execution. Employed, onlookers and scout are three types of bees in MOABC.

Employed, onlookers and scout are three different types of bees in MOABC. In the proposed algorithm the employed bees are considered as solutions for solving MOO problems. The fitness of solutions is measured based on the nectar value of the food. MOABC begins by creating the first population of solutions and then all other steps are iterated in a loop until the termination criteria are satisfied. During the iteration all employed bees manipulate the position of foods to extract the new positions. The new position is accepted if it has more nectar rather than the current position. Meanwhile all new positions are delivered to the onlooker bees at the end of iterations. Thereafter, these bees specify the nectar volume of new foods in order to choose the best food among the all foods. The public archive stores all best foods extracted so far to avoid losing the high-quality solutions [15].

2.3 Multi-Agent Systems

A multi-agent system (MAS) consists of agents acting specific tasks on their environments [19]. In this system, agents monitor the environment and get the perceptions, then decide to act on environment by choosing actions based on their experiences. Consequently, due to those actions, the environment is changed to reach the system goals [19]. Figure 2.2 presents the MAS structure.

In intelligent type of MAS, the agents using the learning capability choose the actions intelligently to make the environment better than before. This way, the environment is improved, and the agents achieve predefined goals. The detailed description of multi-agent systems is given in [16].

In MAS, an agent includes hardware and software which are the components of its architecture [20]. The hardware part comprises sensors and actuators for watching and acting on environment. As well as the software part consists of procedures to process the perceptions, choose appropriate actions to be done on environment and update the knowledge of agent. The different types of agents are as following: reflexive, utility-based, goal-based and maintaining state agents. More information about these agent types are given in [20]. Agents of the new proposed MAS could be considered like utility-based agents that aim to diminish the objective functions in which the benefits of specific activities are evaluated according to the fitness of the objective functions under discussion.

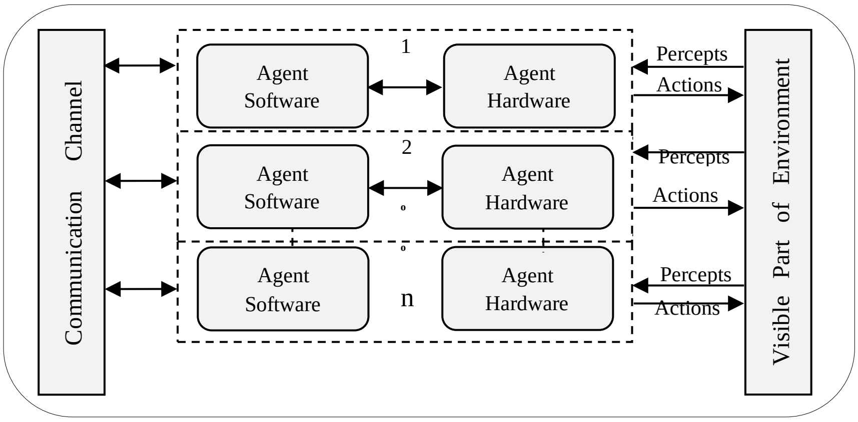


Figure 2.2: Generic illustration of a multi-agent system

As it is shown in Figure 2.2, all the agents in a multi-agent system are communicating together using a channel which is created as star model (in which communicating between agents are done through an agent called master agent) or as distributed inter agent dialogs (that agents are communicating in pair and exchanging messages by means of some predefined protocols) [21]. It should be noted that the second approach is a generic, multi-purpose and adaptable method, despite that it needs special languages and dedicated protocols to be used to communicate and pass messages between agents respectively. Star model is reasonably easy to implement for multi-agent systems, with quite small size and small number of agents, because all agents are supposed to operate based on a communication protocol [21].

Environment is another foundational component of a multi-agent system which is sensed and altered by its agents (as a shared source) to achieve their goals [22]. They can be classified according to accessibility and spatial properties. An overall explanation about environments can be found in [22].

Based on interaction characteristics of agents are discussed in [22], multi-agent systems are categorized into two groups of centralized and decentralized. In the centralized one, there is an agent so-called master agent that panels other agents' actions by getting local strategies of each agent, resolving potential conflicts and passing the related task schedules to achieve the goals. In contrast, a decentralized multi-agent system doesn't have any agent that acts as a central one, so each agent forms its own plans by interacting to other agents in the neighborhood or in the global.

The multi-agent system suggested in this research applies the centralized architecture which detailed description of this system can be found in Section 6.

Multi-agent systems containing metaheuristic algorithms as individual agents are extensively utilized to deliver cooperative and competitive outlines when doing optimization [21, 23]. In this regard, it has been presented over a variety of implementations that multi-agent systems based on metaheuristic algorithms carry productive approaches for resolving difficult MOO problems. More details regarding metaheuristic based MAS can be found in [23].

2.4 Multi-agent Systems for Optimization Problem

2.4.1 Multi-Agent Systems for Single Objective Optimization

The integrated process planning and scheduling problem (IPPS) was solved by Zhang et al using ACO and GA methods embedded in a suggested MAS system. The proposed multi-agent system consists of four agents collaborating together through the system. The first agent is EMA (Environment-Maintenance Agent) agent to create the improvement solution pool, constructive solution pool and global context. The agents use the global context, constructive solution pool and improvement solution pool to run the operations, keep the individuals extracted in each epoch and modify the individuals respectively. IPPS is of great importance in engineering applications.

The algorithm agent develops and enhances the ACO and GA methods applied in the suggested system. The functions participated in the system are used to create the initial population of solutions and to select the method among ACO and GA.

The supervisor agent is the third agent of the proposed system which monitors the system and manages the tasks.

The fourth agent is the transporter agent which prevent unneeded tasks via sending the data between the operators and generators. This is the way how to control the tasks between them. A part of the system (ISA) controls if the necessary solutions exist for the operators. In case of lack of the solutions, ISA generates the required solutions using the generator. Thereafter, the metaheuristic methods are performed over the collected solutions. Once the metaheuristics fulfill the execution, all information regarding to the solutions are updated by the ISA. Afterwards the termination criteria are checked by the ISA to decide if the system should stop. All aforementioned steps are iterated if the criteria are not satisfied.

The obtained results demonstrated that the suggested multi-agent system is robust in increasing the performance of metaheuristics. This way, the metaheuristics become capable of extracting feasible and good solutions for integrated process planning and scheduling problem [26].

Lotfi et al. presented a competitive and cooperative multi-agent system which works based on tournament among its agents to resolve single-objective optimization problems. As demonstrated in Figure 2.3, in this multi-agent system numerous metaheuristic algorithms are employed as problem-solving agents in a way that all the agents share and use the same pool/population. In this system agents work together, and with other architectural agents of the system and communication mechanisms agent. It performs in sequential sessions in which each session holds two steps: Firstly, the sub-populations are created by dividing the main population and a tournament is set to take place between the metaheuristic agents to find an agent with the highest performance and then conducts a search by the winner of tournament for entire population until fulfillment of the defined termination criteria.

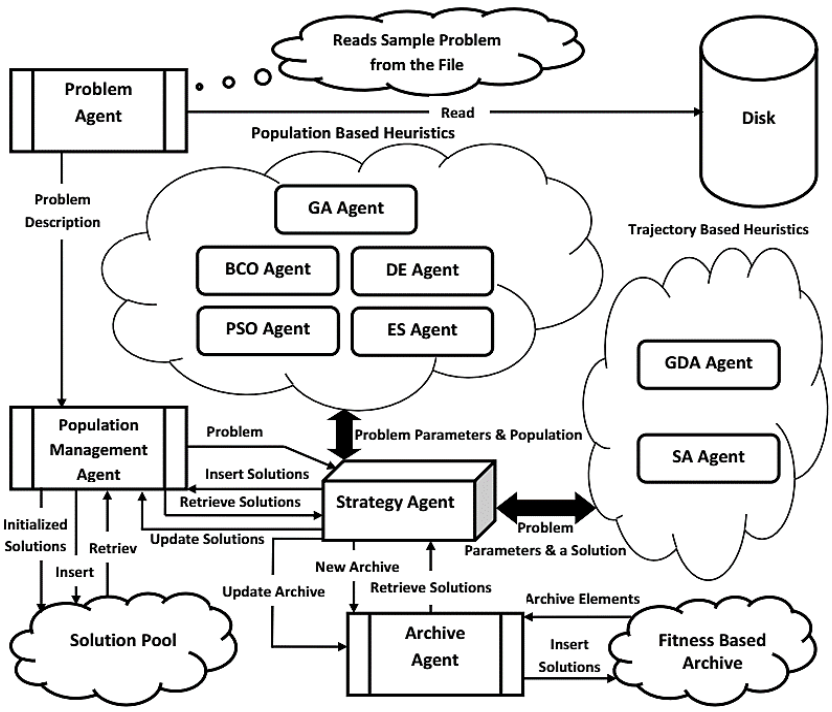


Figure 2.3: The proposed CMH-MAS system by Lotfi et al

The aforementioned steps are iterated until the stopping criteria are satisfied. In such a way, agents racing with each other according to their session wise victory and at the same time they cooperate together through sharing the results by means of saving the found non-dominated solutions in the same place. The achieved experiment results on CEC2005 test instances presented that the suggested method carries out considerably better results than existent methods in most of the test cases [27].

A multi-agent framework was proposed by Malek to provide an environment for metaheuristics cooperation. The proposed framework is illustrated in Figure 2.4 and encompasses four different agents namely: problem agent, solution pool agent, algorithm agent and advisor agent. The problem agent takes all parameters into account and generates a random feasible solution. Thereafter, the solution pool agent uses the first solution to produce the first population of solutions. The advisor agent sets the parameters up to be used by the methods. Likewise, the algorithm agent applies the solutions and parameters prepared by the solution and advisor agents respectively to run GA and TA metaheuristics.

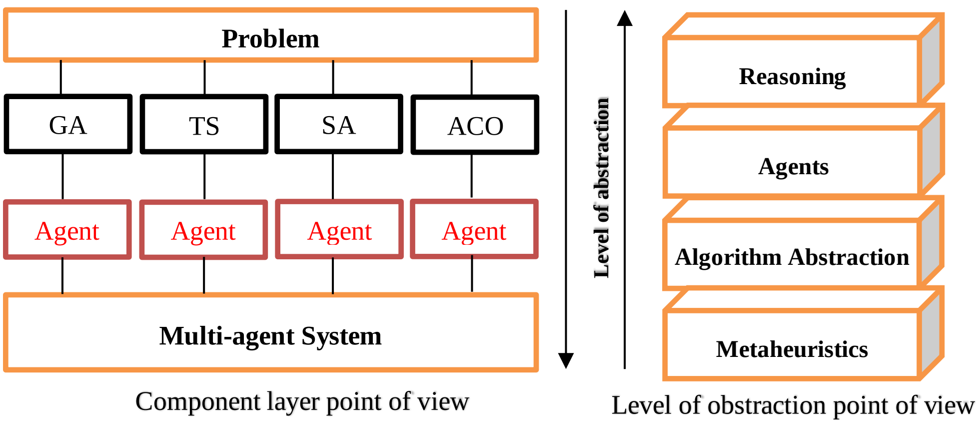


Figure 2.4: Metaheuristics cooperation (proposed by Malek)

Eventually, the evaluation of the suggested system is carried out in two steps: Evaluating the multi-agent system over TSP problem and evaluating of the GA and TA metaheuristics over the same TSP problem. The obtained results indicated that the suggested multi-agent system outperforms the individual GA and TA metaheuristics in majority of benchmark problems [25].

2.4.2 Multi-agent Systems for Multi-objective Optimization

A multi-agent framework was proposed by Jiang et al. to deal with multi-objective optimization problems. Authors applied the general and enhanced versions of MOEA/D, SPEA2 and NSGAII methods and compared their efficiency in the suggested system. To choose the next method in an efficient way, the introduced system uses the 'term' value which is the degree of appropriateness. Meanwhile, it applies the term value to adjust the parameters for crossover, mutation and selection operations. In this method the parameters and solutions behave like services and intelligent agents respectively. The intelligent agents in the system choose the services to take place in the optimization process in which the bigger trust value corresponds to higher selection chance.

When the quality of generated solutions is higher, the solutions stay alive in the population and the trust values increase by a rate. Similarly, the trust value is reduced when the quality of generated solutions is low. These weak solutions do not survive in the next population.

The authors of the paper evaluted their suggested system over 35 different benchmark problems reported in CEC2009 contest. The obtained results illustrated that the proposed system improved the effectiveness of existent multi-objective evolutionary algorithms [28].

Another multi-agent system called (RdMD/MAS) was proposed by Acan et al. to deal with MOO problems. The suggested system comprises the following six methods: AMOSA, NSGAII, MODE, SPEA2 and MOPSO. In this architecture uses dominance rank of solutions to split the population into sub-populations and then they are submitted to the aforementioned six MOO methods. Likewise, one global archive plus six local archives (one for each method) are applied in the proposed system to hold the NDS solutions found so far.

They selected CEC2009 problem set to evaluate the introduced system and the obtained results indicated that RdMD/MAS method outperforms the existent methods in majority of problems [29].

Uhruski et al. invented a two-layered multi-agent system (computing and management layers) called Agent-Oriented Governing System (AOGS) to deal with a well-known problem in distributed systems. The suggested MAS contains diverse agents which supply the resources and services to the agents in such a way that: computing layer is responsible for providing resources services, while management layer take care of responsibilities like scheduling and reserving of resources and individual computational duties (It is done in a place called execution space where it is found by the agents of management layer in the environment). The suggested MAS employs local diffusion scheduling strategy and intercommunication of agents to accomplish the inner duty as quickly as possible. Real-world test outcomes approved the advantages of AOGS for diverse sorts of technological and scientific problems [30].

Mohammadzadeh et al. intrduced a new approach based on multiagent system and metaheuristic algorithms, called MAS, as a Metaheuristic-based (MAMH) method. The suggested system contains 10 basic and robust metaheuristic algorithms as agents. Each metaheuristic agent attempts to find the best solutions while competing and cooperating with others to achieve the common goals. The MAMH method assumes two connections for coordination as follows: in the first of which all agents are joined to a shared memory and in the second one the algorithms interchange the best solution found in each iteration. This method is designed in such a way that agents with stronger processes are assigned with larger populations of solutions. MAMH was tested on 32 complex and high-dimensional optimization test cases related to email spam detection and the result outperform its competitors [31].

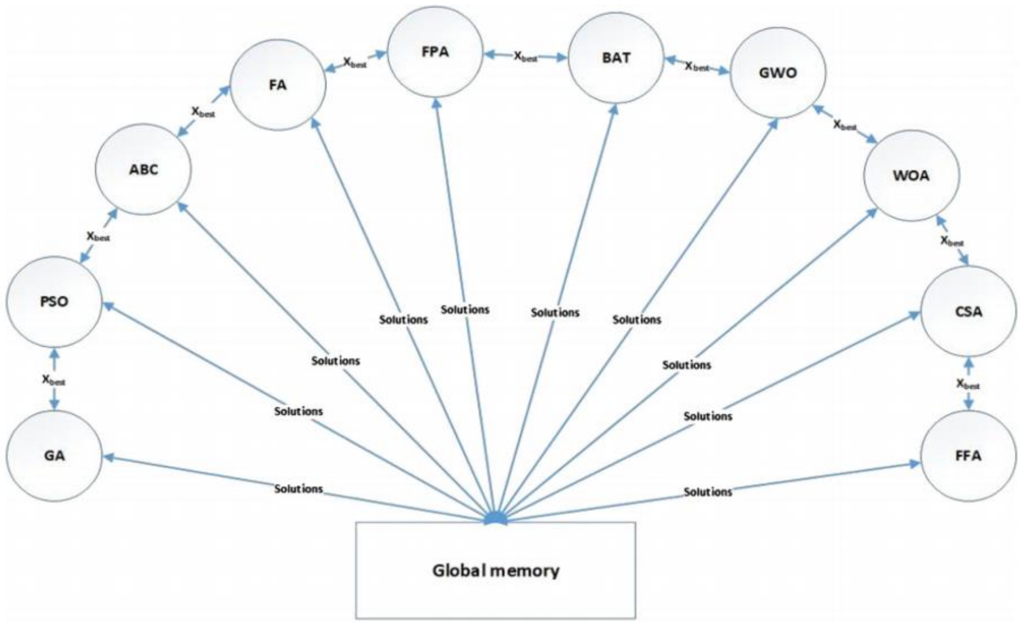


Figure 2.5: Overview of the proposed MAMH approach based on MAS

Shadravan et al. proposed a multiagent-based and distributed method for sailfish optimizer (DSFO) to speeds up the algorithm while keeping the high quality results. This version of SailFish Optimizer (SFO) uses Graphics Processing Units (GPUs) with CUDA architecture (Compute Unified Device Architecture) for solving the non-separable, non-convex, and scalable optimization problems. In this multi-agent system, a single agent in DSFO algorithm implements each group of operations in which it acts in parallel with other agents to either collect or return information from/to other agents and environments. The input information is a combination of the current position and best position of the search agents, and this information will be saved in the memory over the course of iteration during parallelization. In order to speed up the optimization process, their proposed approach includes decision-maker agents, exploration agents, and exploitation agents in which updating the sailfish's and sardine's positions is done by exploration and exploitation agents, respectively. Likewise the decision-maker agents calculate the cost of search agents and decide whether the current search agent in this iteration is the best or not. The obtained results show that the proposed method performs about 14 times faster than the other parallel algorithms and extracts the high quality solutions [32].

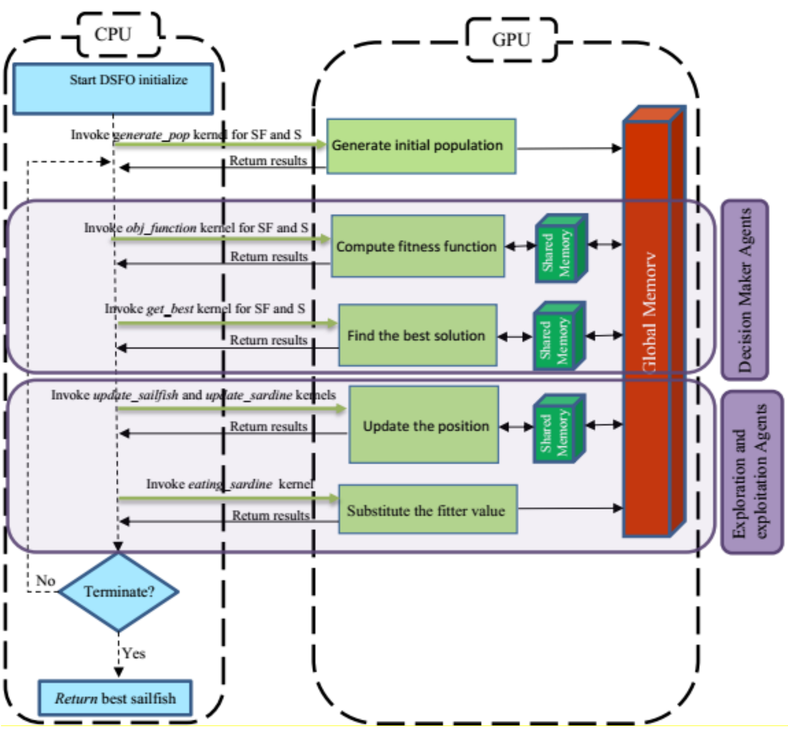


Figure 2.6: multiagent-based and distributed method for sailfish optimizer (DSFO)

To solve the truck dispatching problem in open-pit mines, Icarte et al. developed a multiagent system called MAS-TD. In this system, the intelligent agents representing the real-world equipment interact with each other to generate schedules (real-world equipment for mines such as ShovelAgents, TruckAgents, unloadingPointAgent, and so on) with the goal of maximizing the production at minimum cost. Moreover, the agents update the generated schedules when a major unforeseen event occurs at the mine. For example, making changes on external conditions can affect the schedule. The new proposed MAS was evaluated by comparing it against Tabu Search procedure applied on actual data from a Chilean open-pit mine. The results show that both MAS-TD and the Tabu Search procedure are suitable methods to solve the problem at hand. However, the schedules generated by MAS-TD are more efficient than the schedules generated by the Tabu Search [33].

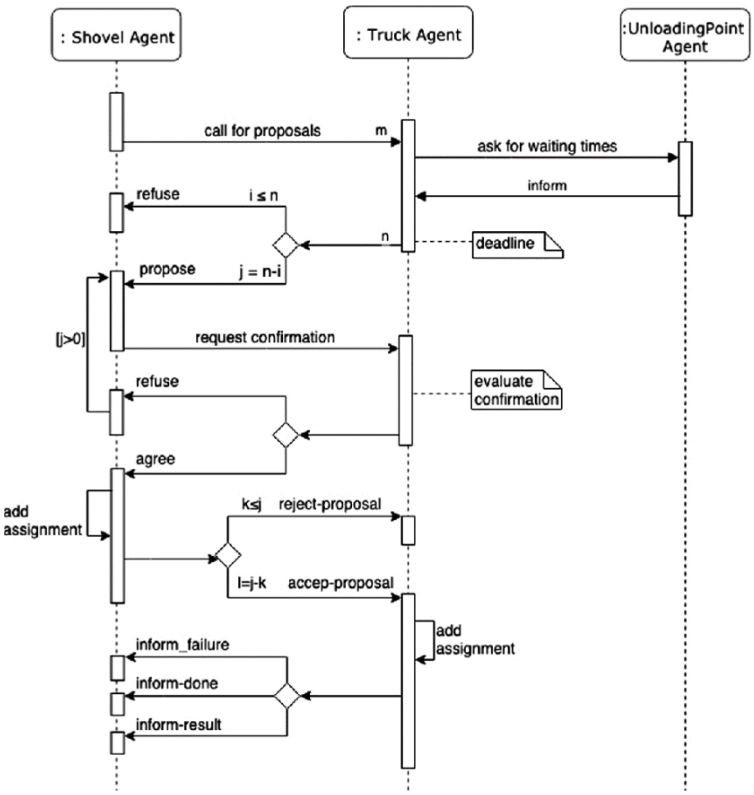


Figure 2.7: The interaction between the agents using the improved Contract Net protocol with the confirmation stage

The proposed multi-agent system in this research holds novel features in comparison with the above mentioned multi-objective multi-agent system frameworks. The suggested system consists of several metaheuristic-based multi-objective optimization agents performing on subsets of a shared population while they also preserve their local archives to store the NDSs found in each session. The suggested MAS system performs in successive sessions in a way that each session contains two phases in which: in phase one, the main population is divided into sub-populations such that an individual could possibly be opted for multiple sub-populations however, there is precisely one duplicate of any of the individuals in a sub-population.

In phase two, every agent functions on its sub-populations using its own multi-objective search strategies and tools. As a result, agents return the improved sub-population and the extracted Pareto front. Each discovered Pareto front is evaluated by the selected multi-objective metric for that phase in a way that the evaluation scores are employed in two ways; firstly, the limit of fitness evaluation for every agent is adjusted according to its performance. Secondly, a sub-population enhanced/upgraded via an agent could possibly be disallowed based on the measured metrics. The NDS solutions extracted during each session are kept by local archives and at the end of session all NDS solutions found so far are saved in a global archive. Therefore, all extracted information is shared between the methods. The proposed multi-agent system contains one managerial agent which monitors agents' communications and activities and manages shared population and archive. The proposed architecture is applied on real-valued multi-objective optimization problems in CEC2009 test instances. Assessment of the achieved results exhibited that the suggested system is truly a strong substitute for the tough numerical MOPs.

2.5 Multi-objective Assessment Metrics of the Proposed MAS

During the last 10 years, a number of multi-objective metrics are being offered to measure the comparative assessment of multi-objective optimization methods. In this study three distinguished multi-objective assessment metrics are taken into account, as follows: inverted generational distance (IGD), generalized spread and ε–indicator. These metrics are employed to estimate achievement of MOO methods in relation to Euclidean mean distance, diversity and additive distance [34]. These metrics are described briefly as follow:

i- ε–indicator: It shows the degree of additive proximity between two Pareto fronts (which means one acts as a reference Pareto front and the other as a computed Pareto front). That is, $I_{\varepsilon+}$ (A, B) describes the minimum value by which all elements of Pareto front 'A' should be transferred to weakly dominate Pareto front 'B'. Mathematical description of this metric is:

$$I_{\varepsilon+}(A,B) = \inf_{\varepsilon \in R}\{\forall \vec{b} \in B \mid \exists \vec{a} \in A : \vec{b} \preccurlyeq \vec{a} + \epsilon\} \quad (2)$$

where, 'B' is the reference Pareto front and 'A' is the estimated one [34, 36].

ii- Generalized Spread metric (Δ*): It is an improved version of the spread metric (Δ) which is used to measure distribution and diversity of the individuals of a estimated Pareto fronts (NDS) 'A' beside a reference Pareto fronts 'B' [31, 33]. Its formula is shown in Equation 3,

$$\Delta^*(A, B) = \frac{\sum_{k=1}^{m} d(\vec{e}_k, A) + \sum_{i=1}^{|A|} |d_i - \bar{d}\,|}{\sum_{k=1}^{m} d(\vec{e}_k, A) + (|A|)\bar{d}} \quad (3)$$

where, 'A' is the reference Pareto front and 'B' is the estimated one (NDS), $\vec{e}_k \in B$ displays the extreme solutions on the k_{th} objective axis, $d(\vec{e}_k, A) = \min_{\vec{a} \in A} ||F(\vec{e}_k) - F(\vec{a}\,)||$ is the minimum Euclidean distance between an extreme solution $\vec{e}_k \in B$ and the Pareto front A, $d_i = \min_{\vec{a}_j \in A, \vec{a}_j \neq \vec{a}_i} ||F(\vec{a}_i) - F(\vec{a}_j)||$ is the Euclidean distance between two closest solutions of 'A', 'm' stands for the number of objectives and $\bar{d}$ refer to the average of d_i.

iii- IGD metric: It is basically used as a convergence metric that calculates the mean value of minimum Euclidean distances between individuals of reference Pareto front 'A' and an estimated one 'B' [34, 37]. Its formula is presented in Equation 4,

$$\text{IGD (B, A)} = \frac{\left(\sum_{i=1}^{|B|} d_i^2\right)^{1/2}}{|B|} \tag{4}$$

where $d_i = \min_{\vec{a} \in A} ||F(\vec{b}_i) - F(\vec{a})||$ stands for the minimum Euclidean distance between $b_i \in B$ and the reference set 'A'.

Convergence to optimal/reference Pareto front and diversity of found solutions are the two major aims of all multi-objective optimization algorithms. Therefore, in this research, the above mentioned three assessment metrics are selected to evaluate the achievement of the agents and lead them during their search processes to reach better convergence and well-spread Pareto-front within the suggested MAS.

2.6 The Proposed Multi-Metric and Multi-Deme Multi-Agent Architecture

This section explains the proposed multi-agent multi-metric framework to deal with real-valued MOO problems. The key concept introduced in this research is a multi-agent framework in which several metaheuristic algorithms for multi-objective optimization are employed as individual agents that are supervised and coordinated by strategy and population management agents. Individual agents are allocated to sub-populations and their achievement is assessed in a session-wise manner by multiple multi-objective assessment metrics. The proposed system contains five metaheuristic based multi-objective optimization agents and three multi-objective assessment metrics. Therefore, full execution of the suggested multi-agent system is completed in three sessions in which the performance of individual agents is assessed using of the three assessment metrics. Each assessment metric is used only one time and experimental outcomes exhibited that the order of their usage has no considerable influence on the overall performance of the suggested MAS. Hereby, each one of the metaheuristic agents attempt to improve Pareto fronts according to diverse assessment metrics in each session. As described above, ε–indicator and generalized spread metrics are employed to measure the degree of convergence and diversity along a reference Pareto front, respectively, while inverted generalized distance is used to measure the degree of both convergence and diversity at the same time. As a result, the proposed multi-agent system targets to extract Pareto fronts with high quality of convergence and diversity. In each session, the suggested MAS system allocates different slices of the main population to metaheuristic agents and evaluates their success based on the same selected metric for that session. If an agent does not obtain sufficient improvement in a session, then the output sub-population is rejected even though its extracted NDSs are passed to archive agent. In this case, all agents in the proposed system cooperate to process and modify the global population due to avoid early convergence and local optimal points. This is better than working on same and fixed populations. Another benefit of the system is that agents can share all results and outcomes with each other, e.g. enhanced subpopulations, non-dominated set and so on. The agents share the improved subpopulations and non-dominated sets at the end of phases using a global population and common global archive, not via direct communications. Therefore, the improved solutions from all agents are collected to provide a new global population to be used by agents in the next phase.

Figure 2.8 illustrates the architectural view of the steps in the suggested system. In this architecture, there are two sorts of agents: one sort is those which are related to organization, second type is those which are metaheuristics. The problem agent in the system reads the problem definitions and initializes the related parameters like objective functions and so on. Thereafter, the solution pool agent receives all problem information from problem agent to handle all actions over the global population. Solution pool agent initializes the first global population by random and calculates their objective function values. Also, this agent divides the global population into subpopulations and distributes them between metaheuristic agents.

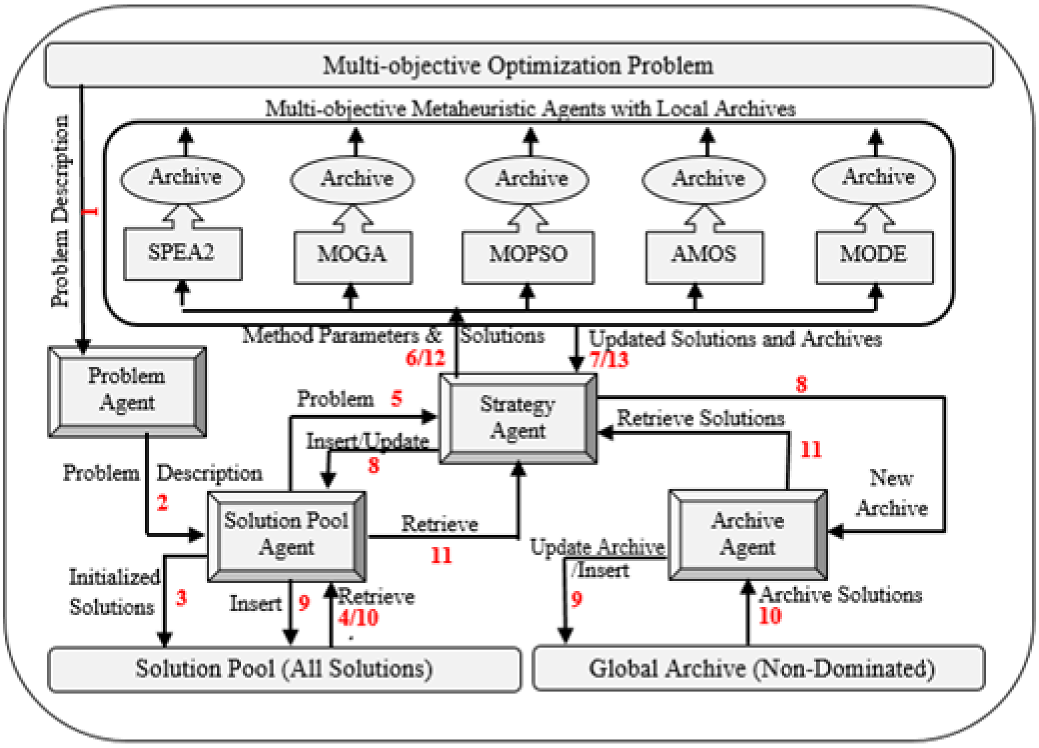


Figure 2.8: Architectural illustration of the suggested multi-agent system

It should be noticed that a solution can be placed inside more than one subpopulation. As well as it is possible for a solution not to be inside any of subpopulations. The solution pool agent sends the subpopulations and all related information like objective functions and metric values upon the strategy agent request.

The archive agent in Figure 2.8 cooperates with strategy agent to retrieve solutions and new extracted archive. It is also in a relation with global archive to update the contents. Whenever a MOO agent submits a new non-dominated set to strategy agent, the archive agent will receive it and combine the new set with its own contents. To avoid having dominated solutions in global archive after combination, the dominated ones are eliminated from the archive. Afterwards, the new global archive is transferred to strategy agent to be applied by all agents. The most important agent in the system is strategy agent. This agent is in relation with all other agents in the system to handle almost all activities like transferring and receiving data.

In this thesis, the suggested MAS executes in three successive phases. Also several metrics are used to measure the agent’s success rate. The first phase assigns the subpopulations and fitness evaluation size to the agents. The fitness evaluation sizes for the agents are calculated in proportion to the success rate of agents in second and third phases. The main agent of the system, strategy agent, calculates the success rate of agents based on their obtained results and metric values at the end of sessions. When an agent has a bigger success rate, it will get a higher iteration number in the next phase. In this context, w_i is calculated as the agent score (which indicate the fitness evaluations count). It should be noticed that the number of fitness evaluations are assumed as α for all agents and it will be changed during the next phases between $\beta_1 * \alpha$ and $\beta_2 * \alpha$.

In order to calculate the value of w_i, all w_i values are set between w_{min} and w_{max}, in which, the w_{max} and w_{min} are assigned to the best and worst agents. Meanwhile, the w_i value for other agents are calculated as shown in Equation 5:

$$w_i = w_{min} + \frac{w_{max} - w_{min}}{S_{max} - S_{min}} (S_i - S_{min}) \quad (5)$$

In this formula, S_i indicates the metric score for agent i. Also, the largest and smallest metric scores at the end of the phase are shown by S_{max} and S_{min}. The system also calculates S_i/S_{best} value and uses it to decide on acceptance or rejection of the subpopulations. If the S_i/S_{best} value is less than or equal to w_i the obtained subpopulation is rejected. It should be considered that the extracted non-dominates individuals are used to update the global archive. The strategy agent plays a significant role in the suggested system. Figure 2.9 represents the description of the system by flowchart.

The global archive holds the best Pareto-front extracted up to now in each phase. It is also used to compute the metrics values for agents. Table 2.1 represents the parameter values.

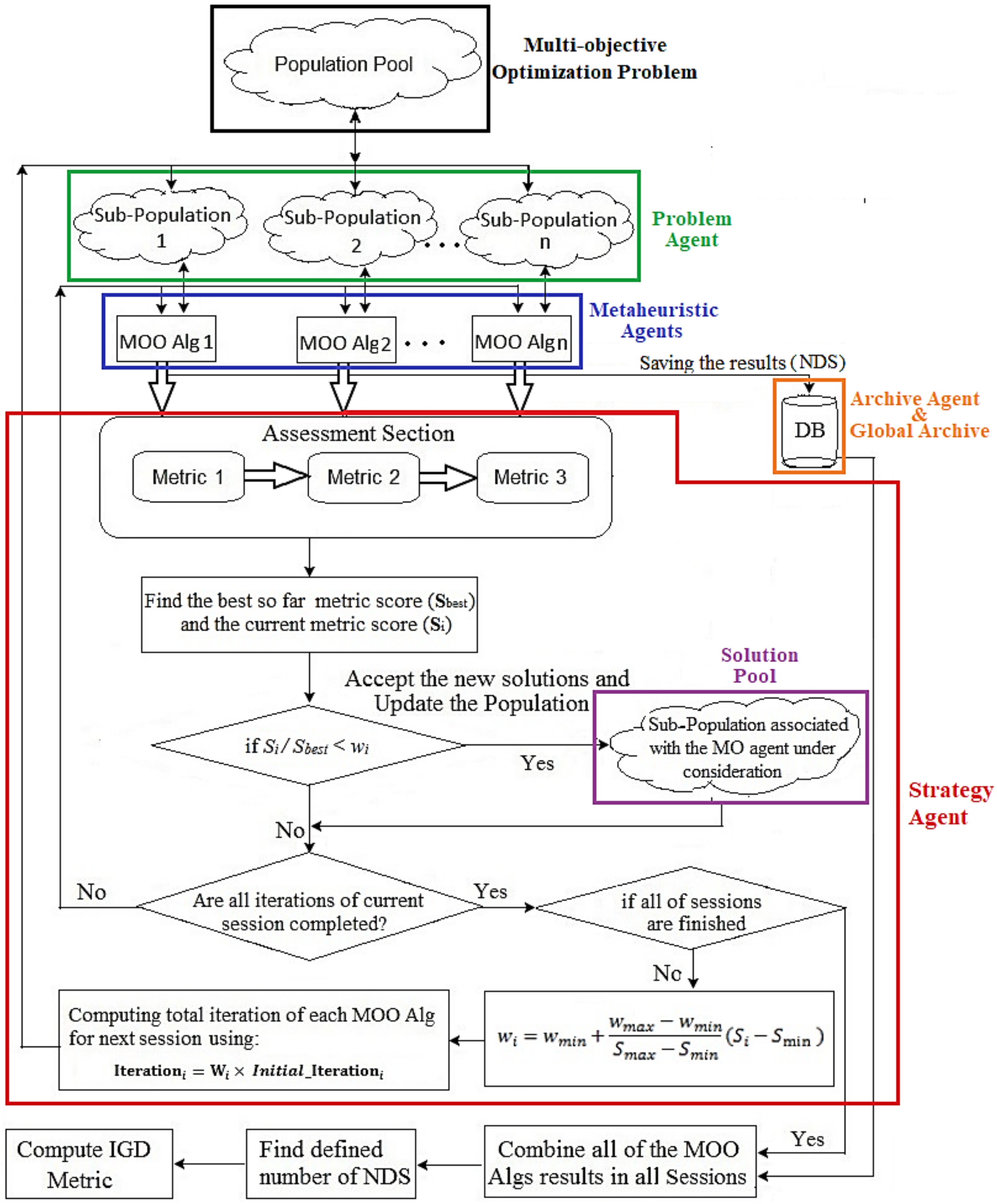


Figure 2.9: Flowchart of process and association among the agents of proposed multi-agent system

2.7 Experimental Results and Evaluations

2.7.1 Experimental Results and Evaluations Over the Test Cases of CEC2009 Benchmarks

Performance assessment of the suggested MAS and together with its comparative success against state-of-the-art metaheuristic methods is done first over the challenging test instances in CEC2009 benchmarks [18]. Definitions of these test instances are explained in detail by Zhang et al (2009). The number of independent runs and fitness evaluation count for each test instances are adjusted as it is described in the literature. Likewise, all the necessary parameters are adjusted similar to the state-of-the-art methods to make the comparisons fair and correct.

Parameters of all the metaheuristics used within the architecture of the suggested multi-agent system are specified in Table 2.1 where the suggested Multi-Metric Multi-Deme MAS is called M^3D/MAS. In the evaluation process, the values of all parameters are set according to the standard versions of the methods reported in the literature. Moreover, for all metaheuristics except AMOSA (since it is not population-based metaheuristic) the size of population (α) is considered as 20.

The proposed system is implemented in Matlab® and executed over a PC with 32 GB RAM and Dual Core 2.7 GHz CPU.

Table 2.1: Initial values of metaheuristics' parameters in M3D/MAS

Metaheuristic Agent	Algorithm Parameters
M^3D/MAS	α=20, w_{min}= 0.6 w_{max}=1.4 β_1=0.1 β_2=1.9
MOGA	P_C =0.7, P_m=0.2, Gaussian_Sigma_Pm=20
MOPSO	C1=2.0, C2=2.0, ω_{max}=0.9, ω_{min}=0.4
MODE	Scaling_Factor=0.5, P_C =0.7
SPEA2	P_C =0.9, P_m=1.0/Num_Vars, Distribution_Index=20
AMOSA	Archive_H_{limit}=20, Archive_S_{limit}=50, Hill_Climbing_Num=20, Max_Temp=200, Min_Temp=0.00025, Gamma=2.0,

The suggested MAS, M^3D/MAS, is comparably assessed over CEC2009 benchmarks. There are 10 multi-objective unconstrained test cases exist in this set; which are generally created from the set of classical benchmarks by random shifting, random shifting and rotating and hybrid composition operations. From these test instances, UF1 to UF7 are problems with two objectives and UF8 to UF10 are the problems with three objectives. [18, 38] indicates the detailed explanation of the test cases and all experimental conditions. According to the rules of CEC2009, the variable numbers is 30 as a problem size. Also, the maximum number of fitness evaluations is considered as 300,000 in the M^{3}D/MAS. Also each result is the average of all results obtained in 30 different runs. Meanwhile, CEC2009 requires using IGD (Inverted Generational Distance) values for comparing the competitor's performances. Therefore, to do the comparison, the results of M^{3}D/MAS are compared to all results of outperforming competitors in CEC2009 provided in [38]. In order to compute IGD values, as explained in [38], it is required to have 100 and 150 solutions in final Pareto-front for two-objective and three-objective test instances respectively. The outperforming methods are marked as bold in all tables of this section. The lowest, highest and average values of IGD in 30 independent runs of M^{3}D/MAS for 10 problems are presented in Table 2.2.

Table 2.2: Lowest, highest and average values of IGDs in 30 runs for M3D/MAS

Function	Average (IGD)	Min (IGD)	Max (IGD)	Std (IGD)
UF1	0.00570	0.00528	0.00626	0.00030
UF2	0.00740	0.00679	0.00782	0.00334
UF3	0.04393	0.03507	0.05142	0.00423
UF4	0.03364	0.02541	0.03566	0.00203
UF5	0.08770	0.07619	0.09800	0.00722
UF6	0.05676	0.03877	0.05988	0.00521
UF7	0.00608	0.00579	0.00731	0.00366
UF8	0.11023	0.10320	0.11198	0.00987
UF9	0.07022	0.06839	0.07327	0.03650
UF10	0.30490	0.22408	0.32672	0.06663

Table 2.2 shows the efficiency and robustness of M^{3}D/MAS based on the small IGD mean scores and standard deviations it earns as a result of the evaluation. The largest IGDs belong to instances UF8 and UF10, however as shown later in Tables 2.6 and

2.7, for the problems UF8 and UF10, the success ranks of M^3D/MAS among 14 competitors are 6th and 2nd respectively.

Tables 2.3, 2.4, 2.5, 2.6 and 2.7 present the ranking of M^3D/MAS and all outperforming methods in CEC2009 based on IGD values. According to the results published in [38], the five best methods are MOEA/D [39], MTS [40], DMOEADD [41], LiuLi [42] and GDE3 [43] respectively. Therefore, the winner of the contest was MOEA/D. It can be seen over the four tables and also Figure 2.10 and 2.11 that M^3D/MAS performed better than MOEA/D in 4 of the 10 test cases. The suggested multi-agent system earns the second place for 4 test cases (namely: UF6, UF7, UF9 and UF10) and overall takes the second or third positions in 70% of the ten test cases.

As shown in the tables, M^3D/MAS takes the worst rank of six for UF8. Also, M^3D/MAS takes the 2nd position for solving the complicated problems UF9 and UF10. In these problems, the IGD value obtained by M^3D/MAS is much better than the majority of competitors.

Table 2.3: Average values of IGDs for UF1and UF2 achieved by M^3D/MAS and its competitors

Rank	UF1	Mean±Std (IGD)	UF2	Mean±Std (IGD)
1	**MOEA/D**	0.00435±0.0002	MTS	0.00615±0.0005
2	GDE3	0.00534±0.0003	MOEADGM	0.00640±0.0007
3	**M^3D/MAS**	**0.00570**±0.0003	DMOEADD	0.00679±0.0020
4	MOEADGM	0.00620±0.0010	**MOEA/D**	0.00679±0.0018
5	MTS	0.00646±0.0003	**M^3D/MAS**	**0.00740**±0.0033
6	LiuLiAlgorithm	0.00785±0.0020	OWMOSaDE	0.00810±0.0023
7	DMOEADD	0.01038±0.0023	GDE3	0.01195±0.0015
8	NSGAIILS	0.01153±0.0073	LiuLiAlgorithm	0.01230±0.0033
9	OWMOSaDE	0.01220±0.0012	NSGAIILS	0.01237±0.0091
10	ClusteringMOEA	0.02990±0.0022	AMGA	0.01623±0.0031
11	AMGA	0.03588±0.0102	MOEP	0.01890±0.0038
12	MOEP	0.05960±0.0128	ClusteringMOEA	0.02280±0.0078
13	DECMOSA-SQP	0.07702±0.0393	DECMOSA-SQP	0.02834±0.0313
14	OMOEAII	0.08564±0.0040	OMOEAII	0.03057±0.0016

Table 2.4: Average values of IGDs for UF3 and UF4 achieved by M3D/MAS and its competitors

Rank	UF3	Mean±Std (IGD)	UF4	Mean±Std (IGD)
1	**MOEA/D**	0.00742±0.0058	MTS	0.02356±0.0006
2	LiuLiAlgorithm	0.01497±0.0240	GDE3	0.02650±0.0003
3	DMOEADD	0.03337±0.0056	**M^{3}D/MAS**	**0.03364**±0.0020
4	**M^{3}D/MAS**	**0.04393**±0.0042	DECMOSA-SQP	0.03392±0.0053
5	MOEADGM	0.04900±0.0659	AMGA	0.04062±0.0017
6	MTS	0.05310±0.0117	DMOEADD	0.04268±0.0013
7	ClusteringMOEA	0.05490±0.0453	MOEP	0.04270±0.0834
8	AMGA	0.06998±0.0139	LiuLiAlgorithm	0.04350±0.0006
9	DECMOSA-SQP	0.09350±0.1979	OMOEAII	0.04624±0.0009
10	MOEP	0.09900±0.0132	MOEADGM	0.04760±0.0026
11	OWMOSaDE	0.10300±0.0190	OWMOSaDE	0.05130±0.0019
12	NSGAIILS	0.10603±0.0686	NSGAIILS	0.05840±0.0051
13	GDE3	0.10639±0.0129	ClusteringMOEA	0.05850±0.0072
14	OMOEAII	0.27141±0.0376	**MOEA/D**	0.06385±0.0053

Table 2.5: Average values of IGDs for UF5 and UF6 achieved by M3D/MAS and its competitors

Rank	UF5	Mean±Std (IGD)	UF6	Mean±Std (IGD)
1	MTS	0.01489±0.0032	**MOEA/D**	0.00587±0.0017
2	GDE3	0.03928±0.0039	**M^{3}D/MAS**	**0.05676**±0.0052
3	**M^{3}D/MAS**	**0.08770**±0.0072	MTS	0.05917±0.0106
4	AMGA	0.09405±0.0120	DMOEADD	0.06673±0.0238
5	LiuLiAlgorithm	0.16186±0.0282	OMOEAII	0.07338±0.0024
6	DECMOSA-SQP	0.16713±0.0895	ClusteringMOEA	0.08710±0.0076
7	OMOEAII	0.16920±0.0039	MOEP	0.10310±0.0345
8	**MOEA/D**	0.18071±0.0681	DECMOSA-SQP	0.12604±0.5617
9	MOEP	0.22450±0.0344	AMGA	0.12942±0.0565
10	ClusteringMOEA	0.24730±0.1307	LiuLiAlgorithm	0.17555±0.0829
11	DMOEADD	0.31454±0.0465	OWMOSaDE	0.19180±0.0290
12	OWMOSaDE	0.43030±0.0174	GDE3	0.25091±0.0195
13	NSGAIILS	0.56570±0.1827	NSGAIILS	0.31032±0.1913
14	MOEADGM	1.79190±0.3181	MOEADGM	0.55630±0.1959

Table 2.6: Average values of IGDs for UF7 and UF8 achieved by M3D/MAS and its competitors

Rank	UF7	Mean±Std (IGD)	UF8	Mean±Std (IGD)
1	**MOEA/D**	0.00444±0.0011	**MOEA/D**	0.05840±0.0032
2	**M^{3}D/MAS**	**0.00608**±0.0036	DMOEADD	0.06841±0.0094
3	LiuLiAlgorithm	0.00730±0.0008	LiuLiAlgorithm	0.08235±0.0073
4	MOEADGM	0.00760±0.0009	NSGAIILS	0.08630±0.0124
5	DMOEADD	0.01032±0.0022	OWMOSaDE	0.09450±0.0119
6	MOEP	0.01970±0.0750	**M^{3}D/MAS**	**0.11023**±0.0098
7	NSGAIILS	0.02132±0.0194	MTS	0.11251±0.0129
8	ClusteringMOEA	0.02230±0.0040	AMGA	0.17125±0.0172
9	DECMOSA-SQP	0.02416±0.0223	OMOEAII	0.19200±0.0122
10	GDE3	0.02522±0.0088	DECMOSA-SQP	0.21583±0.1214
11	OMOEAII	0.03354±0.0017	ClusteringMOEA	0.23830±0.0349
12	MTS	0.04079±0.0144	MOEADGM	0.24460±0.0440
13	AMGA	0.05707±0.0653	GDE3	0.24855±0.0355
14	OWMOSaDE	0.05850±0.0291	MOEP	0.42300±0.0565

Table 2.7: Average values of IGDs for UF9 and UF10 achieved by M3D/MAS and state-of-the-art methods.

Rank	UF9	Mean±Std (IGD)	UF10	Mean±Std (IGD)
1	DMOEADD	0.04896±0.0091	MTS	0.15306±0.0158
2	**M^{3}D/MAS**	**0.07022**±0.0365	**M^{3}D/MAS**	**0.30490**±0.0666
3	NSGAIILS	0.07190±0.0450	DMOEADD	0.32211±0.0222
4	**MOEA/D**	0.07896±0.0531	AMGA	0.32418±0.0957
5	GDE3	0.08248±0.0224	MOEP	0.36210±0.0444
6	LiuLiAlgorithm	0.09391±0.0470	DECMOSA-SQP	0.36985±0.6532
7	OWMOSaDE	0.09830±0.0244	ClusteringMOEA	0.41110±0.0804
8	MTS	0.11442±0.0254	GDE3	0.43326±0.0123
9	DECMOSA-SQP	0.14111±0.3453	LiuLiAlgorithm	0.44691±0.1296
10	MOEADGM	0.18780±0.0441	**MOEA/D**	0.47415±0.0736
11	AMGA	0.18861±0.0421	MOEADGM	0.56460±0.0696
12	OMOEAII	0.23179±0.0647	OMOEAII	0.62754±0.1459
13	ClusteringMOEA	0.29340±0.0850	OWMOSaDE	0.74300±0.0885
14	MOEP	0.34200±0.1584	NSGAIILS	0.84468±0.1626

Comparisons between M^{3}D/MAS and MTS, which is the second best performing method in the competition, display that the suggested method performs better than MTS in 6 out of the 10 cases. Figure 2.10 and 2.11 represent the average IGD as well as IGD rank for UF1 to UF10 benchmarks obtained by M^{3}D/MAS and 13 state-of-the-art methods. Considering the MOEAD as the winner of CEC20009, it can be seen that the proposed method either outperforms or performs same as the winner.

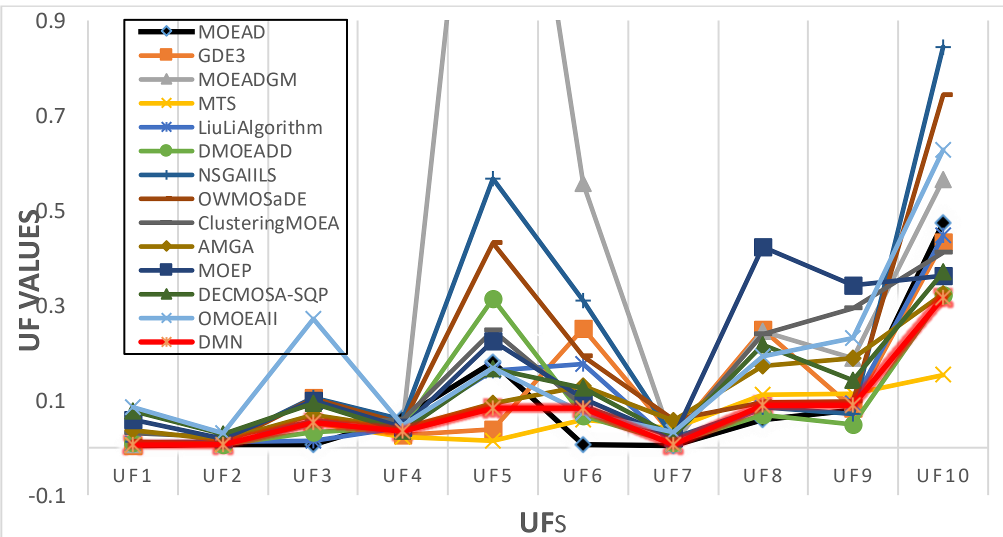


Figure 2.10: Average values of IGDs (for UF1 to UF10) found by M^{3}D/MAS and its competitors

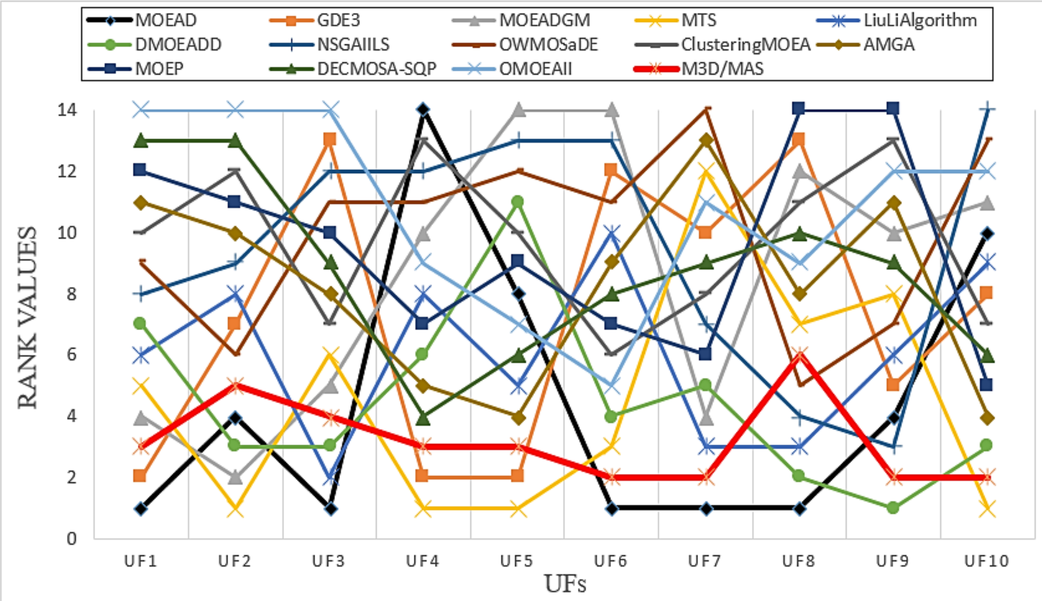
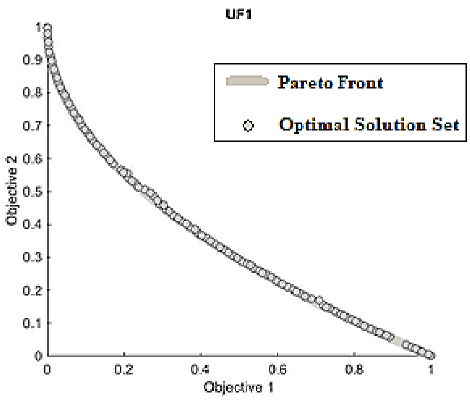
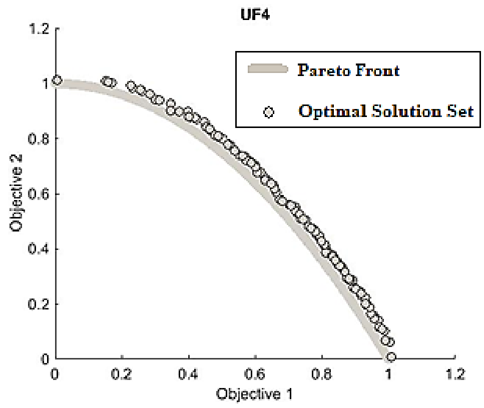


Figure 2.11: Rank values found (for UF1 to UF10) by M^{3}D/MAS and its competitors

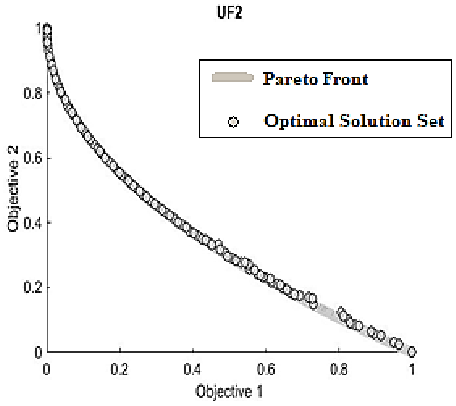
Figure 2.12 shows the plots for best Pareto-fronts extracted by M^{3}D/MAS and Pareto-front true introduced in CEC2009 competition. Based on the rules of this competition, the plots of two-objective test cases have 100 NDSs and the ones for three-objective test cases hold 150 NDSs (which selection of the individuals are carried out according to the information in [38, 44]).



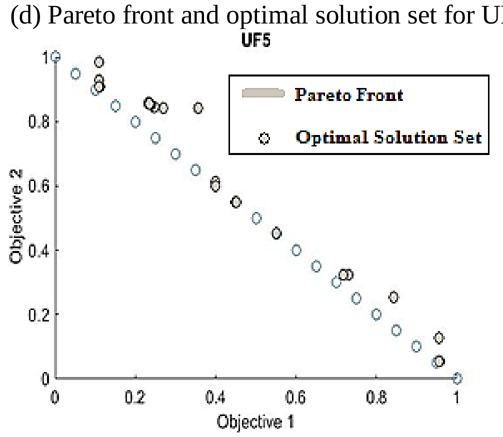
(a) Pareto front and optimal solution set for UF1



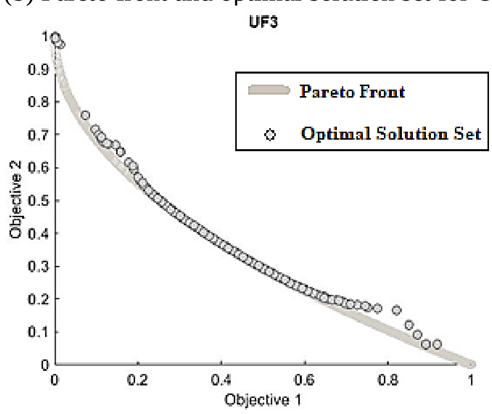
(d) Pareto front and optimal solution set for UF4



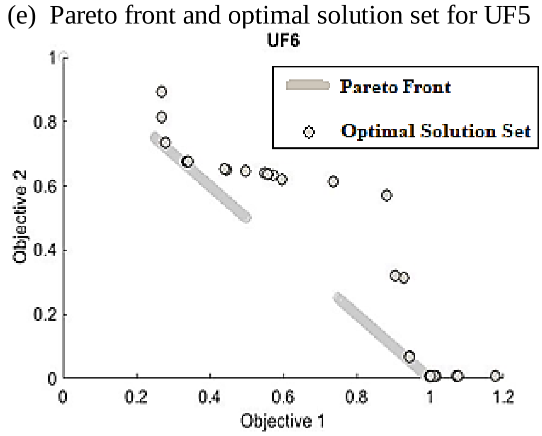
(b) Pareto front and optimal solution set for UF2



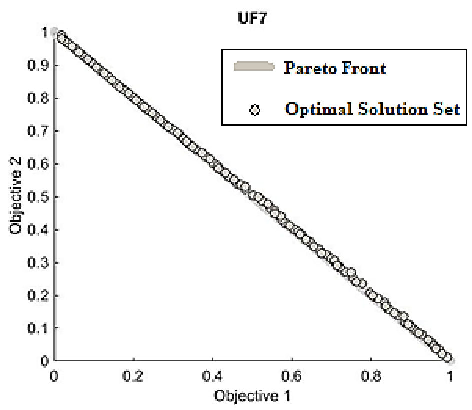
(e) Pareto front and optimal solution set for UF5



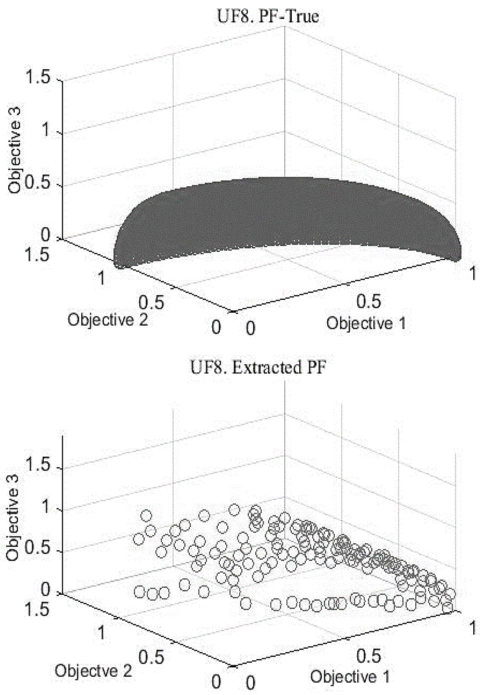
(c) Pareto front and optimal solution set for UF3



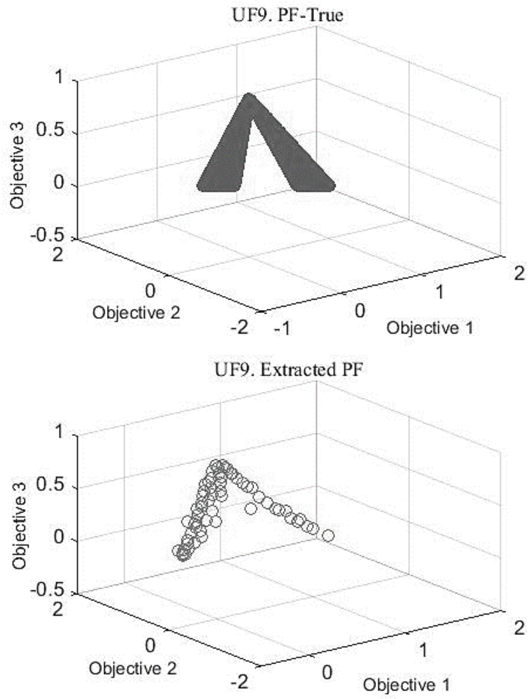
(f) Pareto front and optimal solution set for UF6



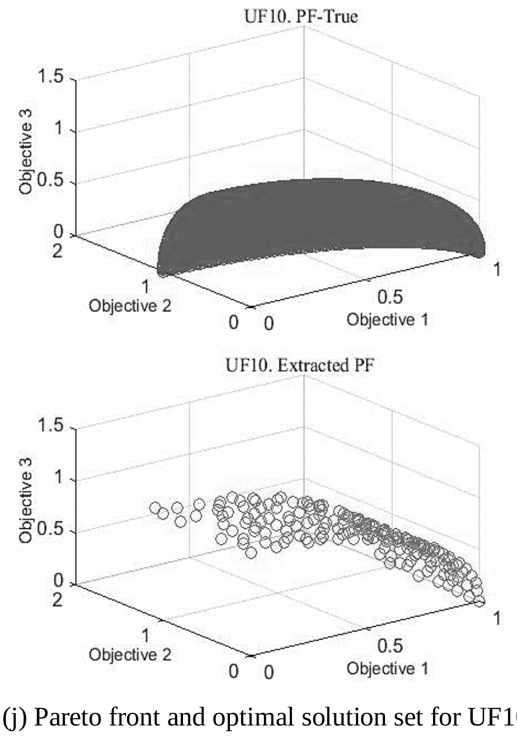
(g) Pareto front and optimal solution set for UF7



(h Pareto front and optimal solution set for UF8



(i) Pareto front and optimal solution set for UF9



(j) Pareto front and optimal solution set for UF10

Figure 2.12: Pareto-fronts computed by M^{3}D/MAS for three objective problems UF1 to UF10

Figure 2.13 (i.e. convergence graphs) illustrates that convergence speed of M^3D/MAS is much faster in comparison to its agents. It means that, the proposed method has considerably a better capability of escaping from locally optimal solutions and consequently more cooperative success than its individual component algorithms.

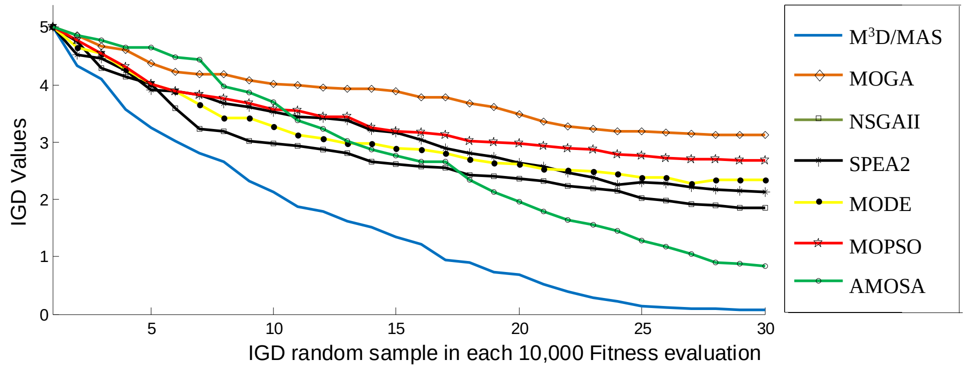


Figure 2.13: Convergence graph of M^3D/MAS and its six elements agents for test case UF5

In the next step of evaluation process, the Friedman Aligned Rank test for M^3D/MAS and other 13 competitors in CEC2009 is carried out over the obtained IGD values. The aim of Friedman Aligned Rank test is to indicate the statistical similarity between the suggested method and its competitors. Likewise, this test indicates the order of the proposed method among all methods taken into account. [45, 46] references describe the test and explain the method in detail.

The average Friedman ranks, FAR and P values of all 14 methods are represented in Table 2.9. As it can be seen from the table, M^3D/MAS has the lowest average rank of 36.1 which indicates that the proposed method outperforms other 13 algorithms. Meanwhile, since M^3D/MAS system has a very low p-value, there is a significant statistical difference between the proposed method and others.

Table 2.8: Friedman aligned ranks for algorithms and benchmark problems

Function	**MOEA/D**	**GDE3**	**MOEADGM**	**MTS**	**LiuLiAlg**	**DMOEADD**	**NSGAIILS**	**OWMOSaDE**	**Clust. MOEA**	**AMGA**	**MOEP**	**DECMOSA-SQP**	**OMOEAII**	**M^3D/MAS**
UF1	49	51	55	56	58	63	65	66	93	99	115	121	122	50
UF2	72	82	70	69	83	73	84	75	97	90	95	100	105	71
UF3	30	113	45	47	34	41	112	111	48	68	108	101	133	40
UF4	107	61	94	54	88	85	102	96	103	80	86	67	91	53
UF5	11	3	140	1	7	74	135	128	29	5	19	8	9	4
UF6	10	127	139	18	106	21	131	117	31	46	37	44	23	14
UF7	52	89	60	104	59	64	79	116	81	114	77	87	98	62
UF8	15	125	124	36	22	16	24	28	123	92	136	120	110	35
UF9	32	33	118	43	38	17	27	39	130	119	134	76	126	26
UF10	109	57	129	2	78	12	138	137	42	13	20	25	132	6
SUM	487	741	974	430	573	466	897	913	777	726	827	749	949	361
AVG	**48.7**	**74.1**	**97.4**	**43**	**57.3**	**46.6**	**89.7**	**91.3**	**77.7**	**72.6**	**82.7**	**74.9**	**94.9**	**36.1**

Table 2.9: Friedman aligned rank and p-value computed for all the methods

Algorithms	**Average values of Friedman Aligned Ranks over all problem instances**
MOEA/D	48.7
GDE3	74.1
MOEADGM	97.4
MTS	43
LiuLiAlgorithm	57.3
DMOEADD	46.6
NSGAIILS	89.7
OWMOSaDE	91.3
ClusteringMOEA	77.7
AMGA	72.6
MOEP	82.7
DECMOSA-SQP	74.9
OMOEAII	94.9
M^3D/MAS	**36.1**
F_{AR}	**34.5146**
p-value	**0.0010**

2.7.1.1 Comparing the Suggested System to the Latest Existent Methods

This section compares the proposed method against 8 latest multi-objective optimization methods published in literature. All the methods are the extended and improved versions of MOEA/D which is the best algorithm of CEC2009. The details of these new methods and their improvement process can be found in [47, 48, 49, 50, 51, 52 and 53].

The IGD values obtained by the suggested approach and all other latest methods are shown in Table 2.10. It can be seen from Table 2.10 that M^{3}D/MAS obtains the smallest IGD values for two problems out of 10 problems. Likewise MOEA/DVA, MOAD/D-HHsw and ENS-MOEA/D methods obtain the best IGD for three, three and two problems respectively.

Table 2.10: Average value of IGDs achieved by M3D/MAS and eight most newly reported methods for UF1-UF10

Function	**MOEA/D-DE+PSO**	**MOEA /DCPDE**	**MOEA /DHHsw**	**MOEA /D-DRA**	**MOEA /D-CMX +SPX**	**ENS-MOEA/D**	**BCE-MOEA/D+TCH**	**MOEA /DVA**	**M^{3}D /MAS**
UF1	0.04647	0.00520	**0.00100**	0.00152	0.00429	0.00164	0.00164	0.00413	0.00570
UF2	0.01516	0.01187	**0.00183**	0.00350	0.00561	0.00404	0.00656	0.00410	0.00740
UF3	0.00920	0.04470	0.00316	0.00394	0.01116	0.00259	0.00957	**0.02271**	0.04393
UF4	0.05796	0.04534	0.05291	0.06028	0.06414	0.04207	0.06606	0.03506	**0.03364**
UF5	0.67420	0.20344	0.27379	0.25493	0.41850	0.24811	0.40341	**0.03259**	0.08770
UF6	0.80570	0.09157	0.09633	0.32617	0.32735	0.06084	0.42500	0.05693	**0.05676**
UF7	0.48180	0.00606	**0.00110**	0.00194	0.00626	0.00172	0.01211	0.00376	0.00608
UF8	0.14980	0.12334	0.03107	0.04066	0.05744	**0.03100**	0.05610	0.05778	0.11023
UF9	0.14231	0.08002	0.04443	0.12307	0.09769	**0.02787**	0.13153	0.12333	0.07022
UF10	0.21276	0.49992	0.46722	0.40877	0.46265	0.21173	0.46496	**0.10352**	0.30490

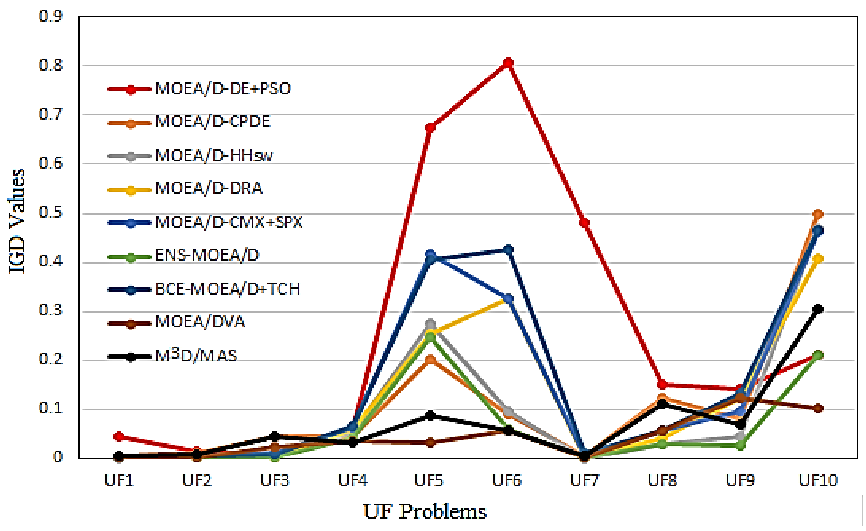


Figure 2.14: The values of IGD for CEC2009 achieved by M^{3}D/MAS and eight new MOO methods

Moreover, the graphical representation of obtained IGD values is represented in Figure 2.14 which obviously shows that despite the M3D/MAS is not best for 8 problems, it is very close to the best ones. Likewise, the small fluctuations in the plot of proposed method indicate that M3D/MAS is the most reliable method among all methods taken into account.

2.7.1.2 Extension, Replacement and Scalability Tests

The proposed method is flexible enough to add or remove a MOEA agent to/from the system. To show the related experiments, three more evaluations are done using the same proposed architecture. First one is to add MOABC method as the sixth agent to the system. MOABC is kind of simple to be implemented as it has been expressed in [54]. Afterwards, the strategy agent is verified to adjust MOABC parameters based on standard values. Later on, the obtained extended system (M^{3}D/MAS +MOABC) is applied to solve four test cases namely UF2, UF5, UF7 and UF9. The obtained results are presented in Table 2.11.

Table 2.11: Comparing of mean IGDs achieved by M3D/MAS and M3D/MAS + MOABC for some of the CEC2009 test cases

Function	M^{3}D/MAS	M^{3}D/MAS + MOABC
UF2	0.00740	0.00738
UF5	0.08770	0.08903
UF7	0.00608	0.00657
UF9	0.07022	0.07115

As it is shown in Table 2.11, a little increase in IGD values for three problems has been achieved. Nevertheless, the position of M^{3}D/MAS+ MOABC is still same as M^{3}D/MAS, when its compared to all thirteen CEC2009 competitors in Tables 2.3-2.7. Also for the test cases UF2, UF5, UF7 and UF9, the ranks of M^{3}D/MAS and M^{3}D/MAS+ MOABC are 3, 3, 4 and 2 respectively. Consequently, adding more MOEA agents to the existing system doesn't affect the success of M^{3}D/MAS.

The second experiment replaces AMOSA by MOABC in the proposed system and evaluates it using the same four CEC2009 problems in the first experiment. Table 2.12 illustrates the obtained IGD values.

Table 2.12: Comparing mean IGDs achieved by M3D/MAS and modified M3D/MAS (AMOSA ← MOABC) for some of the CEC2009 test cases

Function	M^3D/MAS	MOABC instead of AMOSA
UF2	0.00740	0.00763
UF5	0.08770	0.09084
UF7	0.00608	0.00700
UF9	0.07022	0.07219

Small increases in IGD values for UF2, UF5, UF7 and UF9 can be seen in Table 2.12. Nevertheless, the order of proposed system for UF2, UF5, UF7 and UF9 are 4, 3, 5 and 2 respectively. Even though the IGD values for this altered version is a little bit poorer than M^3D/MAS, however, it is after all the third best performing algorithm. Based on the result of these examinations, it can safely be concluded that the replacement of a MOEA agent by the suggested multi-agent framework with a new MOO algorithm does not make any notable downgrade in the performance of it.

In the third experiment, for testing the strength and stability of the proposed MAS against another replacement, MOGA algorithm used in M^3D/MAS is replaced by NSGAII and the mentioned test cases above are solved using the modified MAS. The IGD values found for experimental work are reported in Table 2.13.

Table 2.13: Comparing mean IGDs achieved by M3D/MAS and modified M3D/MAS (MOGA ← NSGAII) for some of the CEC2009 test cases

Function	M^3D/MAS	NSGAII instead of MOGA
UF2	0.00740	0.00725
UF5	0.08770	0.08693
UF7	0.00608	0.00630
UF9	0.07022	0.06858

The IGD values obtained for UF2, UF5, UF7 and UF9 are presented in Table 2.13. It can be seen that there is a little decrease in IGD values for UF2, UF5 and UF9. Consequently, the verified MAS takes the ranks equal to 4, 3, 4 and 2 for UF2, UF5, UF7 and UF9 problems respectively in comparison to all CEC2009 competitors. Therefore, there is only one rank decrease for UF2. Even though the IGD value for UF5 decreased and the position of M^3D/MAS is still third place, the position of M^3D/MAS is changed due to some increase in UF2. Eventually, it can be claimed that replacing a MOEA agent with a new one doesn't affect the success of M^3D/MAS.

Regarding scalability of the proposed methods, as it is experimented above by adding a new metaheuristic doesn't affect the performance. However adding or removing more metaheuristics reduces the performance in some cases. For example, in M3D/MAS method changing the number of metaheuristic agents from 5 to 4, 6, 7 or 8 doesn't affect the results much but reducing 5 to less than 4 or increasing 5 to more than 8 reduces the performance for four randomly selected test cases UF2, UF5, UF7 and UF9. Table 2.14 illustrates the IGD scores obtained by M^3D/MAS over four randomly selected benchmarks.

Table 2.14: scalability of the proposed methods

	Original M^3D/MAS IGD score	**Modified M^3D/MAS IGD score**				
Agents number / **Test case**	5 agents	Average IGD score of 4, 6, 7, 8 agents	3 agents	9 agents	10 agents	11 agents
UF2	0.00740	≈ 0.0082	0.018	0.011	0.013	0.017
UF5	0.08770	≈ 0.0910	0.115	0.099	0.107	0.120
UF7	0.00608	≈ 0.0069	0.009	0.008	0.009	0.009
UF9	0.07022	≈ 0.0782	0.107	0.086	0.093	0.100

2.7.2 Experimental Results and Evaluations Over the Problems in ZDT and DTLZ Benchmarks

In this part two multi-objective optimization assessment metrics, namely ε-indicator and IGD are applied to assess the performance of the Multi-Objective optimization algorithms. Brief description of the mentioned metrics can be found in [34, 35, 36 and 37]. The evaluation of the proposed algorithm is carried out over the difficult ZDT (ZDT1, ZDT2, ZDT3, ZDT4 and ZDT 6) and DTLZ (DTLZ1, DTLZ2, DTLZ3, DTLZ4 and DTLZ7) benchmark problems [55, 56]. Average scores of M^3D/MAS for ZDT and DTLZ benchmark problems over 30 runs are compared to reported results in [20, 21 and 22]. In this section all the results of best-performing methods are shown in bold.

Table 2.15 shows the variable ranges, number of variables and maximum number of fitness evaluations for each benchmark problem instance where Table 2.16 and Table 2.17 illustrate the Min, Max and Average IGD and ε –indicator values related to M^3D/MAS method for the 10 benchmark problems mentioned above. It can be seen from these tables that small values of standard deviations indicate that the proposed algorithm is a stable and robust alternative numerical MOO. Additionally, small values of IGD and ε–indicator metrics show that the proposed algorithm successfully extracts Pareto fronts that close to the optimal one.

Table 2.15: Variable ranges, number of variables and maximum value for fitness evaluations of each benchmark problem instance

Instance	Number and Range of Variables	# Fitness Evaluations
ZDT1	$[0, 1]^n$ **n** = 30	300 000
ZDT2	$[0, 1]^n$ **n** = 30	300 000
ZDT3	$[0, 1]^n$ **n** = 30	300 000
ZDT4	$x_1 \in [0,1]$, $x_i \in [-5,5]$, **n** = 30	300 000
ZDT6	$[0, 1]^n$ **n** = 30	300 000
DTLZ1	$[0, 1]^n$ **n** = 30	300 000
DTLZ2	$[0, 1]^n$ **n** = 30	300 000
DTLZ3	$[0, 1]^n$ **n** = 30	300 000
DTLZ4	$[0, 1]^n$ **n** = 30	300 000
DTLZ7	$[0, 1]^n$ **n** = 30	300 000

Table 2.16: Lowest, highest and average values of IGD in 30 runs for M3D/MAS

Function	Average	Min	Max	Std
ZDT1	3.19e – 03	2.96e – 03	3.58 e – 03	2.1e – 04
ZDT2	3.33e – 03	3.08e – 03	3.63 e – 03	2.0e – 04
ZDT3	3.77e – 03	3.50e – 03	4.33e – 03	2.2e – 04
ZDT4	3.01e – 03	2.59e – 03	3.52e – 03	2.3e – 04
ZDT6	2.54e – 03	2.30e – 03	2.89e – 03	1.9e – 04
DTLZ1	1.40e – 03	1.37e – 03	1.55e – 03	1.2e – 04
DTLZ2	2.97e – 03	2.77e – 03	3.23e – 03	2.0e – 04
DTLZ3	3.47e – 03	3.36e – 03	3.59e – 03	1.8e – 04
DTLZ4	2.95e – 03	2.82e – 03	3.17e – 03	1.7e – 04
DTLZ7	5.55e – 03	5.35e – 03	5.95e – 03	4.2e – 04

Table 2.17: Lowest, highest and average values of ε–indicator in 30 runs for M3D/MAS

Function	Average	Min	Max	Std
ZDT1	5.30e – 03	4.80e – 03	5.66e – 03	3.6e – 04
ZDT2	4.96e – 03	4.75e – 03	5.13e – 03	2.7e – 04
ZDT3	5.34e – 03	4.90e – 03	5.78e – 03	4.3e – 04
ZDT4	4.36e – 03	4.08e – 03	4.88e – 03	4.1e – 04
ZDT6	4.04e – 03	3.89e – 03	4.47e – 03	3.2e – 04
DTLZ1	2.27e – 03	2.16e – 03	2.44e – 03	1.8e – 04
DTLZ2	4.29e – 03	3.79e – 03	4.70e – 03	3.2e – 04
DTLZ3	5.63e – 03	5.40e – 03	6.05e – 03	3.8e – 04
DTLZ4	5.87e – 03	5.72e – 03	6.22e – 03	3.1e – 04
DTLZ7	8.75e – 03	8.31e – 03	9.32e – 03	5.4e – 04

Table 2.18 exhibits IGD results of M^3D/MAS and 5 other recently published state-of-the-art MOO methods for comparative evaluations. For ZTD benchmarks, the best

performing algorithm is Blend Crossover with Non-Uniform Mutation (BX-NU) that got the best IGD values for 3 test problems whereas M^3D /MAS took the second position and exhibited the best scores for 2 of ZDT instances. For the 6 DTLZ instances, Blend Crossover with Non-Uniform Mutation (BLX-NU) [57] and Simulated Binary Crossover with Polynomial Mutation (SBX-PN) [57] are the best performing methods while the proposed algorithm is taking the third position.

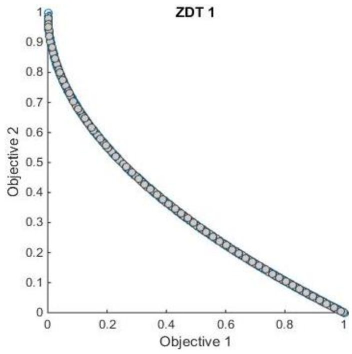
Table 2.18: Comparing average IGD values of M3D/MAS with 5 MOO methods

Function	BLX-NU [57]	SBX-PN [57]	MOEA/D [39]	MOEA/D-DE+PSO [49]	MOEA/D-CPDE [47]	M³D/MAS
ZDT1	5.58e – 03	2.99e – 02	4.05e – 03	4.14e – 03	4.03e – 03	**3.19e – 03**
ZDT2	4.37e – 03	7.44e – 03	3.81e – 03	3.87e – 03	3.80e – 03	**3.33e – 03**
ZDT3	**3.10e – 03**	5.92e – 03	7.08e – 03	9.02e – 03	7.08e – 03	3.77e – 03
ZDT4	**7.90e – 04**	2.15e – 03	1.96e – 01	7.55e – 03	3.95e – 03	3.01e – 03
ZDT6	**1.07e – 03**	1.22e – 03	1.34e – 02	1.45e – 02	5.97e – 03	2.54e – 03
DTLZ1	**2.70e – 04**	5.50e – 04	NA	NA	NA	1.40e – 03
DTLZ2	**7.60e – 04**	2.68e – 03	NA	NA	NA	2.97e – 03
DTLZ3	4.00e – 04	**3.90**e – 04	NA	NA	NA	3.47e – 03
DTLZ4	2.93e – 03	**2.74**e – 03	NA	NA	NA	2.95e – 03
DTLZ7	6.96e – 03	1.17e – 02	NA	NA	NA	**5.55e – 03**

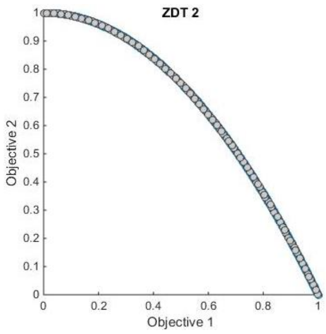
Table 2.19 illustrates ε-indicator values of 7 MOO metaheuristics including M^3D /MAS and 6 other well-known MOO methods. It clear that the proposed methods is the winner against its competitors and achieved the best scores for 6 of the 10 benchmark problems. The second best performing method is SMPSO [58] that took the first position for 4 test instances.

Table 2.19: Comparing the average ε values of M3D/MAS with 6 MOO methods

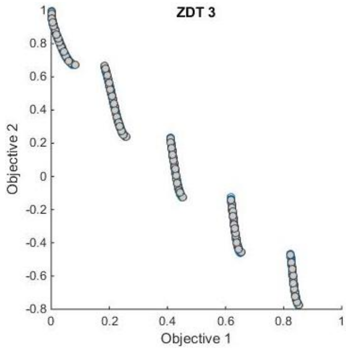
Function	NSGA-II	SPEA2	OMOPSO	AbYSS	MOCell	SMPSO	M³D/MAS
ZDT1	1.37e – 02	8.69e – 03	6.36e – 03	7.72e – 03	6.23e – 03	5.39e – 03	**5.30e – 03**
ZDT2	1.28e – 02	8.73e – 03	6.19e – 03	7.10e – 03	5.57e – 03	5.33e – 03	**4.96e – 03**
ZDT3	8.13e – 03	9.72e – 03	1.32e – 02	6.10e – 03	5.66e – 03	**5.10e – 03**	5.34e – 03
ZDT4	1.49e – 02	3.42e – 02	5.79e + 00	1.14e – 02	8.17e – 03	6.02e – 03	**4.36e – 03**
ZDT6	1.47e – 02	2.42e – 02	4.65e – 03	5.06e – 03	6.53e – 03	4.43e – 03	**4.04e – 03**
DTLZ1	7.13e – 03	5.89e – 03	1.92e + 01	5.85e – 03	4.02e – 03	2.97e – 03	**2.27e – 03**
DTLZ2	1.11e – 02	7.34e – 03	6.72e – 03	5.39e – 03	5.09e – 03	5.17e – 03	**4.29e – 03**
DTLZ3	1.04e + 0	2.28e + 00	8.86e + 01	1.66e + 00	7.91e – 01	**5.39e – 03**	5.63e – 03
DTLZ4	1.13e – 02	7.66e – 03	3.18e – 02	**5.39e – 03**	5.74e – 03	**5.39e – 03**	5.87e – 03
DTLZ7	1.04e – 02	9.09e – 03	7.13e – 03	5.51e – 03	5.19e – 03	**4.95e – 03**	8.75e – 03



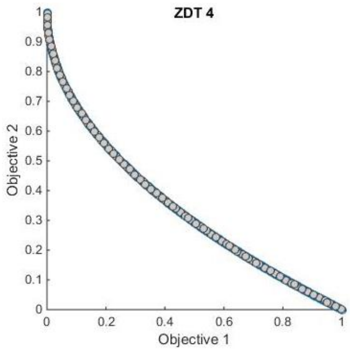
a)) Pareto front and optimal solution set for ZDT1



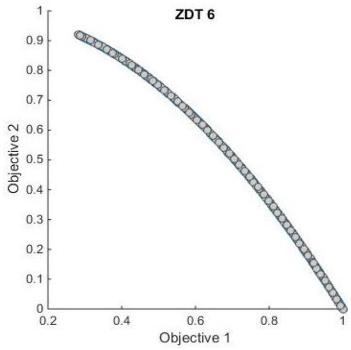
b) Pareto front and optimal solution set for ZDT2



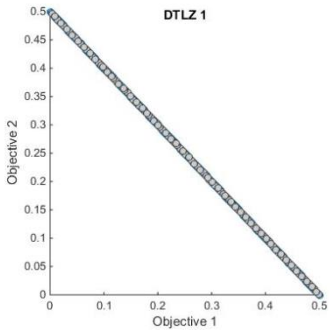
c) Pareto front and optimal solution set for ZDT3



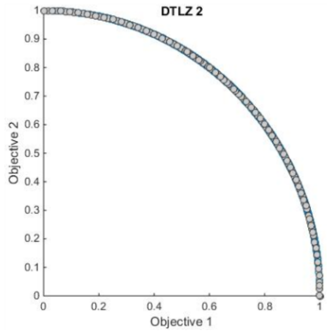
d)) Pareto front and optimal solution set for ZDT4



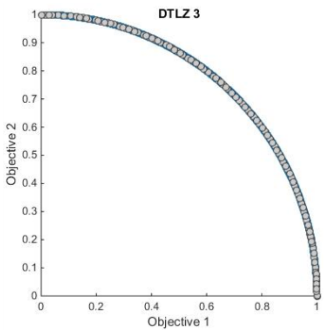
e) Pareto front and optimal solution set for ZDT6



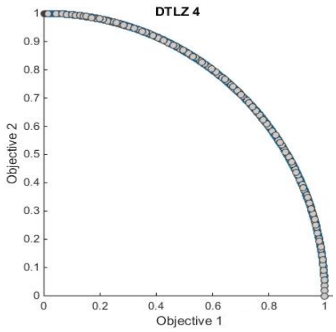
f)) Pareto front and optimal solution set for DTLZ1



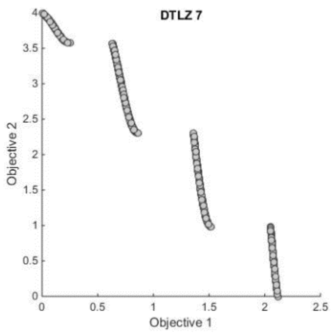
g)) Pareto front and optimal solution set for DTLZ2



h) Pareto front and optimal solution set for DTLZ3



i) Pareto front and optimal solution set for DTLZ4



j) Pareto front and optimal solution set for DTLZ7

Figure 2.15: Plots of pareto fronts computed by M^3D/MAS and optimal pareto fronts of the ten ZDT and DTLZ benchmark problems
Subplots a-e illustrate the computed and optimal PFs of ZDT benchmarks, whereas subplots f-j show those for the DTLZ test instances

Figure 2.15 shows the pareto fronts computed by M^3D/MAS and the optimal pareto fronts for the ZDT and DTLZ benchmark instances. It is seen that both the spread and convergence of the computed Pareto fronts are close to optimal for all the problems under consideration.

Chapter 3

A DYNAMIC METAHEURISTIC NETWORK FOR NUMERICAL MULTI-OBJECTIVE OPTIMIZATION

3.1 Introduction

This section introduces a novel dynamic framework for the collaboration of metaheuristics. The proposed framework consists of a network in which the node and edges represent the metaheuristics and flows respectively. The flows on the network indicate the move of sub-populations through the metaheuristics. The main goal of the proposed network is to use a collection of metaheuristics in such a way that capabilities of a metaheuristic can cover the inabilities of others. In other words the ensemble of metaheuristics is able to work more robust rather than individual metaheuristics. The suggested network distributes the sub-populations over metaheuristics dynamically and this is of great importance in the research. Another remarkable contribution of the study is that network has a dynamic structure to make use of diverse methods and operators.

The evaluation of the proposed framework over well-known test problems proves the effectiveness of the proposed network in solving MOO problems. The suggested network comprising seven multi-objective optimization algorithms is named as Dynamic Metaheuristic Network (DMN) in which the participated algorithms deal with the sub-populations through the sessions.

The structure of proposed network is in the form of 3-3-1 with three layers. The nodes and edges of the network represent the multi-objective optimization algorithms and transmission of sub-populations respectively.

A number of consecutive sessions are carried out in the process of the suggested framework. Each session accomplishes the following steps:

- The sub-populations are assigned to the metaheuristics (nodes).
- The metaheuristics are performed on their assigned sub-populations.
- The improved sub-populations are transmitted to the neighborhood metaheuristics (nodes) connected by the edges.

The size of sub-populations differs from layer to layer in which the size of first layer and output layer are smallest and largest respectively. Likewise, the number of nodes in a layer specifies the size for middle layer. At the end of each session the layers are changed by rotation operator in which the nodes are shifted one layer forward. In the shift operation all metaheuristics (nodes) are shifted one layer forward and the last layer becomes the first one.

Each session ends up with transferring the enhanced sub-populations to all connected neighborhood metaheuristics. Likewise, all NDS solutions found so far are kept in a global NDS set by updating the set with recently found NDS solutions.

The main idea of the proposed framework is to improve the population by dividing it to the sub-populations and enhance them by the collaboration of various metaheuristics. This way the bias of modifying the mechanisms is also eliminated.

CEC2009 and CEC2017 are selected as the main benchmark to evaluate the proposed network system. Furthermore, the suggested method is tested over some real-world multi-objective optimization problems. For all test cases and benchmark problems the parameter values are adjusted according to the related literature.

The quality of obtained results indicates that the proposed innovative system is quite successful in extracting high quality solutions and discovering a good Pareto-front. Likewise, the quality of discovered Pareto-front indicates that the ensemble of metaheuristics improves the optimization power and extracts high quality solutions. Section 5 includes the full explanation of the suggested method.

3.2 Multi-objective Optimization Metaheuristics Used in the System

The proposed network consists of seven multi-objective optimization algorithms namely Multi-objective Artificial Bee Colony (MOABC), Multi-objective Genetic Algorithm (MOGA), Multi-objective Differential Evolution (MODE), Multi-objective Particle Swarm Optimization (MOPSO), Strength Pareto Evolutionary Algorithm 2 (SPEA2), Non-dominated Sorting Genetic Algorithm (NSGAII) and (AMOSA). These algorithms are already explained in Section 2.2.

3.3 New Multi-objective Optimization Algorithms

Some methods among new multi-objective optimization approaches are selected to be used in the evaluation process of the proposed method (DMN). All the selected methods which are the improved versions of MOEA/D were explained in Section 2.7.1.1.

3.4 Ensembles of Metaheuristics as Multi-agent Systems

Metaheuristics is kind of search algorithms to process the search space based on the objective functions, protect the high quality solutions and move towards the best solutions. It’s been experienced by researchers that an algorithm can be successful for some problems and unsuccessful against others. This issue happens because of the search bias affected by the operators defined for the algorithms. Likewise for any metaheuristic, the operators are able to move towards the particular directions on the search space and they don’t extract some parts of the search space. Meanwhile, the performance of a metaheuristic is affected by its neighborhood definition.

Therefore, using an ensemble of metaheuristics and combination of different methods can help the search process to search more parts of the space and extract more promising solutions. This way, a collection of metaheuristics can collaborate and cooperate altogether to cover the individual weaknesses and fulfill the search task. Ensemble systems are constructed in the form of multi-agent systems based on the aforementioned aspects.

A multi-agent system is a framework consisting of a collection of agents collaborating together to achieve some predefined goals. In a multi-agent system several metaheuristic algorithms are employed as problem solvers to deal with MOO problems.

3.4.1 Multi-agent Systems in Multi-objective Optimization

Nowadays, the metaheuristics are widely used to deal with multi-objective optimization problems. NSGA II, MOGA, SPEA 2, MODE, AMOSA, MOPSO and MOABC are some of the most succesfull metaheuristics so that their robustness has been proven by the experiments carried out by researchers. These metaheuristics are used as nodes in the proposed network system. Detailed descriptions of these multi-objective optimization metaheuristics are given in [7, 8, 9, 10, 11, 12, 13 and 14] and also brief description of them are available in Section 2.2.

Meanwhile, some researchers have worked on the ensemble of multi-objective metaheuristics topic which tries to use the collection of metaheuristics together to deal with optimization problems. Sections 2.4.1 and 2.4.2 review and describe the state-of-the-art methods related to this topic.

The suggested innovative method employes the multi-objective optimization algorithms in the dynamic form of a multilayer network. The nodes of proposed network represent the metaheuristics in which each metaheuristic deals with a subset of population. Since the nodes of metaheuristics are changed once they are done with the assigned subpopulations, the proposed network is call as dynamic network. Section 3.5 describe the suggested network model in details.

3.5 A Dynamic Network Framework for MO Optimization

The proposed DMN (Dynamic Metaheuristic Network) method is introduced in this section which is an ensemble approach to deal with multi-objective optimization problems. The architecture of the DMN system is presented in Figure 3.1. As it can be seen from the figure, there exist 7 different MO metaheuristics in which they cooperate and collaborate together to extract a high quality Pareto-front for the optimization problems. The metaheuristics work on the assigned subpopulations and share the improved solutions.

There are three layers based on 3-3-1 topology in the proposed network model. Each node in the network is assigned a metaheuristic and is connected to the nodes in the next layer. The edges in the network are used to transfer the sub-populations between the nodes.

Apart from the fact that the network topology is static, all the nodes are changed in a dynamic way in which all nodes are rotated and consequently the output layer becomes the first layer.

The proposed network framework iterates the phases consecutively. The following operations are carried out in each phase:

- The sub-populations are assigned to the nodes (metaheuristics).
- The metaheuristics of nodes are performed on assigned sub-populations.
- The improved sub-populations are transferred to the neighborhood nodes.

Once a phase ends up, the nodes of all metaheuristics are changed. This is the reason why the proposed architecture is called dynamic. The node change is carried out to eliminate the position bias. As an example, suppose that the position of metaheuristics is as follows in the session 1:

- The first layer includes Metah1, Metah2 and Metah3.
- The second layer consists of Metah4, Metah5 and Metah6.
- The third layer encompasses Metah7.

These positions will be modified in second session as following:

- The first layer includes Metah7, Metah1 and Metah2.
- The second layer consists of Metah3, Metah4 and Metah5.
- The third layer encompasses Metah6.

According to the proposed strategies, the first population is generated randomly, and all metaheuristics are distributed over the nodes in the system. Thereafter the general population is divided into n sub-populations where n is the number of metaheuristics. The population is divided into n sub-populations (but not with equal number of individuals in each) where n is the number of metaheuristics used in the network. The population is divided in a way that the size of sub-populations in layers 2 and 3 are third and ninth times of the size of sub-populations in layer 1 respectively. It means that if the population size in layer1 is S, the sizes in layer 2 and layer 3 are 3×S and 9×S respectively. Consequently, the size of the general population is 21×S. Therefore, to have a right number of individuals to be distributed among the metaheuristic agents, the population size should be divisible by 21 as it is shown in Table 3.1.

Table 3.1: Subpopulation size/distribution

S	Layer 1 (S individuals per node)	Layer 2 (3*S individuals per node)	Layer 3 (9*S individuals)	Total individuals of the general population
1	1+1+1	3+3+3	9	21
2	2+2+2	6+6+6	18	42
3	3+3+3	9+9+9	27	63
4	4+4+4	12+12+12	36	84
⋮	⋮	⋮	⋮	⋮
n	n+n+n	3(n+n+n)	9(n)	21(n)

In the next sessions, the sub-populations are provided by combining the improved sub-populations obtained by the node or other nodes. This operation is carried out as following:

i) Since the previous layer of layer1 is layer3, the members of sub-populations for layer1 are selected from either layer1 or layer3. The probability of this selection operator is 0.5 for both.

ii) The members of sub-populations for layer2 either come from layer1 randomly or remain no change.

iii) Likewise, the members of sub-populations for layer3 either are selected from layer2 randomly or stay no change.

The discovered non-dominated solutions are kept for each metaheuristic during a session. These non-dominated solutions are updated as following when a session finish. Suppose that R is the count of members in all non-dominated solution sets in which the portion of each metaheuristic is supposed to be R/7. To protect the size of non-dominated solutions set for each metaheuristic, if the size after update operation is less than R/7 then extra random solutions are added to the set. Similarly, if the size of set either exceeds R/7 or be equal to R/7, the normal update operation is carried out. The update operation is same as that of sub-populations update method. When the final session terminates, all non-dominated sets extracted by 7 metaheuristics are merged to form the last Pareto-front.

The proposed system provides a framework for the metaheuristics to collaborate and cooperate together to fulfill the tasks and achieve the goals. The cooperation is accomplished by sharing the extracted solutions and experiences. To cooperate, the metaheuristics use their own solutions and the solutions transferred from the nodes in previous layer to speed up the improvement process. This is how the algorithms help each other and cooperate. Likewise, at the end of sessions, the local non-dominated sets for all metaheuristics are updated and shared. Hence, the metaheuristics can use the improved sub-populations and Pareto-fronts obtained by the other metaheuristics. Moreover, when the final session finishes all extracted Pareto-fronts are merged into a final Pareto-front.

It can be seen from Figure 3.1 that there exists 3 elements for the nodes in the sugessted network system. The first element is the metaheuristics being performed in the nodes. The second includes the sub-populations assigned to the nodes and the third element is the local archives to keep the non-dominated solutions found so far.

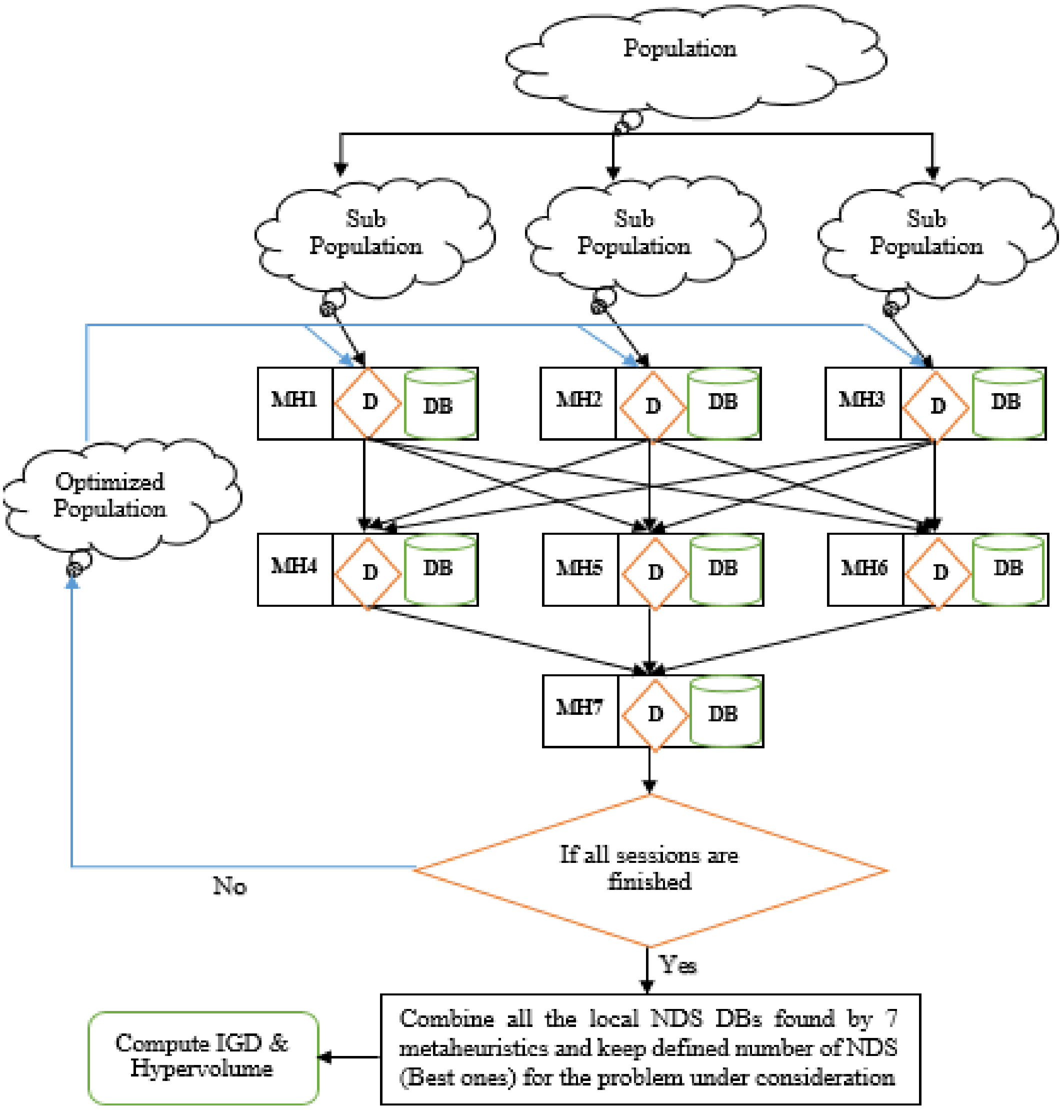


Figure 3.1: Architecture of dynamic metaheuristic network

3.6 Evaluations and Results

This section evaluates the effectiveness of the suggested method using well-known benchmarks. The first evaluation is carried out over CEC2009 benchmark problems which are named as UF1 to UF10. Among these problems UF1, Uf2, Uf3, Uf4, Uf5, Uf6 and UF7 are bi-objective problems and UF8, UF9 and UF10 are 3-objective problems [45]. The evaluation process proceeds with the next benchmarks named as WFG, DTLZ,which have two and three objectives, and ZDT which has two objectives [56]. Likewise the evaluation of the proposed method is accomplished over some real-world multi-objective problems [59]. To conduct these evaluations all parameters are adjusted according to section 2.7.1. Also the IGD and hypervolume values are calculated for thirthy different executions of the method on each problem. The IGD and hypervolume values are used to compare the proposed method to the existent approaches. The average IGD and standard deviation values for UF1-UF10 problems are represented in Table 3.2. As well as, the Minimum and Maximum values of IGD is given in the table. A method is stronger when it has lower IGD and standard deviation. As it can be seen from the table, UF5, UF6, UF8, UF9 and UF10 have bigger IGD values. This is the reason why the rank of the proposed method given in Table 3.5-3.7 is not so good.

Table 3.2: Obtained max, min and standard deviation values by DMN in 30 different executions

Function	Average	Min	Max	Std
UF1	0.00548	0.00496	0.00683	0.00034
UF2	0.00613	0.00588	0.00740	0.00033
UF3	0.05287	0.04903	0.06021	0. 04234
UF4	0.03390	0.02822	0.04263	0.00350
UF5	0.08424	0.06933	0.11245	0.00924
UF6	0.08277	0.07546	0.12340	0.01209
UF7	0.00568	0.00502	0.00698	0.00125
UF8	0.08863	0.08200	0.12105	0.00791
UF9	0.09030	0.08272	0.09900	0.03871
UF10	0.31648	0.28419	0.36618	0..01983

The rank values for the suggested method and all competitor methods of CEC2009 are given in Tables 3.2, 3.3, 3.4, 3.5 and 3.6. In these tables the rank values for the most effective methods are bolded. According to the computed results shown in the tables, for four problems (UF2, UF4, UF5 and UF10) the proposed method is much better than MOEA/D which is the best method of CEC1009. Likewise, the ranks of the proposed method are 1, 2 and 3 for UF2, (UF7 and UF10) and (UF1, UF4 and UF5) respectively.

The rank value of the proposed method for UF3, UF6, UF8 and UF9 is 5th among 14 algorithms. In order to compare the proposed method to MTS method, with rank 2 in CEC2009, it can be said that DMN outperforms MTS in six problems among 10 problems. Meanwhile, the results show that the proposed method outperforms DMOEADD, LiuLi and GDE3 for 6, 8 and 6 problems respectively. These three algorithms take the ranks 3, 4 and 5 in CEC2009.

Table 3.3: Average values of IGDs for UF1 and UF2 achieved by DMN and its competitors

Rank	UF1	Mean±Std (IGD)	UF2	Mean±Std (IGD)
1	**MOEA/D**	0.00435±0.0002	**DMN**	**0.00613**±0.0003
2	GDE3	0.00534±0.0003	MTS	0.00615±0.0005
3	**DMN**	**0.00548**±0.0003	MOEADGM	0.00640±0.0007
4	MOEADGM	0.00620±0.0010	DMOEADD	0.00679±0.0020
5	MTS	0.00646±0.0003	**MOEA/D**	0.00679±0.0018
6	LiuLiAlgorithm	0.00785±0.0020	OWMOSaDE	0.00810±0.0023
7	DMOEADD	0.01038±0.0023	GDE3	0.01195±0.0015
8	NSGAIILS	0.01153±0.0073	LiuLiAlgorithm	0.01230±0.0033
9	OWMOSaDE	0.01220±0.0012	NSGAIILS	0.01237±0.0091
10	ClusteringMOEA	0.02990±0.0022	AMGA	0.01623±0.0031
11	AMGA	0.03588±0.0102	MOEP	0.01890±0.0038
12	MOEP	0.05960±0.0128	ClusteringMOEA	0.02280±0.0078
13	DECMOSA-SQP	0.07702±0.0393	DECMOSA-SQP	0.02834±0.0313
14	OMOEAII	0.08564±0.0040	OMOEAII	0.03057±0.0016

Table 3.4: Average values of IGDs for UF3 and UF4 achieved by DMN and its competitors

Rank	UF3	Mean±Std (IGD)	UF4	Mean±Std (IGD)
1	**MOEA/D**	0.00742±0.0058	MTS	0.02356±0.0006
2	LiuLiAlgorithm	0.01497±0.0240	GDE3	0.02650±0.0003
3	DMOEADD	0.03337±0.0056	**DMN**	**0.03390**±0.0035
4	MOEADGM	0.04900±0.0659	DECMOSA-SQP	0.03392±0.0053
5	**DMN**	**0.05287**±0.04234	AMGA	0.04062±0.0017
6	MTS	0.05310±0.0117	DMOEADD	0.04268±0.0013
7	ClusteringMOEA	0.05490±0.0453	MOEP	0.04270±0.0834
8	AMGA	0.06998±0.0139	LiuLiAlgorithm	0.04350±0.0006
9	DECMOSA-SQP	0.09350±0.1979	OMOEAII	0.04624±0.0009
10	MOEP	0.09900±0.0132	MOEADGM	0.04760±0.0026
11	OWMOSaDE	0.10300±0.0190	OWMOSaDE	0.05130±0.0019
12	NSGAIILS	0.10603±0.0686	NSGAIILS	0.05840±0.0051
13	GDE3	0.10639±0.0129	ClusteringMOEA	0.05850±0.0072
14	OMOEAII	0.27141±0.0376	**MOEA/D**	0.06385±0.0053

Table 3.5: Average values of IGDs for UF5 and UF6 achieved by DMN and its competitors

Rank	UF5	Mean±Std (IGD)	UF6	Mean±Std (IGD)
1	MTS	0.01489±0.0032	**MOEA/D**	0.00587±0.0017
2	GDE3	0.03928±0.0039	MTS	0.05917±0.0106
3	**DMN**	**0.08424**±0.0092	DMOEADD	0.06673±0.0238
4	AMGA	0.09405±0.0120	OMOEAII	0.07338±0.0024
5	LiuLiAlgorithm	0.16186±0.0282	**DMN**	**0.08277**±0.0120
6	DECMOSA-SQP	0.16713±0.0895	ClusteringMOEA	0.08710±0.0076
7	OMOEAII	0.16920±0.0039	MOEP	0.10310±0.0345
8	**MOEA/D**	0.18071±0.0681	DECMOSA-SQP	0.12604±0.5617
9	MOEP	0.22450±0.0344	AMGA	0.12942±0.0565
10	ClusteringMOEA	0.24730±0.1307	LiuLiAlgorithm	0.17555±0.0829
11	DMOEADD	0.31454±0.0465	OWMOSaDE	0.19180±0.0290
12	OWMOSaDE	0.43030±0.0174	GDE3	0.25091±0.0195
13	NSGAIILS	0.56570±0.1827	NSGAIILS	0.31032±0.1913
14	MOEADGM	1.79190±0.3181	MOEADGM	0.55630±0.1959

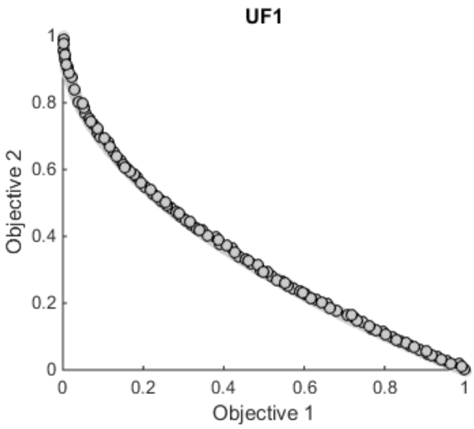
Table 3.6: Average values of IGDs for UF7 and UF8 achieved by DMN and its competitors

Rank	UF7	Mean±Std (IGD)	UF8	Mean±Std (IGD)
1	**MOEA/D**	0.0044±0.001	**MOEA/D**	0.0584±0.003
2	**DMN**	**0.0056**±0.001	DMOEADD	0.0684±0.009
3	LiuLiAlgorithm	0.0073±0.000	LiuLiAlgorithm	0.0823±0.007
4	MOEADGM	0.0076±0.000	NSGAIILS	0.0863±0.012
5	DMOEADD	0.0103±0.002	**DMN**	**0.0886**±0.007
6	MOEP	0.0197±0.075	OWMOSaDE	0.0945±0.011
7	NSGAIILS	0.0213±0.019	MTS	0.1125±0.012
8	ClusteringMOEA	0.0223±0.004	AMGA	0.1712±0.017
9	DECMOSA-SQP	0.0241±0.022	OMOEAII	0.1920±0.012
10	GDE3	0.0252±0.008	DECMOSA-SQP	0.2158±0.121
11	OMOEAII	0.0335±0.001	ClusteringMOEA	0.2383±0.034
12	MTS	0.0407±0.014	MOEADGM	0.2446±0.044
13	AMGA	0.0570±0.065	GDE3	0.2485±0.035
14	OWMOSaDE	0.0585±0.029	MOEP	0.4230±0.056

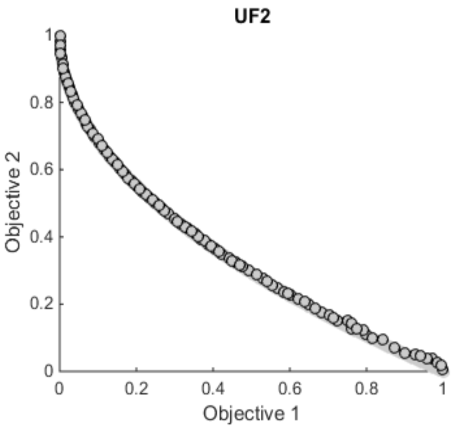
Table 3.7: Average values of IGDs for UF9 and UF10 achieved by DMN and its competitors

Rank	UF9	Mean±Std (IGD)	UF10	Mean±Std (IGD)
1	DMOEADD	0.0489±0.009	MTS	0.1530±0.015
2	NSGAIILS	0.0719±0.045	**DMN**	**0.3164**±0.019
3	**MOEA/D**	0.0789±0.053	DMOEADD	0.3221±0.022
4	GDE3	0.0824±0.022	AMGA	0.3241±0.095
5	**DMN**	**0.0903**±0.038	MOEP	0.3621±0.044
6	LiuLiAlgorithm	0.0939±0.047	DECMOSA-SQP	0.3698±0.653
7	OWMOSaDE	0.0983±0.024	ClusteringMOEA	0.4111±0.080
8	MTS	0.1144±0.025	GDE3	0.4332±0.012
9	DECMOSA-SQP	0.1411±0.345	LiuLiAlgorithm	0.4469±0.129
10	MOEADGM	0.1878±0.044	**MOEA/D**	0.4741±0.073
11	AMGA	0.1886±0.042	MOEADGM	0.5646±0.069
12	OMOEAII	0.2317±0.064	OMOEAII	0.6275±0.145
13	ClusteringMOEA	0.2934±0.085	OWMOSaDE	0.7430±0.088
14	MOEP	0.3420±0.158	NSGAIILS	0.8446±0.162

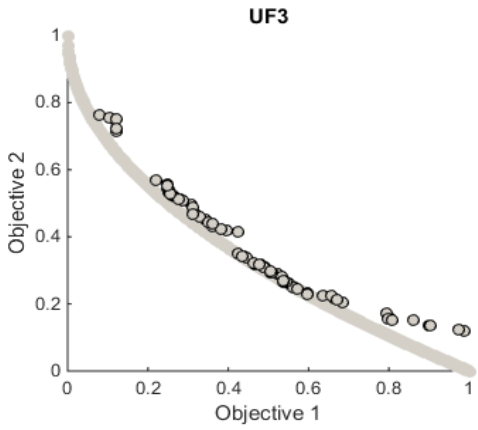
The extracted PF's by the proposed method for UF1-UF10 problems are illustrated in Figure 3.2. Also the optimal PF's are represented in the figure. According to CEC2009, the number of nodes in PF for bi-objective and 3-objectives are considered as 100 and 150 respectively. It can be seen from Figure 3.2 that the PF extracted by the proposed method is very close to the optimal PF and is well-spread.



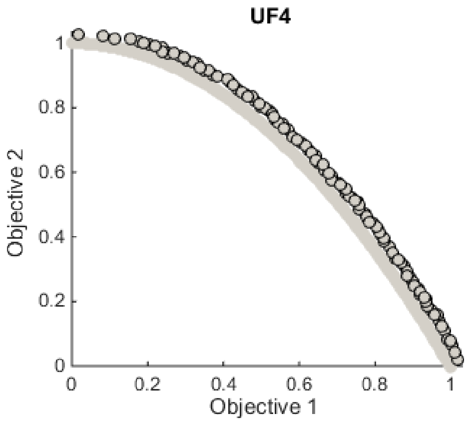
Pareto front and optimal solution set for UF1



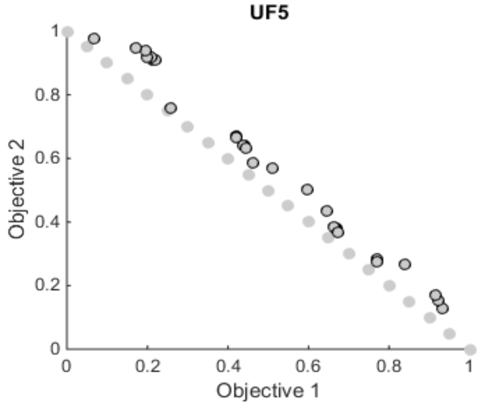
Pareto front and optimal solution set for UF2



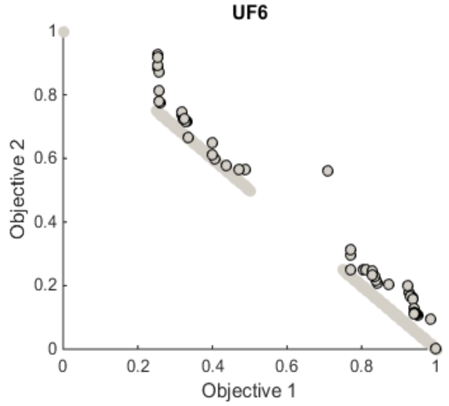
Pareto front and optimal solution set for UF3



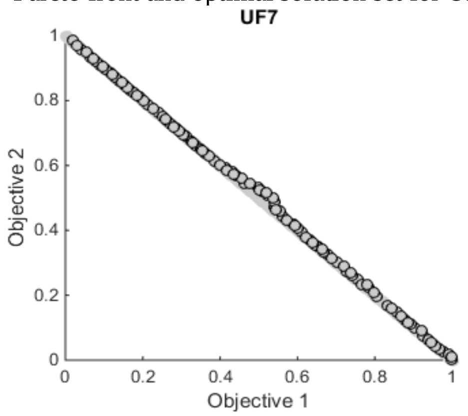
Pareto front and optimal solution set for UF4



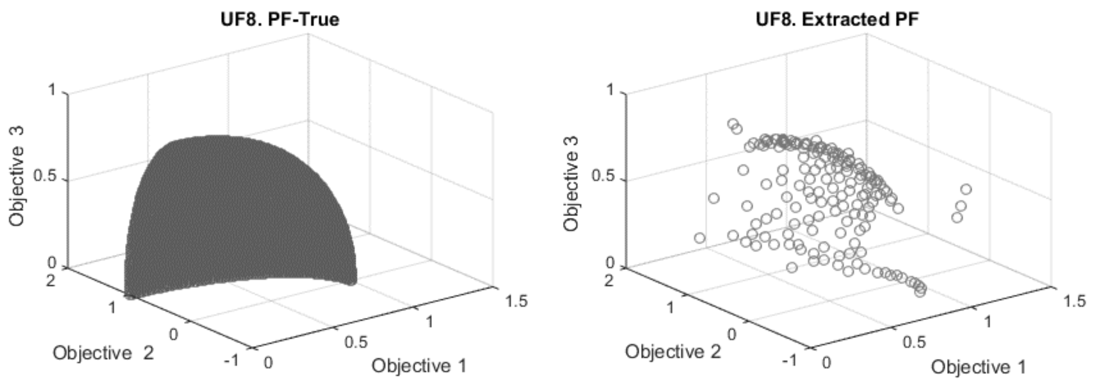
Pareto front and optimal solution set for UF5



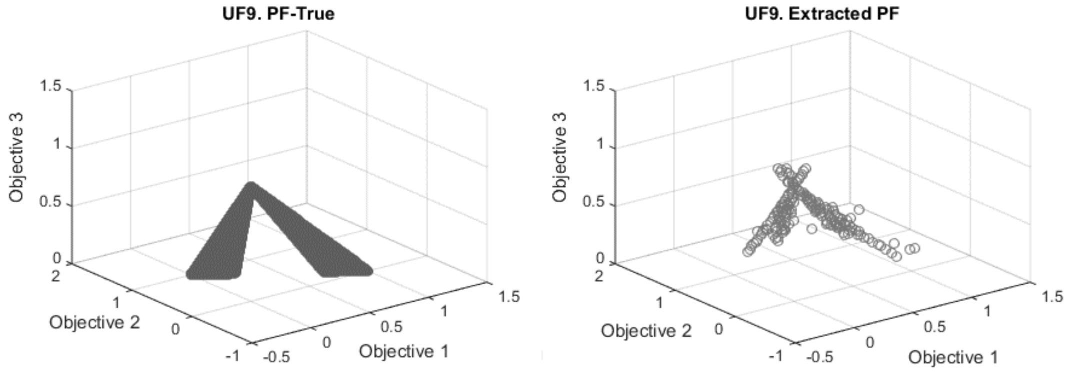
Pareto front and optimal solution set for UF6



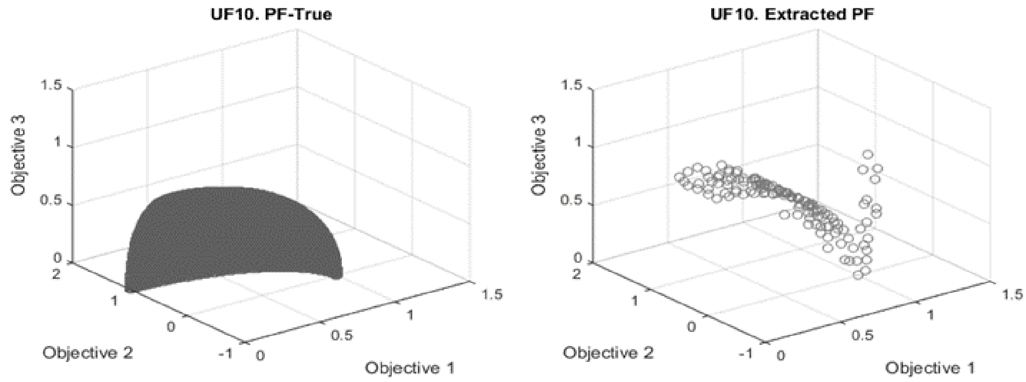
Pareto front and optimal solution set for UF7



Pareto front and optimal solution set for UF8



Pareto front and optimal solution set for UF9



Pareto front and optimal solution set for UF10

Figure 3.2: Pareto-fronts computed by DMN for problems UF1 to UF10

In Figure 3.3 are shown the average and rank of the IGD's provided by the proposed method and other algorithms for UF1 to UF10 test cases correspondingly in Figure 3.3 and 3.4. As indicates in these figures, the result of suggested method (shown in red) is very nearby or even better than MOEAD (shown in black) -the winner of the CEC2009 contest- in the majority of test instances.

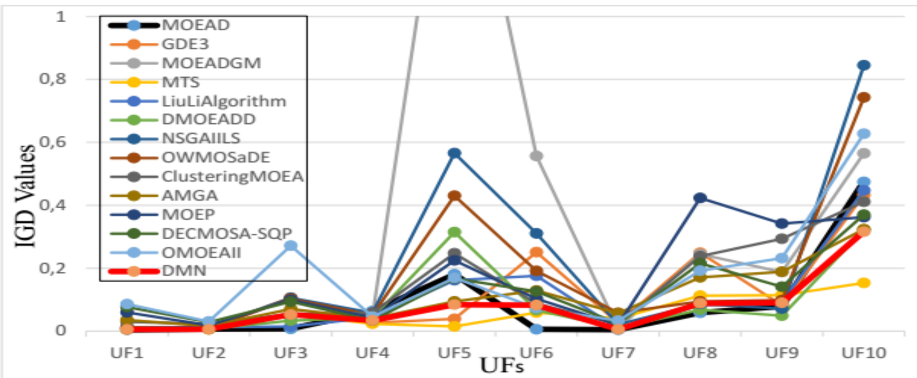


Figure 3.3: Average values of IGD for UF1 to UF10 achieved by DMN and its competitors

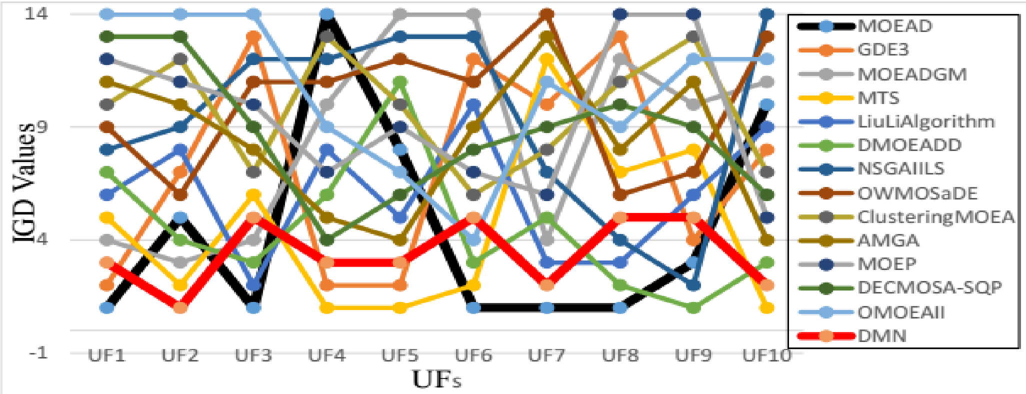


Figure 3.4: Rank values obtained for DMN and its competitors for test instances of UF1 to UF10

The Friedman Aligned Ranks test for the proposed method and all other methods of CEC2009 is carried out over the obtained results and is represented in Table 3.8. The test is done to show the similarity of all algorithms taken into account and also to find the rank for the proposed method. All the details regarding to this test approach is given in [45, 46]. Eventually, the FAR value, p value and average of Friedman ranks are indicated in Table 3.9. As it can be seen from the table, the lowest average value belongs to the proposed method which is 45.6. This means that the proposed method is the best method among all algorithms under consideration. Moreover, the proposed method has a very low p value which means that there is a significant difference between the proposed method and other algorithms.

Table 3.8: Friedman aligned ranks for all benchmarks and algorithms

Function	**MOEA/D**	**GDE3**	**MOEADGM**	**MTS**	**LiuLiAlgorithm**	**DMOEADD**	**NSGAIILS**	**OWMOSaDE**	**ClusteringMOEA**	**AMGA**	**MOEP**	**DECMOSA-SQP**	**OMOEAII**	**DMN**
UF1	50	52	55	56	58	62	63	65	94	101	115	119	122	53
UF2	73	81	71	70	84	73	85	76	98	91	95	104	109	69
UF3	47	125	100	103	48	74	124	123	107	113	121	120	135	102
UF4	111	59	93	51	88	86	105	97	106	78	87	67	92	66
UF5	22	8	140	2	17	75	136	131	41	14	33	18	19	12
UF6	1	26	134	3	15	5	43	16	9	13	10	11	6	7
UF7	54	90	61	108	60	64	79	116	82	114	77	89	99	57
UF8	29	128	127	44	37	31	38	42	126	96	137	118	112	39
UF9	27	28	80	40	32	20	25	35	129	83	132	45	117	30
UF10	110	49	130	4	68	23	139	138	46	24	34	36	133	21
AVG	52.4	64.6	99.1	48.1	50.7	51.3	83.7	83.9	83.8	72.7	84.1	72.7	94.4	**45.6**

Table 3.9: P-value and Friedman aligned ranks obtained by all algorithms

Algorithms	**Average values of Friedman Aligned Ranks overall problem instances**
MOEA/D	52.4
GDE3	64.6
MOEADGM	99.1
MTS	48.1
LiuLiAlgorithm	50.7
DMOEADD	51.3
NSGAIILS	83.7
OWMOSaDE	83.9
ClusteringMOEA	83.8
AMGA	72.7
MOEP	84.1
DECMOSA-SQP	72.7
OMOEAII	94.4
DMN	**45.6**
$\mathbf{F_{AR}}$	**65.985**
p-value	**0.001**

To represent the speed of convergence for the proposed method, the graph in Figure 3.5 is prepared based on the IGD values for UF5. The graph indicates that proposed method is remarkably faster than its competitors.

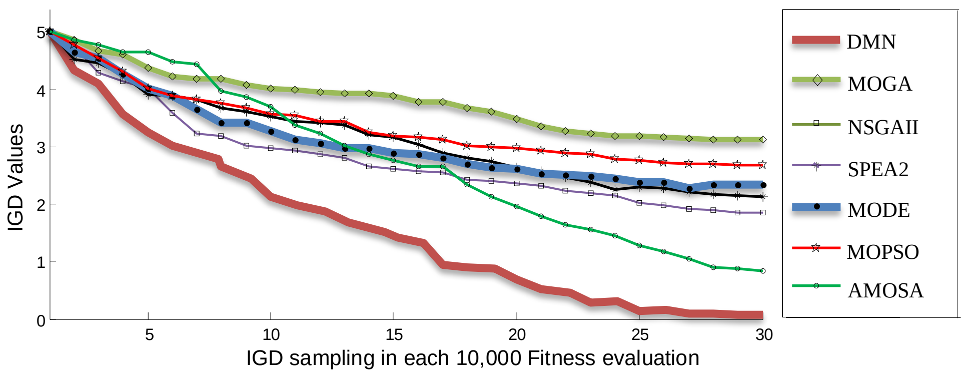


Figure 3.5: Convergence speed plots of DMN and its components agents for UF5

3.6.1 Experimental Results Over ZDT, DTLZ and WFG Test Problems

There exist six test problems in DTLZ benchmark set. The number of objectives and number of variables for all these problems is adjustable. The PF for DTLZ2, DTLZ3 and DTLZ4 is in the form of spherical, while the PF of DTLZ1 is planar.

Meanwhile since the PF for DTLZ7 is not continuous, it can be used to test if an algorithm is able to extract solutions in those undiscovered parts.

Also, there are five two-objective benchmarks in ZDT problem set called by ZDT1 to ZDT5. Among these problems, the PF of ZDT2 and ZDT6 are in the form of concave and PF of ZDT1 and ZDT4 are from convex type [56]. Likewise, there are 9 benchmarks in WFG set in which their number of variables and objectives are adjustable as well. The WFG problems are more challenging since they consist of unbiased and biased parameter types.

The evaluation of proposed method (DMN) over WFG, DTLZ and ZDT problem sets is given in the following. According to the given tables, it can be seen that the PF's extracted by DMN match the optimal PF's.

Table 3.10 illustrates the hypervolume, standard deviation and average IGD values for WFG, DTLZ and ZDT problem sets. These values are measured through the thirty different executions of the method. According to the table, it can be proved that DMN is effective because it has low standard deviation and IGD values. It means that the PF extracted by DMN is close to the optimal one. Likewise, the DMN has a small hypervolume indicating that DMN in a robust method.

The DMN is also evaluated against new methods to prove its effectiveness. The evaluation results related to DMN and five recent algorithms are given in Table 3.11, 3.12 and 3.13 in which the results of those five algorithms are taken from the literature [39, 47, 49, 57 and 59]. It should be mentioned that the same experimental circumstances are applied over all methods.

Table 3.10: Hypervolume and IGD values obtained by DMN for WFG, DTLZ and ZDT in 30 different executions

Function	IGD Score	Hypervolume Score	Function	IGD Score	Hypervolume Score
ZDT1	0.0031±0.0001	0.923±0.0027	**WFG1**	0.1951±0.0871	5.920±0.3243
ZDT2	0.0032±0.0002	0.902±0.0041	**WFG2**	0.0134±0.0060	6.104±0.0100
ZDT3	0.0037±0.0002	0.953±0.0433	**WFG3**	0.0106±0.0043	5.621±0.0052
ZDT4	0.0032±0.0002	0.861±0.1508	**WFG4**	0.0140±0.0072	3.359±0.0047
ZDT6	0.0026±0.0001	0.682±0.0003	**WFG5**	0.0424±0.0158	3.039±0.1201
DTLZ1	0.0014±0.0001	0.785±0.0446	**WFG6**	0.0783±0.0122	2.960±0.0656
DTLZ2	0.0027±0.0002	0.435±0.0175	**WFG7**	0.0090±0.0019	3.407±0.0022
DTLZ3	0.0039±0.0002	0.420±0.0017	**WFG8**	0.1127±0.0210	2.780±0.0328
DTLZ4	0.0028±0.0001	0.368±0.0083	**WFG9**	0.0230±0.0063	3.195±0.0653
DTLZ7	0.0049±0.0005	0.481±0.3280			

Table 3.11: Average IGD values obtained by DMN and state-of-the-art methods for bi-objectives ZDT problems.

Function	BLX-NU	SBX-PN	MOEA/D	MOEA/D-DE+PSO	MOEA/D-CPDE	DMN	DMN Rank Out of 7
ZDT1	0.00558	0.02990	0.00405	0.00414	0.00403	**0.00319**	**1**
ZDT2	0.00437	0.00744	0.00381	0.00387	0.00380	**0.00329**	**1**
ZDT3	**0.00310**	0.00592	0.00708	0.00902	0.00708	0.00395	2
ZDT4	**0.00079**	0.00215	0.19600	0.00755	0.00395	0.00323	3
ZDT6	**0.00107**	0.00122	0.01340	0.01450	0.00597	0.00260	3

Table 3.12: Average IGD values obtained by DMN and state-of-the-art methods for bi-objectives DTLZ problems.

Function	BLX-NU	SBX-PN	DMN	DMN Rank Out of 3
DTLZ1	**0.00027**	0.00055	0.00142	3
DTLZ2	**0.00076**	0.00268	0.00275	3
DTLZ3	0.00040	**0.00039**	0.00391	3
DTLZ4	0.00293	**0.00274**	0.00283	2
DTLZ7	0.00696	0.01170	**0.00496**	**1**

The experimental results indicate that the proposed method is effective and robust. As it can be seen from the tables, DMN is the best performing method over the 2 out of 5 ZDT test problems. Meanwhile, DMN takes the second rank for other 3 test problems. The best performing algorithm for these problems is BLX-NU. Based on the results obtained for DTLZ benchmarks, it can be observed that DMN takes the third rank after BLX-NU and SBX-PX methods. Also it takes the 2nd and 1st ranks for DTLZ4 and DTLZ7. It should be mentioned that IGD values obtained by DMN for DTLZ1 are much close to the values obtained by BLX-NU and SBX-PX.

The proposed DMN method is also evaluated over WFG test set. In 3 out 9 benchmarks, DMN takes the first place. In the other hand, the worst position of DMN is 4 among 14 methods. Even though the WFG test set includes hard problems, the proposed DMN performs better than the most of methods e.g. MOEADD, NSGAII, SPEA2-SDE and MOEAD. Taking the hypervolume into account, Hype method outperforms all methods under the consideration and the proposed method is the second best performing method for nine WFG test problems. Hype method tries to maximize the hypervolume value.

Table 3.13: Hypervolume and average IGD values obtained by DMN and state-of-the-art methods for bi-objectives WFG problems.

Metrics	Problem	MSOPSII	MOEAD	HypE	PICEA-g	SPEA2 SDE	GrEA	NSGAII	KnEA	RVEA	Two-Arch2	Θ-DEA	MOEADD	AnD	DMN
IGD Scores	WFG1	0.211 ±0.091	0.527 ±0.059	0.726 ±0.153	0.203 ±0.036	0.192 ±0.048	**0.191 ±0.090**	0.270 ±0.050	0.287 ±0.139	0.581 ±0.047	0.257 ±0.090	0.272 ±0.082	0.311 ±0.036	0.284 ±0.026	0.195 ±0.087
	WFG2	0.026 ±0.003	0.109 ±0.068	**0.010 ±0.000**	0.025 ±0.048	0.012 ±0.000	0.031 ±0.002	0.015 ±0.000	0.918 ±0.270	0.077 ±0.010	0.012 ±0.001	0.030 ±0.047	0.024 ±0.001	0.030 ±0.003	0.013 ±0.006
	WFG3	0.016 ±0.000	0.026 ±0.004	0.012 ±0.000	0.018 ±0.001	0.013 ±0.000	0.023 ±0.000	0.013 ±0.000	0.017 ±0.000	0.057 ±0.010	0.014 ±0.001	0.012 ±0.000	0.015 ±0.001	0.019 ±0.001	**0.010 ±0.004**
	WFG4	0.018 ±0.001	0.036 ±0.004	0.017 ±0.001	0.018 ±0.001	0.031 ±0.005	0.026 ±0.001	**0.013 ±0.001**	0.025 ±0.004	0.094 ±0.015	0.016 ±0.000	0.014 ±0.001	0.014 ±0.000	0.019 ±0.002	0.014 ±0.007
	WFG5	0.066 ±0.000	0.073 ±0.001	0.066 ±0.001	0.065 ±0.002	0.078 ±0.004	0.074 ±0.002	0.064 ±0.001	0.078 ±0.009	0.101 ±0.014	0.065 ±0.002	0.065 ±0.002	0.067 ±0.002	0.066 ±0.001	**0.042 ±0.015**
	WFG6	0.076 ±0.021	0.095 ±0.023	0.081 ±0.020	0.095 ±0.019	0.094 ±0.018	0.080 ±0.025	0.086 ±0.024	0.308 ±0.069	0.169 ±0.024	**0.073 ±0.020**	0.080 ±0.014	0.083 ±0.018	0.084 ±0.018	0.078 ±0.012
	WFG7	0.019 ±0.001	0.033 ±0.003	0.017 ±0.000	0.016 ±0.000	0.045 ±0.005	0.029 ±0.001	0.012 ±0.000	0.140 ±0.054	0.066 ±0.012	0.016 ±0.000	0.012 ±0.000	0.013 ±0.000	0.019 ±0.001	**0.009 ±0.001**
	WFG8	0.112 ±0.002	0.126 ±0.005	0.111 ±0.003	0.120 ±0.004	0.117 ±0.003	0.121 ±0.000	0.112 ±0.001	0.501 ±0.072	0.206 ±0.015	0.117 ±0.008	0.114 ±0.003	**0.110 ±0.022**	0.118 ±0.002	0.112 ±0.021
	WFG9	0.047 ±0.072	0.074 ±0.054	**0.020 ±0.001**	0.042 ±0.050	0.035 ±0.005	0.030 ±0.002	0.023 ±0.001	0.037 ±0.041	0.060 ±0.005	**0.020 ±0.002**	0.021 ±0.002	0.022 ±0.002	0.028 ±0.003	0.023 ±0.006
Hypervolume Scores	WFG1	5.726 ±0.519	4.057 ±0.272	4.528 ±0.596	5.666 ±0.174	5.971 ±0.307	**6.011 ±0.433**	5.587 ±0.176	5.785 ±0.406	3.827 ±0.202	5.528 ±0.516	5.512 ±0.376	5.228 ±0.194	5.293 ±0.174	5.920 ±0.324
	WFG2	6.100 ±0.008	5.877 ±0.143	**6.129 ±0.004**	6.094 ±0.048	6.121 ±0.006	6.100 ±0.004	6.110 ±0.006	4.632 ±0.553	5.835 ±0.056	6.104 ±0.016	6.092 ±0.047	6.090 ±0.012	6.079 ±0.014	6.104 ±0.010
	WFG3	5.606 ±0.007	5.526 ±0.030	**5.623 ±0.004**	5.577 ±0.010	5.616 ±0.002	5.584 ±0.004	5.610 ±0.007	5.602 ±0.009	5.369 ±0.056	5.605 ±0.012	5.615 ±0.005	5.598 ±0.011	5.589 ±0.011	5.621 ±0.005
	WFG4	3.341 ±0.003	3.232 ±0.027	**3.363 ±0.001**	3.322 ±0.007	3.352 ±0.002	3.328 ±0.002	3.345 ±0.004	3.300 ±0.022	3.051 ±0.046	3.357 ±0.003	3.346 ±0.003	3.339 ±0.007	3.341 ±0.003	3.359 ±0.004
	WFG5	3.018 ±0.004	2.961 ±0.006	3.033 ±0.013	3.013 ±0.020	3.012 ±0.020	2.984 ±0.020	3.022 ±0.012	2.940 ±0.050	2.883 ±0.035	3.023 ±0.021	3.015 ±0.019	2.994 ±0.022	3.020 ±0.014	**3.039 ±0.12**
	WFG6	2.962 ±0.118	2.882 ±0.121	2.959 ±0.112	2.871 ±0.105	2.938 ±0.100	2.958 ±0.150	2.912 ±0.129	2.067 ±0.233	2.603 ±0.108	**2.990 ±0.111**	2.940 ±0.076	2.917 ±0.102	2.940 ±0.099	2.960 ±0.065
	WFG7	3.339 ±0.002	3.230 ±0.016	3.366 ±0.000	3.332 ±0.003	3.356 ±0.001	3.329 ±0.001	3.353 ±0.001	2.865 ±0.184	3.149 ±0.026	3.362 ±0.000	3.353 ±0.001	3.345 ±0.001	3.343 ±0.002	**3.407 ±0.002**
	WFG8	2.776 ±0.013	2.707 ±0.034	**2.797 ±0.010**	2.748 ±0.011	2.786 ±0.008	2.775 ±0.004	2.776 ±0.008	1.415 ±0.121	2.364 ±0.056	2.789 ±0.020	2.770 ±0.012	2.778 ±0.009	2.759 ±0.009	2.780 ±0.032
	WFG9	3.156 ±0. 393	2.986 ±0. 272	**3.315 ±0.010**	3.149 ±0.266	3.292 ±0.017	3.266 ±0.019	3.264 ±0.016	3.190 ±0.235	3.110 ±0.029	3.301 ±0.020	3.267 ±0.020	3.264 ±0.017	3.256 ±0.018	3.195 ±0.065

The Friedman aligned ranks test is also carried out over the results obtained for WFG benchmark set. Tables 3.14 and 3.15 illustrate that the proposed method is the best performing algorithm. Also, the rank for DMN is 3 in terms of hypervolume values.

Table 3.14: Friedman aligned rank values for WFG problems.

Metrics	Function	MSOPSII	MOEAD	HypE	PICEA-g	SPEA2 SDE	GrEA	NSGAII	KnEA	RVEA	Two-Arch2	Θ-DEA	MOEADD	AnD	DMN
IGD	WFG1	5	122	125	4	2	1	18	27	123	16	19	65	22	3
	WFG2	13	111	6	12	8	17	10	126	48	8	15	11	15	9
	WFG3	89	104	74	93	78	101	78	92	115	80	74	85	95	69
	WFG4	72	108	70	72	103	97	58	94	119	68	64	64	76	64
	WFG5	88	100	88	84	107	102	79	107	113	84	84	91	88	36
	WFG6	32	67	39	67	59	38	46	121	118	28	38	42	43	35
	WFG7	56	98	51	50	110	90	45	120	114	50	45	47	56	41
	WFG8	25	40	21	33	30	34	25	124	117	30	26	20	31	25
	WFG9	109	116	53	105	96	81	61	99	112	53	54	57	75	61
	AVG	54.3	96.2	58.5	57.7	65.8	62.3	46.6	101.1	108.7	46.3	46.5	53.5	55.6	**38.1**
Hypervolume	WFG1	122	3	5	121	125	126	120	123	1	119	118	17	25	124
	WFG2	108	18	117	104	116	108	114	2	14	112	100	99	97	112
	WFG3	46	24	64	31	58	33	52	41	12	45	56	39	34	62
	WFG4	51	19	73	35	61	37	54	30	9	67	55	49	51	68
	WFG5	42	26	59	38	36	29	47	23	15	48	40	32	44	65
	WFG6	95	53	92	43	86	91	79	6	10	109	88	82	88	93
	WFG7	66	21	83	60	78	57	77	7	13	82	77	72	69	94
	WFG8	103	80	115	90	110	101	103	4	8	113	98	105	96	106
	WFG9	22	11	89	20	84	74	71	27	16	85	75	71	63	28
	AVG	72.7	28.3	77.4	60.2	**83.7**	72.8	79.6	29.2	10.8	**86.6**	78.5	62.8	63.0	**83.5**

Table 3.15: Friedman aligned rank values for bi-objective WFG problems obtained by all methods

IGD		**Hypervolume**	
Algorithms	Average values of Friedman Aligned Ranks overall problem instances	Algorithms	Average values of Friedman Aligned Ranks overall problem instances
MSOPSII	54.31	MSOPSII	72.7
MOEAD	96.22	MOEAD	28.3
HypE	58.55	HypE	77.4
PICEA-g	57.77	PICEA-g	60.2
SPEA2 SDE	65.88	SPEA2 SDE	**83.7**
GrEA	62.33	GrEA	72.8
NSGAII	46.66	NSGAII	79.6
KnEA	101.11	KnEA	29.2
RVEA	108.77	RVEA	10.8
Two-Arch2	46.33	Two-Arch2	**86.6**
Θ-DEA	46.55	Θ-DEA	78.5
MOEADD	53.55	MOEADD	62.8
AnD	55.66	AnD	63
DMN	**38.11**	**DMN**	**83.5**
F_{AR}	**62.327604**	**F_{AR}**	**62.432964**
p-value	**0.00000002**	**p-value**	**0.000000019**

WFG and DTLZ test problems can be adjusted in terms of objectives and decision variables. The proposed method is also evaluated using 3-objective type of these benchmarks and obtained results are compared to state-of-the-art methods. Table 3.16 demonstrates the hypervolume and IGD values for this evaluation. It can be observed from the table that proposed method is effective and robust.

Table 3.16: Average IGD values obtained by DMN and state-of-the-art methods for three-objectives WFG and DTLZ problems.

Function	NSGA-III-WA	NSGA-III	VAEA	RVEA	MOEA/D	MOEA/D-M2M	DMN	DMN Rank out of 7
DTLZ1	0.0314 ±0.0006	**0.0209 ±0.0006**	0.0777 ±0.0008	0.0620 ±0.0027	0.0408 ±0.0071	0.0431 ±0.0055	0.0400 ±0.0009	3
DTLZ2	0.0547 ±0.0002	**0.0545 ±0.0004**	0.0563 ±0.0004	0.0549 ±0.0001	0.0639 ±0.0007	0.0941 ±0.002	0.0546 ±0.0006	2
DTLZ3	0.0589 ±0.0007	0.0993 ±0.0008	**0.0559 ±0.0019**	0.0660 ±0.0044	0.0638 ±0.0014	0.0949 ±0.001	0.0621 ±0.0028	3
DTLZ4	**0.0029 ±0.0001**	0.0036 ±0.0007	0.0553 ±0.1937	0.0033 ±0.0002	0.0643 ±0.1009	0.0793 ±0.0316	0.0502 ±0.0078	4
DTLZ5	0.1281 ±0.0158	**0.1143 ±0.0056**	0.1674 ±0.0570	0.2057 ±0.0032	0.4196 ±0.0023	0.0432 ±0.0088	0.1984 ±0.0064	5
DTLZ6	**0.9766 ±0.0252**	1.516 ±0.0912	1.656 ±0.0509	1.303 ±0.0202	1.515 ±0.0075	1.826 ±0.0036	1.4830 ±0.0032	3

Table 3.17: Average IGD values obtained by DMN and state-of-the-art methods for three-objectives DTLZ problems.

Function	Average IGD
DTLZ1	0.0400±0.000
DTLZ2	0.0546±0.000
DTLZ3	0.0621±0.002
DTLZ4	0.0502±0.007
DTLZ5	0.1984±0.006
DTLZ6	1.4830±0.003
WFG1	0.331 ±0.023
WFG2	0.123 ±0.037
WFG3	0.088 ±0.026
WFG4	0.225 ±0.012
WFG5	0.230 ±0.010
WFG6	0.265 ±0.019
WFG7	0.220 ±0.014
WFG8	0.300 ±0.011
WFG9	0.229 ±0.009

Meanwhile, the hypervolume and IGD values obtained over 3-objective WFG and DTLZ benchmarks are represented in Table 3.17 and Table 3.18. The values in the table are taken from [61]. It can be seen from the table that proposed method takes 4th place among the all methods under consideration. The rank of proposed method for DTLZ1, DTLZ3 and DTLZ is 3 and for DTLZ2 is 2. Likewise, it is clear that the proposed method works better than the best algorithm of CEC2009 (MOEA/D) in all DTLZ test problems. If we take the hypervolume into account, the rank of suggested method among all methods is 3. Also the proposed system outperforms the other 4 methods in majority of DTLZ benchmarks.

The evaluation process is continued by taking the WFG benchmarks into consideration. The results obtained by DMN are compared to the results reported in [60]. According to the IGD values, the proposed method outperforms all competitors in WFG2 and it is the 2^{nd} best method for WFG5 and WFG7. Also, the proposed method works better than most of the methods in most of the benchmarks. The lowest rank of the proposed method among fourteen algorithms is eight for WFG8.

As it can be seen from the Table 3.18, even though the best method in terms of hypervolume is Hype, the proposed method obtains better hypervolume values rather than most of the other algorithms.

The Friedman Aligned Ranks test for the proposed method and all other methods is carried out over the obtained results and is represented in Table 3.19 and 3.20. The test is done to show the similarity of all algorithms taken into account and also to find the rank for the proposed method.

Table 3.18: Average IGD values obtained by DMN and state-of-the-art methods for three-objectives WFG problems.

Problem	WFG1	WFG2	WFG3	WFG4	WFG5	WFG6	WFG7	WFG8	WFG9
MSOPSII	0.3861 ±0.071	0.272 ±0.034	0.097 ±0.024	0.260 ±0.009	0.280 ±0.009	0.320 ±0.017	0.271 ±0.013	0.391 ±0.012	0.256 ±0.030
MOEAD	0.652 ±0.095	1.022 ±0.033	0.204 ±0.057	0.263 ±0.005	0.254 ±0.003	0.299 ±0.008	0.373 ±0.045	0.325 ±0.011	0.303 ±0.037
HypE	1.330 ±0.122	0.271 ±0.043	**0.037 ±0.003**	0.333 ±0.014	0.362 ±0.011	0.372 ±0.022	0.383 ±0.014	0.371 ±0.014	0.362 ±0.013
PICEA-g	0.978 ±0.107	0.153 ±0.009	0.125 ±0.010	0.223 ±0.003	**0.227 ±0.003**	0.263 ±0.021	**0.218 ±0.003**	0.309 ±0.004	**0.221 ±0.011**
SPEA2 SDE	**0.294 ±0.051**	0.247 ±0.055	0.066 ±0.005	0.328 ±0.013	0.334 ±0.016	0.355 ±0.019	0.327 ±0.014	0.360 ±0.011	0.311 ±0.013
GrEA	0.304 ±0.044	0.260 ±0.026	0.091 ±0.008	0.241 ±0.002	0.260 ±0.004	0.272 ±0.009	0.255 ±0.009	0.302 ±0.008	0.239 ±0.005
NSGAII	0.555 ±0.077	0.182 ±0.005	0.119 ±0.008	0.222 ±0.000	0.231 ±0.000	0.251 ±0.012	0.222 ±0.000	0.295 ±0.005	0.235 ±0.031
KnEA	0.379 ±0.053	0.236 ±0.043	0.136 ±0.057	0.254 ±0.010	0.268 ±0.015	0.302 ±0.014	0.252 ±0.013	0.338 ±0.012	0.229 ±0.005
RVEA	0.654 ±0.065	0.217 ±0.020	0.230 ±0.019	0.243 ±0.005	0.237 ±0.002	0.272 ±0.017	0.239 ±0.005	0.328 ±0.012	0.236 ±0.006
Two-Arch2	0.458 ±0.114	0.153 ±0.003	0.087 ±0.006	0.227 ±0.005	0.237 ±0.003	0.253 ±0.013	0.224 ±0.004	0.311 ±0.005	**0.221 ±0.003**
ϴ-DEA	0.475 ±0.065	0.210 ±0.022	0.134 ±0.018	**0.221 ±0.000**	0.230 ±0.000	**0.246 ±0.011**	0.222 ±0.000	**0.293 ±0.004**	0.232 ±0.030
MOEADD	1.023 ±0.149	0.485 ±0.011	0.260 ±0.010	0.241 ±0.000	0.245 ±0.001	0.261 ±0.012	0.244 ±0.001	0.305 ±0.003	0.239 ±0.001
AnD	0.479 ±0.049	0.249 ±0.022	0.154 ±0.018	0.228 ±0.006	0.238 ±0.004	0.255 ±0.016	0.229 ±0.004	0.328 ±0.009	0.226 ±0.009
DMN	0.331 ±0.023	**0.123 ±0.037**	0.088 ±0.026	0.225 ±0.012	0.230 ±0.010	0.265 ±0.019	0.220 ±0.014	0.300 ±0.011	0.229 ±0.009
DMN Rank out of 14	3	1	4	4	2	8	2	3	4

The obtained values demonstrate that the proposed DMN system is an effective method to deal with real-valued multi-objective optimization problems.

It is observed that the proposed method outperforms other algorithms in terms of IGD values and takes the 5th position in terms of hypervolume values.

Table 3.19: Friedman aligned rank values obtained by all methods for three-objective WFG problems.

Metrics	Function	MSOPS II	MOEA D	HypE	PICEA-g	SPEA2 SDE	GrEA	NSGAII	KnEA	RVEA	Two-Arch2	Θ-DEA	MOEA DD	AnD	DMN
IGD	WFG1	5	106	126	123	1	2	29	4	107	9	10	124	11	3
	WFG2	65	125	63	8	19	38	12	17	15	8	14	122	23	6
	WFG3	32	111	13	82	16	26	76	91	116	20	90	121	100	22
	WFG4	94	95	114	46	113	79	44	89	81	54	41	79	56	48
	WFG5	99	83	117	33	112	85	45	93	58	58	43	70	60	43
	WFG6	102	97	115	59	110	75	31	98	75	37	28	53	40	64
	WFG7	92	119	120	18	108	80	25	77	50	27	25	66	30	21
	WFG8	109	84	103	69	101	55	39	96	87	71	34	62	87	49
	WFG9	88	104	118	36	105	73	67	52	68	36	61	73	47	52
	AVG	76.2	102.6	98.7	52.6	76.1	57.0	40.8	68.5	73.0	35.5	38.4	85.5	50.4	**34.2**
Hypervolume	WFG1	123	3	4	2	126	124	33	125	7	120	121	1	118	122
	WFG2	52	8	106	39	86	57	66	67	34	94	84	47	58	99
	WFG3	65	28	101	53	80	75	60	37	21	72	63	19	38	73
	WFG4	27	11	113	12	93	103	79	29	45	92	88	51	61	32
	WFG5	25	16	85	24	76	74	83	30	64	54	78	42	70	81
	WFG6	13	10	116	14	96	98	62	17	49	110	82	55	109	46
	WFG7	26	5	102	23	107	111	91	50	43	105	95	40	90	104
	WFG8	9	20	89	18	114	115	68	22	35	97	69	56	59	108
	WFG9	41	6	119	15	112	117	31	71	36	100	48	44	87	77
	AVG	42.3	11.8	92.7	22.2	**98.8**	**97.1**	63.6	49.7	37.1	93.7	80.8	39.4	76.6	82.4

Table 3.20: Friedman aligned rank values obtained by all methods for three-objectives WFG problems.

IGD		**Hypervolume**	
Algorithms	Average values of Friedman Aligned Ranks overall problem instances	Algorithms	Average values of Friedman Aligned Ranks overall problem instances
MSOPSII	76.222	MSOPSII	42.3
MOEAD	102.667	MOEAD	11.8
HypE	98.778	HypE	92.7
PICEA-g	52.667	PICEA-g	22.2
SPEA2 SDE	76.111	SPEA2 SDE	**98.8**
GrEA	57.000	GrEA	**97.1**
NSGAII	40.889	NSGAII	63.6
KnEA	68.556	KnEA	49.7
RVEA	73.000	RVEA	37.1
Two-Arch2	35.556	Two-Arch2	93.7
Θ-DEA	38.444	Θ-DEA	80.8
MOEADD	85.556	MOEADD	39.4
AnD	50.444	AnD	76.6
DMN	**34.222**	**DMN**	82.4
F_{AR}	**48.402574**	**F_{AR}**	**72.812914**
p-value	**0.0000055**	**p-value**	**0.0000000002**

3.6.2 Comparing the Proposed Algorithm with Other State-of-the-art Algorithms

This section compares the results obtained by 8 recent improved versions of the best algorithm in CEC2009, MOEA/D, namely: MOEA/DVA, MOEA/D-DE+PSO, BCE-MOEA/D+TCH, MOEA/D-CPDE, MOEA/D-DRA, MOEA/D-CMX-SPX, ENS-MOEA/D, and MOEA/D-HHsw. The descriptions of these methods are given in [47, 48, 49, 50, 51, 52 and 53].

The comparison of the proposed method to 8 recent methods is indicated in Table 3.21. It can be observed from the table that MOEA/DVA, MOAD/D-HHsw and ENS-MOEA/D are the best in 2, 3 and 4 problems, while the proposed method is the best in 1 benchmark.

The IGD values are also plotted in Figure 3.6 to visually represent the comparison. Figure 3.6 shows that the proposed method either outperforms all other methods or is much close to the best in majority of benchmarks. Small fluctuations on the plot of the suggested method prove the stability of the DMN.

Table 3.21: Average value of IGDs achieved by DMN and eight most recent reported methods for UF1-UF10

Function	MOEA/D-DE+PSO	MOEA/D-CPDE	MOEA/D-HHsw	MOEA/D-DRA	MOEA/D-CMX + SPX	ENS-MOEA/D	BCE-MOEA/D+TCH	MOEA/DVA	DMN
UF1	0.0464	0.0052	**0.0010**	0.0015	0.0042	0.0016	0.0016	0.0041	0.0055
UF2	0.0151	0.0118	**0.0018**	0.0035	0.0056	0.0040	0.0065	0.0041	0.0061
UF3	0.0092	0.0447	0.0031	0.0039	0.0111	**0.0025**	0.0095	0.0227	0.0529
UF4	0.0579	0.0453	0.0529	0.0602	0.0641	0.0420	0.0660	0.0350	**0.0339**
UF5	0.6742	0.2034	0.2737	0.2549	0.4185	0.2481	0.4034	**0.0325**	0.0842
UF6	0.8057	0.0915	0.0963	0.3261	0.3273	**0.0608**	0.4250	0.0569	0.0828
UF7	0.4818	0.0060	**0.0011**	0.0019	0.0062	0.0017	0.0121	0.0037	0.0057
UF8	0.1498	0.1233	0.0310	0.0406	0.0574	**0.0310**	0.0561	0.0577	0.0886
UF9	0.1423	0.0800	0.0444	0.1230	0.0976	**0.0278**	0.1315	0.1233	0.0903
UF10	0.2127	0.4999	0.4672	0.4087	0.4626	0.2117	0.4649	**0.1035**	0.3164

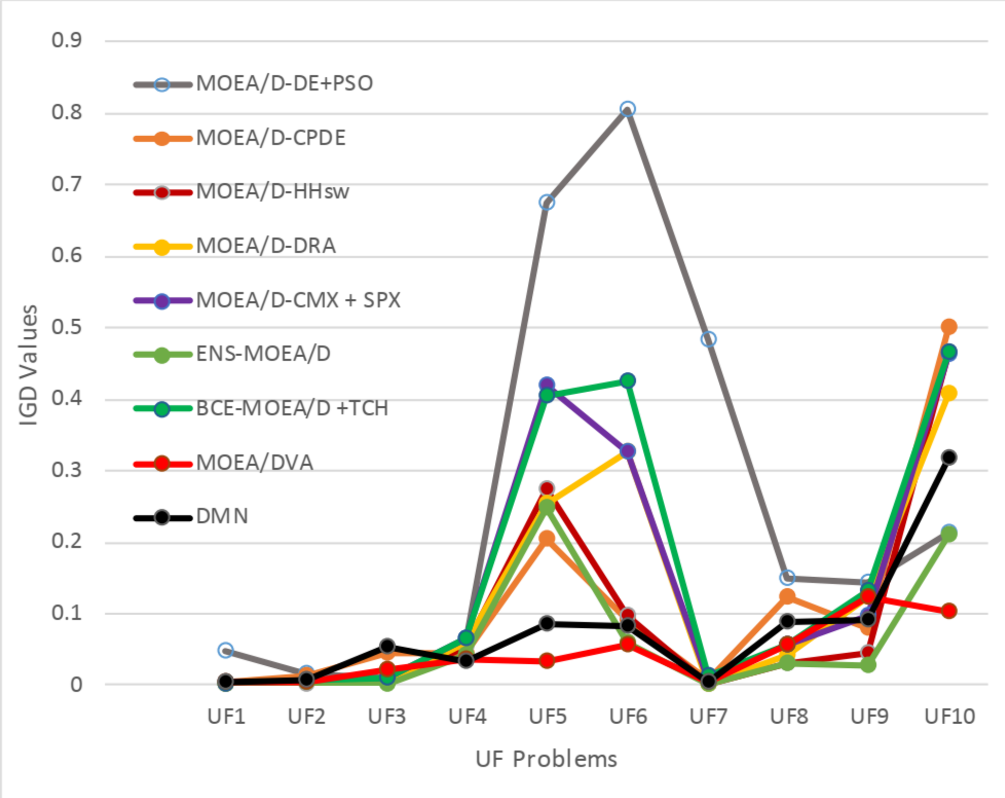


Figure 3.6: IGD values of DMN and eight new MOO methods tested on CEC2009 benchmarks

3.6.3 Evaluation of DMN Over Real-world Benchmarks

This section evaluates the proposed method using 4 well-known real-world benchmarks. The details of the problems are given in [59]. The first evaluation is carried out over the crashworthiness problem. This problem is an unconstrained problem with 5 decision variables and 3 objectives.

The toe-board intrusion, integration of collision and mass of vehicle are the objectives of the problem which are going to be minimized. The aim of the problem is to protect the passengers from accidents.

The second real-world benchmark problem is called as car-side impact problem which has 7 variables and 3 objectives. The average velocity of the V-pillar, public force and vehicle's weight are 3 objectives of the problem to be minimized. The goal of this problem is to measure the safety of passenger from car-side accidents.

The metal removal rate, total life, surface integrity and surface roughness are the objectives for the 3rd real-world problem. Among these objectives, the last objective is going to be minimized and others are supposed to be maximized.

Meanwhile, there exist 3 decision variables and 5 minimization objectives in the 4th real-world problem taken into account. The problem deals with the optimal planning for a storm drainage system. The cost of expected flood damage, the expected economic loss due to flood damage, the storage facility cost, drainage network cost and the treatment facility cost are the list of objectives. Additional information about the problems is given in [59].

The proposed method is evaluated over aforementioned 4 problems and the obtained results are compared to the 5 recent multi-objective optimization methods. According to the description in [59], the size of population is considered as 100 and the algorithms are independently performed 40 times. Likewise the generation size is considered as 1000. Table 3.22 demonstrates the IGD values obtained by the proposed method and other 5 algorithms.

Table 3.22: IGD values of 4 real-world benchmarks for DMN and state-of-the-art methods.

Problem	GDE3	IBEA	NSGAII	SPEA2	MOABHH	DMN
Car Side Impact	0.000788	0.000819	0.001287	0.000753	0.000668	**0.000659**
Crash Worthiness	0.000696	0.002624	0.001263	0.000758	**0.000425**	0.000715
Water R. Planning	0.001486	0.003524	0.002149	0.001904	0.000890	**0.000857**
Machining	0.001690	0.000513	0.001695	0.001752	**0.000505**	0.001103

It can be seen from the represented results that the proposed method and MOAGHH outperform other methods under consideration. MOABHH is a multi-agent system which tries to select the most effective algorithm using the voting system for generating the solutions. It can be seen from [59] that MOABHH outperforms other algorithms to solve aforementioned real-world benchmarks but in water resource planning and car-side impact and problems the proposed method works more effective than MOABHH. Likewise the suggested system outperforms 4 methods in solving machining and crash worthiness problems

3.6.4 DMN Scalability

Regarding DMN scalability, changing the number of agents from 7 to 5, 6, 8, 9, 10, or 11 doesn't affect the results but changing the number of metaheuristic agents in the network to less than 4 , or more than 11 reduces the performance. The reason of this decrease is that there exists 10 metaheuristics in total and for more agents the same metaheuristics are assigned. Table 3.23 illustrates the IGD scores obtained by DMN over four randomly selected benchmarks.

Table 3.23: DMN scalability test

	Original DMN IGD score	**Modified DMN IGD score**				
Agents number / Test case	7 agents	Averge IGD score of 5, 6, 8, 9, 10 and 11 agents	4 agents	12 agents	13 agents	14 agents
UF2	0.00613	≈ 0.0072	0.008	0.008	0.009	0.012
UF5	0.08424	≈ 0.089	0.094	0.093	0.100	0.102
UF7	0.00568	≈ 0.0067	0.008	0.009	0.012	0.015
UF9	0.09030	≈ 0.1023	0.202	0.144	0.188	0.197

Chapter 4

CONCLUSIONS AND FUTURE WORKS

Two new cooperative multi-agent systems are illustrated in this study for solving real-parameter MOO problems.

First proposed method (M^3D/MAS) presents a novel cooperative MAS containing several multi-objective metaheuristic agents to deal with real-valued multi-objective optimization problems. The proposed MAS is tested using CEC2009 unconstrained MOO test instances. The experimental results prove the effectiveness of M^3D/MAS and demonstrates that M^3D/MAS outperforms most of the methods participated in CEC2009 competition.

Performance of M^3D/MAS is also assessed with eight recently enhanced types of MOEA/D. The evaluation outcomes illustrate that the proposed method is very successful and its IGD scores are so close to that of best algorithms.

Three extra experiments are applied and displayed due to presenting the stability and power of the newly suggested MAS architecture, when one new MOO metaheuristic is added to its architecture or replaced with one in the architecture. Results obtained from these experiments clearly displayed robustness of the M^3D/MAS under mentioned circumstances too.

Strength of M^3D/MAS is also assessed using ZDT and DTLZ test cases. The obtained results prove the effectiveness of the proposed method and demonstrate that M^3D/MAS performs better than majority of methods participated in CEC2009 contest.

The DMN is another proposed multi-agent system consisting of 7 significant multi-objective optimization algorithms (metaheuristic) which cooperate together to solve the problems. In the proposed system, the nodes of the network represent the metaheuristics and edges indicate the connection between the metaheuristics to transfer the solutions. There exist three layers in the proposed DMN with 3-3-1 structure. Meanwhile in DMN the metaheuristics can change their nodes dynamically when the sessions end up. The solutions are fed up to the metaheuristics in each session in which they are selected from their own sub-populations or from the sub-populations improved by the previous layers. The local archives for all nodes are also updated at the end of the sessions. Likewise, when the final session ends up, all local archives are merged to form the global archive including all non-dominated solutions found so far by all metaheuristics. Thereafter the hypervolume and IGD metrics are calculated for all solutions in the global archive. CEC2009 benchmark set is used to evaluate the proposed method and carry out the experiments. The experimental results are used to compare the suggested method to all state-of-the-art algorithms. The obtained experimental results prove the effectiveness of the proposed DMN and illustrate that DMN outperforms all competitors participated in CEC2009 competition.

The evaluation of DMN is also carried out over well-known WFG, DTLZ and ZDT test sets and the obtain results are compared to majority of robust multi-objective optimization methods. Also, some new and novel methods are taken into consideration to be compared to DMN. It should be noticed that all methods under consideration are tuned and adjusted in the same way and all parameters are initialized with same values. The experimental results show that the suggested method works better than most of the existent algorithms. Likewise, some methods e.g. Hype focus on optimizing some metrics and hence they outperform the proposed method.

The proposed DMN applies a collection of metaheuristics to fulfill the search operation in efficient way. The idea behind the DMN is that several metaheuristics can process more promising parts of the search space and extract more solutions in different parts of the space. This way the capabilities of a heuristic can cover the inabilities of others. The effectiveness of DMN is proven by the numerical results calculated for IGD and hypervolume metrics. Moreover the results obtained by DMN are compared to 8 new multi-objective optimization algorithms and it is seen that DMN is as robust as the most effective methods.

To conduct the evaluation process, some real-world problems are taken into account as well. The difference between these problems and previous ones is regarded to the number of objectives and variables in which their objectives are more and variables are less. It is reported that MOABHH is the most effective method for solving the real-world problems taken into account and according to the experimental results the performance of the proposed method is similar to MOABHH.

As it is shown in Table 4.1, Figure 4.1 and Figure 4.2, both of the proposed methods converge much faster than their components in all different number of fitness evaluations.

Table 4.1: Convergence table of proposed methods and their components

Methods \ Fitness Evaluation	50000	100000	150000	200000	250000	300000
MOGA	4.4	4.1	4.0	3.7	3.5	3.4
NSGAII	4.0	3.6	3.4	2.9	2.4	2.1
SPEA2	4.0	3	2.8	2.4	2.2	1.9
MODE	4.1	3.4	3.2	2.8	2.5	2.3
MOPSO	4.2	3.7	3.5	3.3	3.1	2.9
AMOSA	4.7	3.9	3.1	2	1.4	0.9
MOABC	4.3	3.3	3	2.8	2.7	2.4
M^{3}D/MAS	3.2	2.3	1.5	0.8	0.2	**0.0877**
DMN	3.7	2.8	1.9	1.0	0.2	**0.0842**

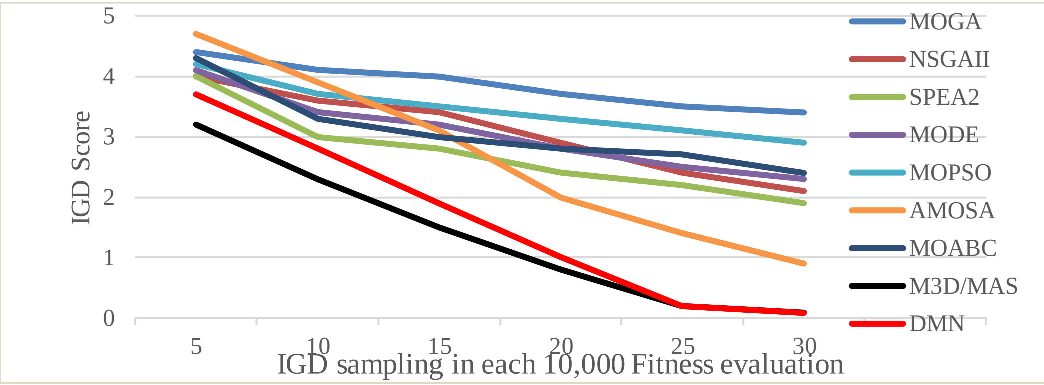


Figure 4.1: Convergence line chart of proposed methods and their components

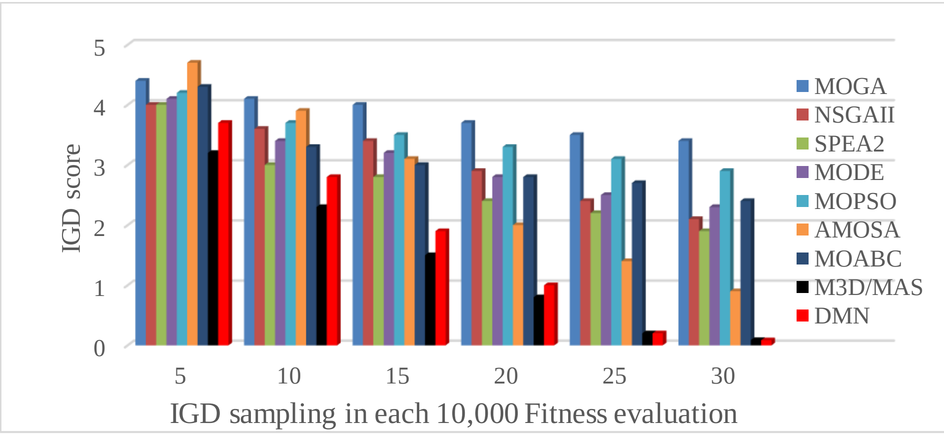


Figure 4.2: Convergence bar chart of proposed methods and their components

By applying the following formula in Equation 6, the convergence percentage of the proposed methods in comparison to their components is caclulated for each fitness evaluation. The results are shown in Table 4.2.

$$\text{convergence\_percentage}_{p,c,f} = (\text{IGD}_{p,f} - \text{IGD}_{c,f})/\text{IGD}_{c,f} * 100 \quad (6)$$

Where:

$\text{convergence\_percentage}_{p,c,f}$ is the convergence speed of proposed method 'p' in comparison to component 'c' with fitness evaluation of 'f'.

Also, $\text{IGD}_{p,f}$ is the IGD score of the proposed method when using 'f' amount of fitness evaluation.

Likewise, $\text{IGD}_{c,f}$ is the IGD score of component 'c' when using 'f' amount of fitness evaluation.

Table 4.2: Convergence table of proposed methods and their components

Methods \ Fitness Evaluation		50000	100000	150000	200000	250000	300000
MOGA	**M^3D/MAS**	-27.27	-43.90	-62.50	-78.37	-94.28	-97.42
	DMN	-15.90	-31.70	-52.50	-72.97	-94.28	-97.52
NSGAII	**M^3D/MAS**	-20.0	-36.11	-55.88	-72.41	-91.66	-95.82
	DMN	-7.50	-22.22	-44.11	-65.51	-91.66	-95.98
SPEA2	**M^3D/MAS**	-20.00	-23.33	-46.42	-66.66	-90.90	-95.38
	DMN	-7.50	-6.66	-32.14	-58.33	-90.90	-95.56
MODE	**M^3D/MAS**	-21.95	-32.35	-53.12	-71.42	-92.00	-96.18
	DMN	-9.75	-17.64	-40.62	-64.28	-92.00	-96.33
MOPSO	**M^3D/MAS**	-23.80	-37.83	-57.14	-75.75	-93.54	-96.97
	DMN	-11.90	-24.32	-45.71	-69.69	-93.54	-97.09
AMOSA	**M^3D/MAS**	-31.91	-41.02	-51.61	-60.00	-85.71	-90.25
	DMN	-21.27	-28.20	-38.70	-50.00	-85.71	-90.64
MOABC	**M^3D/MAS**	-25.58	-30.30	-50.00	-71.42	-92.59	-96.34
	DMN	-13.95	-15.15	-36.66	-64.28	-92.59	-96.49

It can be seen that M^3D/MAS converges 27.27% more/faster than MOGA when the fitness evaluation is 5 (50,000) and DMN converges 15.90% more/faster than MOGA with the same amount of fitness evaluations and so on. The negative numbers in the table show that the convergence speed of the component under consideration is less than the proposed method.

Bellow in Table 4.3 are shown CPU times and runtime complexities of both proposed methods and their components. Based on this table, MOABC, DMN and M^3D/MAS are the most time consumed ones respectively.

Table 4.3: CPU times and runtime complexities of proposed methods and their components

Speed / Methods	CPU Time in second	Runtime complexity
MOGA	370.22	$O(N^2)$
NSGAII	363.03	$O(N^2)$
SPEA2	458.32	$O(N^2 \text{ Log } N)$
MODE	390.09	$O(N^2)$
MOPSO	480.85	$O(N^2 \text{ Log } N)$
AMOSA	636.67	$O(N^2 \text{ Log } N^{\sqrt{2}})$
MOABC	922.70	$O(N^2 \text{ Log } N^2)$
M^3D/MAS	723.65	$O(N^2 \text{ Log } N^{\sqrt{2}})$
DMN	866.39	$O(N^2 \text{ Log } N^2)$

Bellow in Table 4.4 are shown the main differences between two proposed systems in this thesis (structure-wise and performance-wise). According this table M3D/MAS performs better and faster than DMN.

Table 4.4: Comparison between M3D/MAS and DMN in both structure-wise and performance-wise

	M^3D/MAS	**DMN**
Number of Metaheuristic agents	5	7
Metaheuristic agents' relationship	Cooperative and Competitive	Cooperative
Architecture type	Flat	Hierarchy
Number of assessment metrics	3	1
Number of sessions	3 (based on number of metrics)	7 (based on the number of nodes/agents)
Friedman aligned rank on CEC2009	34.51	45.6
Average CPU time in second (in 30 times run)	723.65	866.39
Runtime complexity	$O(N^2 \log N^{\sqrt{2}})$	$O(N^2 \log N^2)$

As a future work for DMN, the tournament-based distribution of population will be implemented in a way that nodes/metaheuristics with better results will get a bigger portion of the population and weaker ones will get a smaller portion. In this way, the robust agents can evolve/improve a bigger portion of the population which may lead to improvement in the performance of the current architecture.
This technique (tournament-based distribution of the population) will be also applied on the population size of the metaheuristic agents in M3D/MAS (it is already used in M3D/MAS for calculating the fitness evaluation number of each agent/metaheuristic in a way that better-performed agents get higher fitness evaluation number for their next session).

Since the proposed architectures fall also in the domain of multiobjective hybrid metaheuristics, therefore the experimental evaluation of the proposed architectures against the state-of-the-art multiobjective hybrid metaheuristics will be taken into the consideration in the feature work.

It is also planned to extend the M^3D/MAS with supplementary multi-objective optimization agents and also expand the DMN with more MOO algorithms and layers and survey their usage for real-world and combinatorial optimization problems. Besides this, these two proposed methods are rather well qualified to be done in a GPU (Graphics Processing Unit) or parallel programming.

REFERENCES

[1] Bosman, P. A. N.: On gradients and hybrid evolutionary algorithms for real-valued multiobjective optimization, *IEEE Trans. on Evolutionary Computation*, Vol. 16, No. 1, pp. 51-69, (2014).

[2] Rizzo A S., Salerno N., Dilettoso E.: A Weakly Pareto Compliant Quality Indicator, Mathematical and Computational Applications. doi:10.3390/mca22010025, (2017).

[3] Bandyopadhyay, S., Saha, S.: Unsupervised classification, similarity mesures, classical and metaheuristic approaches and applications , DOI 10.1007/978-3-642-32451-2_2, *Springer*, Berlin, (2013).

[4] Irina Dumitrescu and Thomas Stutzle.: A survey of methods that combine local search and exact algorithms. *European Journal of Operational Research*, Germany, (2006).

[5] Lalwani , S., Singhal, S., Kumar, R., Gupta, N.: A comprehensive survey: Applications of Multi-Objective Particle Swarm Optimızation algorithm (MOPSO), *Transactions on Combinatorics*, Vol. 2 No. 1, Iran, (2013).

[6] Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P.N.: Multiobjective evolutionary algorithms: a survey of the state-of-the-art, *Swarm and Evolutionary Computation* , Vol. 1, pp. 32-49, (2011).

[7] Deb. K., Agrawal, S., Pratap, A., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Trans. on Evolutionary Computation*, vol. 6, No. 2, pp. 182–197, (2002).

[8] Mousa, A. A. A.: Study on Multiobjective Optimization Using Improved Genetic Algorithm (Methodology and Application), *LAP (Lambert Academic Publishing),* Germany, (2011).

[9] Fonseca, C. M., Fleming, P.J.: Genetic algorithm for multiobjective optimization, formulation, discussion and generalization, Proc. of the Fifth Intl. *Conf. on Genetic Algorithms*, pp. 416-423, (1993).

[10] Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the strength Pareto evolutionary algorithm for multiobjective optimization, *Evolutionary Methods for Design Optimization and Control with Applications to Industrial Problems*, pp. 95–100, (2001).

[11] Xue, F., Sanderson, A. C., Graves, R. J.: Pareto-based multiobjective differential evolution, *IEEE Congress on Evolutionary Computation (CEC'2003)*, pp. 862-869, Australia, (2003)

[12] Bandyopadhyay, S., Saha, S., Maulik, U., Deb, K.: A Simulated Annealing Based Multiobjective Optimization Algorithm: AMOSA : *IEEE Trans. on Evolutionary Computation*, Vol. 12, No. 3, PP. 269-283, (2008).

[13] Coello, C. A., Lechuga, M. S.: MOPSO: a proposal for multiple objective optimization, *Proc. of IEEE Congress on Evolutionary Computation (CEC'2002)*, pp.1051-1056, US, (2002).

[14] Hirano, H., Yoshikawa, T. A: Study on two-step search using global-best in PSO for multi-objective optimization problems. *In Proc. of the 6th International Conference on Soft Computing and Intelligent Systems/13th International Symposium on Advanced Intelligent Systems*, pp. 1894-1897, Japan, (2012).

[15] Karaboga D, Basturk B. Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: International Fuzzy Systems Association World Congress. Berlin, Heidelberg: Springer; 2007.

[16] Hoen, P.J., Tuyls, K., Panait, L., Luke, S., La Poutre, J.A.: An overview of cooperative and competitive multiagent learning, In: Tuyls K., Hoen P.J., Verbeeck K., Sen S. (eds) Learning and Adaption in Multi-Agent Systems. *Lecture Notes in Computer Science*, vol 3898. Springer, Berlin, (2006).

[17] Panait, L., Luke, S.: Cooperative multiagent learning: *the state of the art, Autonomous Agents and Multiagent Systems*, pp. 387-434, (2005).

[18] Zhang, Q., Zhou, A., Zhao, S., Suganthan, P., Liu, W., Santosh Tiwari, S.: *Multiobjective optimization test instances for the CEC 2009 special session and competition, Technical Report CES-487,* The School of Computer Science and Electronic Engineering, University of Essex, (2009).

[19] Stone, P., Veloso, M.: Multiagent systems: a survey from a machine learning perspective: *Autonomous robotics*, Vol. 8, pp. 345-383, (2000).

[20] Stuart, R., Norvig, P.: Artificial Intelligence: A Modern Approach (2nd ed.): *Prentice Hall*, ISBN. 0-13-790395-2, chap. 2, (2003).

[21] Teixeira, F., Castro, A.J.M., Rocha, A.P., Oliveira, E.: Multiagent learning in both cooperative and competitive environments, *XVI Portuguese Conf. on AI – EPAI* 2013, pp. 370-381, (2013).

[22] Sycara, K. P.: Multiagent systems: American association for artificial intelligence, *AI magazine*, vol. 19, no. 2, pp. 79-92, (1998).

[23] Meignan, D., Creput, J. C., Koukam, A.: An organizational view of metaheuristics: Proc. of First Intl. *Workshop on Optimization on Multiagent Systems*, pp. 77-85, (2008).

[24] Stone, P., Veloso, M.: Multiagent systems: a survey from a machine learning perspective: *Autonomous robotics*, Vol. 8, pp. 345-383, (2000).

[25] Malek R.: Collaboration of Metaheuristic Algorithms through a Multi-Agent System. Conference: *4th International Conference on Industrial Applications of Holonic and Multi-Agent Systems, HoloMAS, Springer*, (2009).

[26] Zhang L, Wong TN, Zhang S, Wan SY.: A multi-agent system architecture for integrated process planning and scheduling with meta-heuristics: *Proceedings of*

the 41st International Conference on Computers & Industrial Engineering, USA, (2011).

[27] Lotfi N, Acan A.: A tournament-based competitive-cooperative multiagent architecture for real parameter optimization. *Soft Computing*, DOI 10.1007/s00500-015-1768-4, (2015).

[28] Jiang, S., Zhang, J., Ong, Y. S.: A multiagent evolutionary framework based on trust for multiobjective optimization: *Proc. of the 11th Intl. Conf. on Autonomous Agents and Multiagent Systems*, pp.299-306, (2012).

[29] Acan A., Lotfi N.: A multiagent, dynamic rank-driven multi-deme architecture for real-valued multiobjective optimization, *Artificial Intelligence Review*, DOI: 10.1007/s10462-016-9493-7, (2016).

[30] Uhruski P, Grochowski M, And Schaefer R.: A two-layer agent-based system for large-scale dıstrıbuted computatıon, *Computational Intelligence*, Volume 24, Number 3, (2008).

[31] Mohammadzadeh, H., Soleimanian, F.: A multi-agent system based for solving high-dimensional optimization problems: A case study on email spam detection. International Journal of Communication Systems. 34. 10.1002/dac.4670 (2020).

[32] Shadravan, S., Naji, H., Khatibi, V.: A Distributed Sailfish Optimizer Based on Multi-Agent Systems for Solving Non-Convex and Scalable Optimization

Problems Implemented on GPU. Journal of AI and Data Mining. doi: 10.22044/jadm.2020.9389.2075 (2021).

[33] Icarte, A, G., Herzog, O.: Application of Multiagent System and Tabu Search for Truck Dispatching in Open-pit Mines. In Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART, ISBN 978-989-758-484-8, pages 160-170. DOI: 10.5220/0010391101600170, (2021).

[34] Jiang S, Zhang J, Ong YS, Feng L.: Consistencies and contradictions of performance metrics in multiobjective optimization, *IEEE Trans. on Cybernetics*, Vol. 44 , No. 12, pp. 2391-2404, (2014).

[35] Zitzler E, Thiele L, Laumanns M, Fonseca CM, Fonseca VG.: Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Trans. Evoutionary Computation,* Vol. 7, No. 2, pp. 117–132, (2003).

[36] Zhou A, Jin Y, Zhang Q, Sendhoff B, Tsang E.: Combining model-based and genetics-based offspring generation for multi-objective optimization using a convergence criterion. *In Proc. IEEE Conf. on Evolutionary Computation*, pp. 892–899, (2006).

[37] Veldhuizen DV, Lamont G.: On measuring multiobjective evolutionary algorithm performance. *In Proc. of the Congress on Evolutionary Computation*, Vol. 1, pp. 204-211, (2000).

[38] Zhang, Q., Suganthan, P. N.: Final report on CEC'09 MOEA competition, *Working Report, CES-887, School of Computer Science and Electrical Engineering*, University of Essex, (2008).

[39] Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC09 unconstrained MOP test instances, CEC 2009, *Proc. of the IEEE Eleventh Conf. on Evolutionary Computation*, pp. 203-208, Norway, (2009).

[40] Tseng, L. Y., Chen, C.: Multiple trajectory search for unconstrained/constrained multiobjective optimization, CEC 2009, *Proc. of the IEEE Eleventh Conf. on Evolutionary Computation*, pp. 1951-1958, (2009).

[41] Liu, M., Zou, X., Chen, Y., Wu, Z.: Performance assessment of DMOEA-DD with CEC 2009 MOEA competition test instances, CEC 2009, *Proc. of the IEEE Eleventh Conf. on Evolutionary Computation*, pp. 1951-1958, (2009).

[42] Liu, H., Li, X.: The multiobjective evolutionary algorithm based on determined weight and sub-regional search, CEC 2009, *Proc. of the IEEE Eleventh Conf. on Evolutionary Computation*, pp. 1928-1934, (2009).

[43] Kukkonen, S., Lampinen, J.: Performance assessment of generalized differential evolution 3 with a given set of constrained multiobjective test problems, CEC 2009, *Proc. of the IEEE Eleventh Conf. on Evolutionary Computation*, pp. 2913-2918, (2009).

[44] Dobes, J., Michal, J., Biolkova, V.: Multiobjective optimization for electronic circuit design in time and frequency domains. *Radioengineering*, vol. 22, no. 1, pp. 136-152, (2013).

[45] Zhang, Q., Liu, W., Li, H.: The performance of a new version of MOEA/D on CEC2009 unconstrained MOP test instances, *IEEE Congress on Evolutionary Computation – CEC2009*, pp. 203-208, (2009).

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

[47] Li, W., Wang, L., Jiang, Q., Hei, X., Wang, B.: Multiobjective cloud particle optimization algorithm based on decomposition, Algorithms, Vol. 8, No. 2, pp.157-176, (2015).

[48] Khan, W., Zhang, Q.: MOEA/D-DRA with two crossover operators, *UK Workshop on Computational Intelligence* – UKCI2010, pp. 1-6, (2010).

[49] Mashwani, W., Salhi, A.: Multiobjective memetic algorithm based on decomposition, *Applied Soft Computing*, Vol. 21, pp. 221-243, (2014).

[50] Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Decomposition-based multiobjective evolutionary algorithm with an ensemble of neighborhood size, *IEEE Trans. Evol. Computation*, Vol. 16, No.3, pp. 442-446, (2012).

[51] Goncalves, R.A., Kuk, J.N., Almeida, C.P., Venske, S.M.: Decomposition based multiobjective hyperheuristic with differential evolution, *Intl. Conf. on Computational Collective Intelligence –ICCCI2015*, pp.129-138, (2015).

[52] Li M, Yang S, Liu X.: Pareto or Non-Pareto: Bi-Criterion Evolution in Multiobjective Optimization. *IEEE Transactions on Evolutionary Computation*, 20(5): 645-665, (2016).

[53] Ma X, Liu F, Qi Y, et al.: A multiobjective evolutionary algorithm based on decision variable analyses for multiobjective optimization problems with large-scale variables. *IEEE Transactions on Evolutionary Computation*, 20(2): 275-298, (2016).

[54] Xiang Y., Zhou Y., Liu H.: An elitism based multi-objective artificial bee colony algorithm, *European Journal of Operations Research*, Vol. 245, pp. 168-193, (2015).

[55] Yang K, Mu L, Yang, D, Zou F, Wang L, and Jiang Q, Multiobjective Memetic Estimation of Distribution Algorithm Based on an Incremental Tournament Local Searcher, *Hindawi Publishing Corporation*, The Scientific World Journal, Volume 2014, Article ID 836272, 2014.

[56] Huband S, Hingston P, Barone L, While L, A Review of Multi-objective Test Problems and a Scalable Test Problem Toolkit, *IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION*, VOL. 10, NO. 5,

[57] Ojha M., Singh K.P., Chakraborty P., Verma S., An Aggregation Based Approach with Pareto Ranking in Multiobjective Genetic Algorithm. In: Proceedings of Fifth International Conference on Soft Computing for Problem Solving. *Advances in Intelligent Systems and Computing. Springer*, Singapore, 2016.

[58] Nebro A J, Durillo J J, García-Nieto J, Luna F, SMPSO: A new PSO-based metaheuristic for multi-objective optimization, Conference Paper ·DOI: 10.1109/MCDM.2009.4938830 · Source: *IEEE Xplore*, 2009.

[59] V.R. de Carvalho, J.S. Sichman, Solving real-world multi-objective engineering optimization problems with an Election-Based Hyper-Heuristic, *International Workshop on Optimization in Muiti-agent Systems – OPTMAS18*, Stockholm, Sweden, July 14, 2018 (Paper 7).

[60] Li K, Wang R, Zhang T and Ishibuchi H.: Evolutionary Many-Objective Optimization: A Comparative Study of the State-of-the-Art. *IEEE Access*. 6. 1-1. 10.1109/ACCESS.2018.2832181, (2018).

[61] Wang Y and Sun X.: A Many-Objective Optimization Algorithm Based on Weight Vector Adjustment. *Computational Intelligence and Neuroscience*. Volume 2018, Article ID 4527968, (2018).