Adaptive bio-inspired firefly and invasive weed algorithms for global optimisation with application to engineering problems

Thesis submitted to the University of Sheffield for the degree of Doctor of Philosophy

by

Hyreil Anuar Kasdirin

Department of Automatic Control and Systems Engineering
The University of Sheffield
Mappin Street,
Sheffield, S1 3JD
United Kingdom

December 2016

Abstract

The focus of the research is to investigate and develop enhanced version of swarm intelligence firefly algorithm and ecology-based invasive weed algorithm to solve global optimisation problems and apply to practical engineering problems. The work presents two adaptive variants of firefly algorithm by introducing spread factor mechanism that exploits the fitness intensity during the search process. The spread factor mechanism is proposed to enhance the adaptive parameter terms of the firefly algorithm. The adaptive algorithms are formulated to avoid premature convergence and better optimum solution value. Two new adaptive variants of invasive weed algorithm are also developed seed spread factor mechanism introduced in the dispersal process of the algorithm. The working principles and structure of the adaptive firefly and invasive weed algorithms are described and discussed. Hybrid invasive weed-firefly algorithm and hybrid invasive weed-firefly algorithm with spread factor mechanism are also proposed. The new hybridization algorithms are developed by retaining their individual advantages to help overcome the shortcomings of the original algorithms. The performances of the proposed algorithms are investigated and assessed in single-objective, constrained and multi-objective optimisation problems. Well known benchmark functions as well as current CEC 2006 and CEC 2014 test functions are used in this research. A selection of performance measurement tools is also used to evaluate performances of the algorithms. The algorithms are further tested with practical engineering design problems and in modelling and control of dynamic systems. The systems considered comprise a twin rotor system, a single-link flexible manipulator system and assistive exoskeletons for upper and lower extremities. The performance results are evaluated in comparison to the original firefly and invasive weed algorithms. It is demonstrated that the proposed approaches are superior over the individual algorithms in terms of efficiency, convergence speed and quality of the optimal solution achieved.

Acknowledgement

All praise be to Allah, the Most Merciful and Beneficent. There is no God but Allah and Prophet Muhammad (Peace be upon him) is the Messenger of Allah. With His wills and gracious, I have successfully completed my thesis.

I would like to thank all of the people who have aided and supported me during my doctoral studies. Many people have helped me over the last few years including instructors, supervisors, friends and family. While there are too many of you to mention by name, I do want to mention a few people who have especially had an effect on me. First of all, I wish to express my gratitude to my supervisor Dr. Osman Tokhi and to my second supervisor Dr HL Wei for the support and encouragement during my earlier studies in the department. They have given me the inspiration to step out of the boundaries of my earlier studies and helped me focus on this research.

I would like to express special thanks to Dr. Nafrizuan Yahya, Mr. Ruzaini Hashim, Dr Md Nor Ahmad and other group members for their kindness and guidance through the exciting phases of my doctoral studies. Financial support provided by Ministry of Education, Malaysia and Universiti Teknikal Malaysia Melaka (UTeM) are gratefully acknowledged. I would like to express my deepest gratitude to my family, especially my mothers, Hjh Kamaliah Reduan and Hjh Puteh, and my lovely family for their love, support and understanding during those years being far from home. Finally, I wish to thank my wife, Zuraiha Mohamed for her kindness, love and encouragement. Also, last but not least, my children, Amni Safiya, Muaz Wadi , Asmaa Amani and my precious little son, Amru Hilmi who have always been with me through this wonderful journey.

Contents

List of Figures

List of Tables

Acronyms

ACO	Ant colony optimisation
ARX	Autoregressive exogenous
BFA	Bacterial foraging algorithm
CEC	Congress of Evolutionary Computation
DE	Differential evolution
FA	Firefly algorithm
FA-eSF	Firefly algorithm with exponential spread factor
FA-NSF	Firefly algorithm with nonlinear spread factor
FLC	Fuzzy logic controller
FMS	Flexible manipulator system
GA	General algorithm
HS	Harmony search
HIWFO	Hybrid invasive weed firefly optimisation
HIWFO-SF	Hybrid invasive weed optimisation with spread factor
IWO	Invasive weed optimisation
IWO-eSSF	Invasive weed optimisation with exponential seed spread spread factor
MIWO-eSSF	Modified invasive weed optimisation with exponential seed spread factor
MO	Multi-objective
MSE	Mean-squared error
NFE	Number of function evaluations
PD	Proportional derivation
PSO	Particle swarm optimisation
TRS	Twin rotor system

Chapter 1

Introduction

1.1 Introduction

Optimisation saturates what we do and drives almost every aspect of engineering.

- Dennis Bernstein (Bernstein, 2006)

According to the quote above, everything we do involves optimisation. From a simple application such as our individual personal schedule to more complex applications such as health care, biological, engineering and economic systems; needs to be optimised. Optimisation shows a universal applicability hence making it a very interesting topic to study. The possibility of using biologically-based and evolutionary-based algorithms for optimisation has widely been researched and applied in the past few decades (Simon, 2013). This thesis aims to develop new variants of biologically inspired optimisation algorithm and discusses approaches for solving global optimisation problems.

In this chapter, a brief overview of biologically inspired (bio-inspired) optimisation is introduced. It is followed by brief introduction of the bio-inspired algorithms used, research objectives and methodology. This chapter also presents the research contribution and ends with the organization of the thesis.

1.2 Bio-inspired Optimisation Algorithm

Technologies such as machine learning, high performance computing and other innovative approaches have helped us extensively in solving complex problems in science and engineering. However, the extent of the complexity and diversity of the problems have also urged researchers to look at various ways in solving those problems especially to ensure flexibility, robustness and reliability as well as low computational cost. As a result, researchers tend to go back to look upon the nature or biological point of view on how this biological inspired mechanism could help them solve various complex problems.

In order to tackle this issue, computing inspired by nature, very often referred to biologically inspired computing is developed and explored based on behaviours of living species encountered or by exploiting natural processes observed. By looking at all the creatures, nature has given them biological intelligence of life (Frohlich, 2009). If observed carefully, there are unlimited ways for problem solving provided by nature. Nunes de Castro (2012) mentioned that natural computing is a research field that is aimed at developing new computational techniques, methods and algorithms and tools for solving problems inspired by nature.

One of the important aspects of current bio-inspired computing is optimisation, since people are interested in achieving optimality in solving those problems (Yang, 2010a). Regardless of the complexity and higher dimensional problems as well as computational drawback of existing numerical methods, capability of solving those numerical optimisation problems is still a challenge. Recent biologically-inspired algorithms have been shown to be capable of solving these problems more efficiently. In recent years, the biologically inspired algorithms have been adopted to solve hard optimisation problems and they have shown great potential in solving complex engineering optimisation problems (Yang and He, 2013). Bio-inspired optimisation technique is developed to solve optimisation problems by iteratively improving the problem solution. It is one type of metaheuristics methods and is related to the field of artificial intelligence. This evolution began when John Holland proposed genetic algorithm (GA) in 1975 based on the Charles Darwin's principle, survival of the fittest, from the process of natural evolution. GA has been widely applied in economics, physics, engineering, and various other fields.

In a simple terms, according to the definition of Cambridge Dictionaries, optimisation is the process of making something as effective as possible. Thus, optimisation can be illustrated as an effort of obtaining the optimal solution of a problem under particular circumstances (Yang and Deb, 2014). Most of the systems that seek optimisation have an objective function and a number of decision variables that affect the functions over a certain search space. The optimisation method is a process of getting optimal solution that satisfies the given function as mentioned above. A generic mathematical optimisation (Yang, 2010a) can be formulated as;

$$\underset{x \in \Re}{Minimize}\ f_i(\overrightarrow{x}), \overrightarrow{x} = [x_1, x_2, \cdots, x_n] \tag{1.1}$$

subject to

$$g_i(x) \leq 0, for i = 1, \cdots, q$$
$$h_j(x) = 0, for j = 1, \cdots, m$$

where $f_i(x), \phi_j(x) and \psi_k(x)$ are functions of the design vector

$$x = (x_1, x_2, ..., x_n)^T$$

where x constitutes components of decision variables or design variables. The variables are either continuous, discrete or a mixture of continues and discrete. The $f_i(x)$ are called the objective functions. The $\phi_j(x)$ are called the equalities and and $\psi_k(x)$ the problem inequalities, for constrained optimisation problems. The $\Re^n$ is called the search space, where the space is spanned by the decision variables, x_i. In order to classify the optimisation problem in terms of objective function, if $M = 1$, the problem is called single objective optimisation. Whereas, if $M > 1$, it is called multi-objective optimisation. Multi-objective optimisation is also referred to as multi-criteria in the literatures. In real engineering applications, multi-objective optimisation problems are mostly dealt with.

Generally, optimisation algorithms can be categories into two; stochastic and deterministic algorithms. Classical optimisation methods such as Newton method, gradient method, golden mean, modified Newton method, as well as methods for constrained optimisation such as Lagrange methods, including Linear and Quadratic Programming are all in the class of deterministic methods. They are largely dependent on gradient information and ideal for unimodal functions that have one global optimum. However, deterministic algorithms face difficulty in solving problems with multimodal functions or problems where the gradient is very small such as flat regions (Tang and Wu, 2009). Therefore, the introduction of stochastic algorithms is preferred as they can escape from local minima and produced better performance (Yang, 2010a).

Metaheuristic algorithms could be regarded as a subset of stochastic algorithms. Some studies in the literature tend to refer to stochastic algorithms as metaheuristics (Blum and Roli, 2003; Yang and He, 2013). Heuristic means 'to discover solution by trial and error' (Yang, 2010d). Meta-heuristic is defined as 'higher-level' heuristic, where the process of search is influenced by certain trade-off between randomisation and local search (Yang, 2010d). Furthermore, the search process in a meta-heuristic algorithm and in the research in this thesis, with focus on bio-inspired algorithm, depends on balancing between exploration and exploitation or diversification and intensification.

In recent years, biologically inspired algorithms have been adopted to solve hard optimisation problems and they have shown great potential in solving complex engineering optimisation problems (Yang and He, 2013). The success of these methods depends on their ability to maintain proper balance between exploration and exploitation by using a set of candidate solutions and improving them from one generation to another generation. According to Simon (2013), the exploitation refers to the ability of the algorithm to apply the knowledge of previously discovered good solutions to better guide the search towards the global optimum. The exploration refers to the ability to investigate the unknown and less promising regions in the search space to avoid getting trapped in local optima (Simon, 2013).

Numerous biologically inspired algorithms have been developed by researchers. Most of the algorithms are nurtured and inspired by the evolution of genetic, the swarm behaviour of animal and also inspired from common ecological phenomena. Between the 1950s and late

1970s, these algorithms such as evolutionary algorithms (EA) (Fraser, 1957), evolutionary programming (EP) (Fogel, 1966), evolutionary strategy (EP) (Rechenberg, 1973) and genetic algorithm (GA) (Holland, 1975) have been developed. They are mostly inspired by the process of genetic evolution. They are also population-based stochastic algorithms that perform based on best-to-survive criteria (Tang and Wu, 2009). These algorithms are introduced as alternatives to deterministic method (Binitha and Sathya, 2012) and are becoming powerful in modern numerical optimisation (Yang, 2009).

There are a number of algorithms that inspired by animal swarm behaviours or swarm-based algorithms have been developed. These swarm-intelligence based algorithms full under bio-inspired optimisation algorithms where the intelligence is attributed to the social behaviour of animals and insects in nature. In the past two decades, these algorithms have drawn attention of research communities as they appear differently from the classical EAs. They operate without using evolutionary operators, hence, the stochastic search tracking of the algorithms are more direct (Tang and Wu, 2009). Among these, Kennedy and Eberhart (1995) proposed particle swarm optimisation (PSO) based on social behaviour of bird swarms. Inspired on foraging of ants, Dorigo et al. (1996) proposed ant colony optimisation (ACO). Other examples include, bacteria foraging algorithm (BFA) which inherit the characteristics of bacterial foraging patterns (Passino, 2002) and artificial bee colony (ABC), which simulates the foraging behaviour of a swarm of bees (Karaboga, 2005). Inspired by the flashing pattern of a swarm of fireflies, Yang (2010d) proposed a new swarm intelligence based algorithm called firefly algorithm (FA).

Another class of population-based optimisation models is inspired from natural ecology phenomena. Examples of bio-inspired algorithm based on ecological mechanism are invasive weed optimisation (Mehrabian and Lucas, 2006), gravitational search algorithm (Rashedi et al, 2009), spiral optimisation (Tamura and Yasuda, 2011), galaxy-based search algorithm (Shah-Hosseini, 2011) and flower algorithm (Yang et al, 2013). Invasive weed optimisation (IWO) algorithm is one of the promising recent developments in this field. The IWO algorithm is inspired by the natural ecological phenomenon and mimics the behaviour of weeds occupying suitable place to grow, reproduce and colonize the area. It has the robustness, adaptation, and randomness features and is simple but effective with accurate global search ability. This section will concentrate on the FA and IWO and their potential in building novel bio-inspired optimisation algorithms for solving problems in engineering and sciences.

1.2.1 Firefly Algorithm

Firefly algorithm is one of the population-based optimisation algorithms and in the family of swarm intelligence algorithms introduced by (Yang, 2009). It is inspired by the social behaviour of a group of fireflies that interact and communicate via the phenomenon of bioluminescence produced in the insect's body.

This metaheuristic algorithm is much simpler in concept and implementation than other swarm algorithms because it has the advantage of finding optimal solution with its exploitation capability. In general, FA is based on random search movement of fireflies, and so it is easy to achieve the global best values. Yang (2009) proves that FA is very efficient in dealing with multimodal problems as well as performs better than other bio-inspired optimisation algorithms. As such, it has attracted much attention to solve various optimisation problems (Apostolopoulos and Vlachos, 2010; Coelho and Mariani, 2012; Maricelvam et al., 2014; Olamaei et al., 2013). Appendix A.1 shows the basic flow-chart of the firefly algorithm.

1.2.2 Invasive Weed Optimisation

Another promising recent development in the area of bio-inspired optimisation algorithm is the IWO algorithm, which was proposed by Mehrabian and Lucas (2006). This population-based optimisation model is inspired from common ecological phenomena of survival of weeds. The algorithm is inspired by the natural ecological phenomenon and mimics the behaviour of weeds occupying suitable place to grow, reproduce and colonize the area. It has the robustness, adaptation, and randomness features and is simple but effective with accurate global search ability. It has also been applied to many engineering and non-engineering fields (Ahmadi and Mojallali, 2012; Ghasemi et al., 2014; Nikoofard et al., 2012; Zaharis et al., 2013). Appendix A.2 shows the basic flow-chart of the invasive weed optimisation algorithm.

1.3 Problem Statement

The discussion above of reported literature has highlighted the capability of bio-inspired optimisation especially FA and IWO in solving complex problems in science and engineering. Later chapters will highlight performances of variants of FA and IWO algorithms enhanced or hybridization with other bio-inspired algorithm as reported by other researchers.

Therefore, there is a need to develop enhanced algorithms mimicking the exact working mechanism of firefly and weed population. This potential could lead to self-evolving, truly intelligent, more powerful and more biologically-based algorithms. The natural swarm of fireflies and weeds survival provides rich source of mechanism that could improve the algorithms.

A great potential can also be explored through hybridization with other algorithms. To date, there has been no research effort at hybridizing FA and IWO algorithms. As both algorithms have their own strong features in solving single and multiple objective problems, hybridizing them could utilize both potentials to produce novel algorithms that perform better and more efficiently. Furthermore, a study of using swarm-based algorithm and nature-based algorithm is also a potential domain of the research.

1.4 Aim of The Research

The aim of the research is to improve the performance of FA, IWO and develop hybridization versions of both algorithms. The developed optimisation problems are aimed to have improvement in terms of convergence speed and accuracy especially in comparison to their respective predecessors in solving single objective, constrained and multiple objective optimisation problems.

There are opportunities for improvement of FA and IWO in terms of convergence accuracy and speed. Local information during the optimisation process is one of the areas that could be explored more and used to improve the search process. Some researchers such as Yu et al. (2014) and Wang et al. (2016) have initiated improvements by using local information. However, there are still further strategic approaches and the use of information that could lead to improved performance of the algorithm. On the other hand, at present, there is no literature referring to hybridization between firefly and invasive weed optimisation algorithms for improved performance.

Hence, the adaptive versions of FA and IWO are developed in this research by utilising the information available during the search and iteration process. The algorithms are aimed to reflect the given information and the movement of the group to respond. The innovative movement mechanism of fireflies and weeds to improve their diversification and intensification process in finding the optimum solution. The adaptation will help the algorithm to get better solution, fast convergence and maintain good accuracy to the global optimum solution of the problem in hand.

Through benchmark and practical applications, the proposed variants will be examined in solving various optimisation problems. The algorithms will be subjected to tests with single and multi-objective well-known benchmark functions including benchmark functions provided by Congress on Evolutionary Computation (CEC). For single optimisation problems, 12 test functions of CEC 2014 are used for benchmark function and for constrained optimisation problem, 10 functions of CEC 2006 are used in this thesis.

The proposed optimisation algorithms are further subjected to tests with engineering problems particularly in dynamic system modelling and controller design. These include modelling and control of flexible systems and wearable exoskeletons. The flexible systems considered comprise a twin rotor system (TRS) and a single-link flexible manipulator system (FMS). The wearable exoskeletons considered include models of upper and lower limb exoskeletons. The performances of the developed algorithms are assessed in comparison to the original FA and IWO.

1.4.1 Research Objectives

The main objectives of the research are as follows:

1. Investigate and develop adaptive FA and IWO algorithms that are better than their pre-

decessors in solving single unconstrained and constrained single objective optimisation problems.

2. Research and develop hybrid versions of FA and IWO that are better in convergence and fitness accuracy in solving single optimisation problems and constrained objective optimisation problems.

3. Investigate and test the developed algorithms to solve multiple objective optimisation problems.

4. Assess the performance of the developed adaptive FA, IWO and hybrid versions by employing the algorithms in dynamic modelling and control of a twin rotor system, a flexible manipulator and wearable upper and lower extremity exoskeletons.

1.5 Research Methodology

This section describes the adopted methodology and techniques used in this research. A flow of the research methodology is presented in Figure 1.1 and the various steps are briefly described below.

1. Formulating research problem.

   A thorough search of the given topic in the potential area in biologically inspired optimisation is carried out to provide the idea of formulating the research problem. Thus, expected improvements of firefly algorithm and invasive weed optimisation and associated validation of the algorithms are highlighted.

2. Literature survey.

   An extensive literature survey of the problem domain is carried out. This will allow identify current trends of the techniques and methodologies and associated problems encountered. Potential improvements and open problems are noted, and the findings are categorised into developments in firefly algorithm and its applications, invasive weed and its applications.

3. Finding research gap.

   The literature review will provide a clear picture of the research gap. The variants and improvements to the algorithms with recent applications will be reviewed to identify the shortcomings and potential areas for further improvement.

4. Based on the identified research gap, proposed approaches will be formulated.

   - Formulation of proposed approaches with focus at modification of the algorithms (FA and IWO) and whether they could achieve improved performance in comparison to the original algorithms.

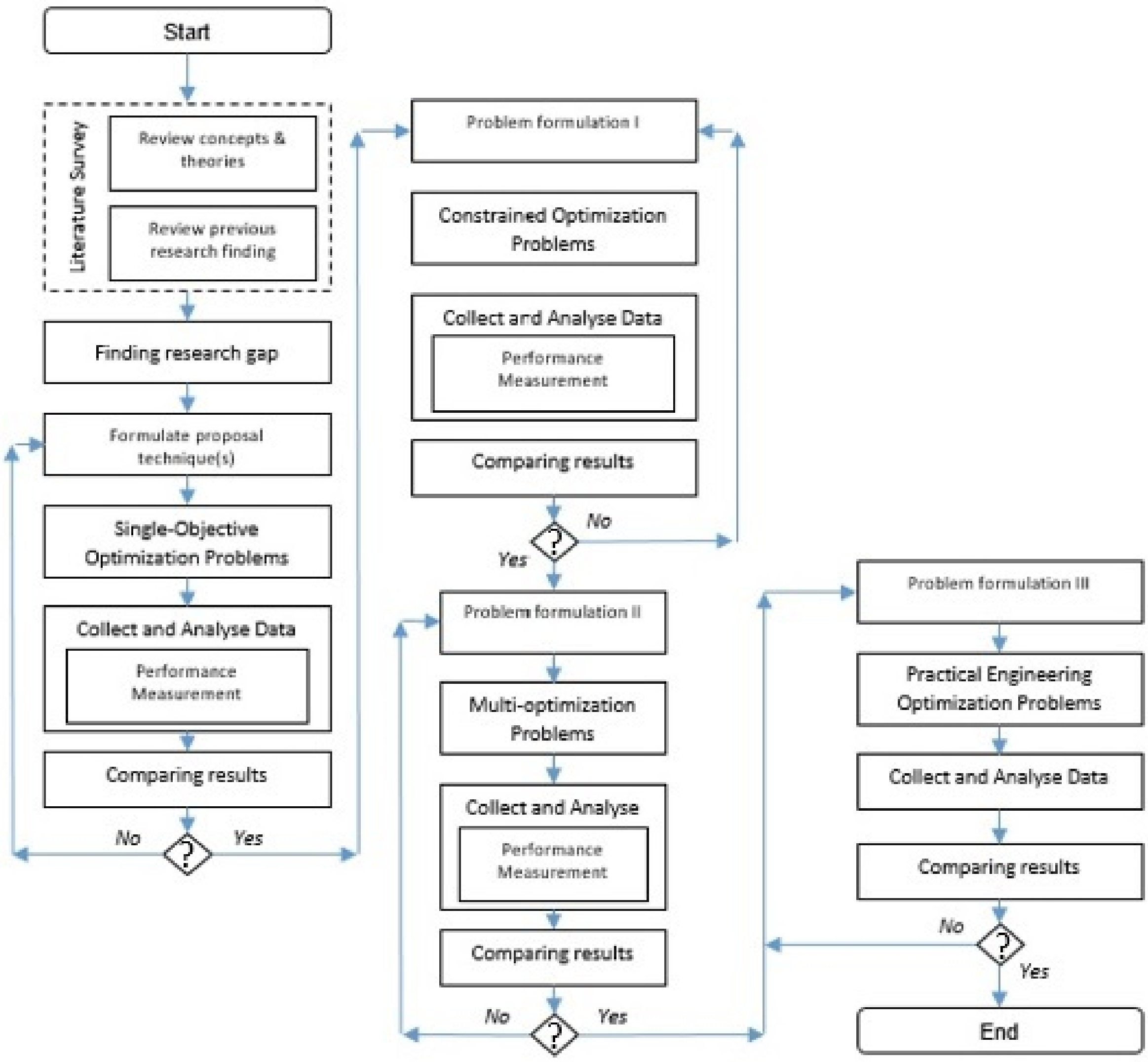


Figure 1.1: The proposed flow-chart for the research

- Problem formulation I and II focus at development of proposed algorithms in context of specific optimisation problems (i.e., constrained optimisation problems and multi-objective problems). Comparative assessments with the original algorithms are also carried out.
- Problem formulation III is concerned with the development of modelling and control techniques using the proposed algorithms and the original FA and IWO algorithms. The main objectives are to arrive at optimal parameter set in a dynamic modelling and controller design contexts.

5. Determining the optimisation problem.

   In this research, the proposed algorithms are tested, evaluated and verified with two sets of optimisation problems; benchmark functions and engineering applications.

   - Benchmark functions.

A set of benchmark test functions are used in the research. The target is to assess the performances of the proposed algorithms.

- Single objective optimisation problem.

  Well-known benchmark functions and CEC 2014's benchmark problems are used to test the proposed algorithms. Certain performance metrics are used to evaluate the proposed algorithms.

- Constrained optimisation problem.

  A basic benchmark problems and selected CEC 2006's benchmark problems for constrained problems are used. Four well known practical engineering problems that are concerned with constrained optimisation problems are also being used.

- Multi-objective optimisation problem.

  Well-known benchmark problems are selected and used. Performance metrics are also used in this section to evaluate the performance of the proposed algorithms in comparison to their predecessors.

- Practical engineering optimisation problems.

  There are four engineering applications used in this research; twin rotor system (TRS), single-link flexible manipulator system (FMS), human arm model and lower limb exoskeleton model. The proposed algorithms are tested in modelling and control exercises in these applications.

6. Collect and analysed data.

   The algorithm's performance is measured on each optimisation problem independently. The results obtained for each problem are aggregated to form a more general picture. For a comparative assessment of the algorithms, the number of iteration (it_{max}) and the number of function evaluations (NFE) are used as standard approach in the computation process. Most of the convergence graphs will be shown over a fixed number of iterations.

   Most of the performance measurements of the algorithms shown in this thesis are mainly done by determining the fixed target approach (Hansen et al., 2010). In this approach, final optimisation value is measured by an accurate time target or in this case fixed number of iterations. The number of population is also fixed for all the algorithms used, hence, the NFE are calculated the same for each problem.

   A good algorithm will gives better convergence and fitness accuracy. As most of the optimisation problems focus on minimisation problems, the optimum fitness value, f_{best} is calculated at the end of the fixed iteration period. Another concept called, optimisation error shows the difference between $f_{best} - f_{optimum}$, where $f_{optimum}$ is the optimum value of the objective problem. Over a fixed iteration, the 'best' algorithms can be said to solve the problem accurately (effectively) and fast (efficiently) after running the optimiser over a certain period of time (Opara and Arabas, 2011). In evaluation using benchmark functions, the number of runs are number of runs is set the same. This is because, it could be more precise on relative terms.

7. Comparing results, evaluation and interpretation.

   The analysed data shows the performance and achievement of the proposed algorithms. The results are evaluated and these interpretation these interpretations may lead for to further algorithm enhancement and future research. For single optimisation, constrained and multi-objective optimisation problems, the quality of final solution achieved by the algorithms will be compared and evaluated. In single optimisation problems, the algorithms also are evaluated in context of robustness. In this study, each function is evaluated by a set of selected pre-defined threshold value. The stopping criterion is fixed and it is smaller than pre-defined threshold. The robustness of each algorithm is measured by the success rate (SR) (Roy et al.,2013), which evaluates the algorithm based on consistency and successful converge to the threshold value. For handling multi-objective problems, a set of performance metrics is defined to measure the properties of the non-dominated solution obtained by the algorithms. The hypervolume (HV), spacing (SP) and maximum spread (MS) will show a measure of the convergence, uniform distribution and extensiveness (Jariyatantiwait and Yen, 2014) of the population during the search process in obtaining the non-dominated solutions of each multi-objective problem.

1.6 Contributions and Publications of The Research

The main contributions of this research can be highlighted as follows:

1. An adaptive parameter mechanism of firefly's movement in the firefly algorithm. The randomization and attractiveness parameter are adapted with a range between the lowest and highest fitness value during the iteration process. The mechanism is enhanced by decrementing nonlinear and exponential changes of the parameters. The corresponding improved versions of FA include the following:

   - Firefly algorithm with nonlinear spread factor, FA-NSF.
   - Firefly algorithm with exponential spread factor, FA-eSF.

2. An adaptive parameter mechanism of seeds distribution in the invasive weed optimisation algorithm. An exponential decrement mechanism is proposed to the value of standard deviation, SD of seeds distribution. The mechanism is also enhanced by the range of lowest and highest fitness values of plants in each generation during the iteration process. The resulting improved versions of IWO include the following:

- Invasive weed optimisation with exponential seeds spread factor, IWO-eSSF.
- Modified invasive weed optimisation with exponential seeds spread factor, MIWO-eSSF.

3. A novel algorithm by hybridizing the firefly algorithm and invasive weed optimisation. The algorithm is also enhanced by adaptive parameter mechanism as implemented in item 1 and 2. The hybridization algorithms are:

   - Hybrid invasive weed firefly optimisation, HIWFO.
   - Hybrid invasive weed firefly optimisation with spread factor, HIWFO-SF.

4. The proposed algorithms are evaluated in numerical benchmark problems such as 10 well known benchmark problems and CEC 2014 for single objective optimisation problems, CEC 2006 for constrained optimisation problems and practical engineering constrained problems. The algorithms are also evaluated with multi-objective benchmark problems.

The findings in this research have produced new contributions to knowledge of bio-inspired optimisation algorithm that will benefit the optimisation communities. The algorithms may also outperform other bio-inspired optimisation methods in certain types of problems.

Publications from this research either accepted or in print as follows. There are also further publications that are being prepared for submission.

- Hyreil A.K., Yahya N. M., Tokhi, M.O. (2015). Hybridizing firefly algorithm with invasive weed optimisation for engineering design problems. In Evolving and Adaptive Intelligent Systems (EAIS), 2015 IEEE Conference on (pp. 41-46). IEEE.

- Hyreil A.K., Assemgul M., Tokhi, M.O. (2015). Fuzzy logic based controller for a single-link flexible manipulator using modified invasive weed optimisation. In Evolving and Adaptive Intelligent Systems (EAIS), 2015 IEEE Conference on (pp. 117-122). IEEE.

- Hashim R., Hyreil A.K., Tokhi, M.O. (2015). Control of a single link flexible manipulator system using simple modified artificial bee colony optimisation algorithm. Poster session presented at the ACSE PGR Symposium 2015. Department of Automatic Control and Systems Engineering, The University of Sheffield, United Kingdom.

- Hyreil A.K., Yahya, N.M., Tokhi, M.O. (Submission 2016). Hybridizing invasive weed optimisation with firefly algorithm for unconstrained and constrained optimisation problem. Journal of Theoretical and Applied Information Technology (submitted on 30th September 2016, under review).

- Hyreil A.K., Yahya, N.M., Tokhi, M.O. (Submission 2016). Improved invasive weed algorithm with seed-spread factor for solving numerical constrained optimisation problems. Applied Soft Computing (submitted on 15th January 2016, under review).

1.7 Organisation of The Thesis

This section presents a brief description of the contents and organisation of the thesis.

Chapter 1 This chapter presents brief description of bio-inspired optimisation algorithm, the research background, aims and objectives. The chapter also explains the research methodology, contributions of the research and organization of the thesis.

Chapter 2 This chapter contains brief overview of FA and IWO and associated developments and applications.

Chapter 3 This chapter describes the development and modification made to improve the FA and IWO algorithms and introduces the new hybrid optimisation algorithms by combining firefly and invasive weed optimisation algorithms.

Chapter 4 This chapter presents the testing method to verify the performance of the proposed algorithms by using various benchmark test functions. In this chapter, single objective and constrained optimisation benchmark functions are used. The proposed algorithms are validated with performance metrics and statistical analysis.

Chapter 5 This chapter presents a brief summary of multi-objective test functions. The proposed algorithms are compared and evaluated with the multi-objective optimisation problems. The performance of the proposed algorithms on the benchmark functions are shown and further evaluated with a specific performance metric.

Chapter 6 This chapter investigates the application of the proposed algorithms on practical engineering applications. The proposed algorithms are applied to parametric modelling of twin rotor system and brief explanations of the modelling strategy that utilise the proposed algorithms are given. A Proportional-Derivative (PD)-like fuzzy logic control (FLC) is optimised by the proposed algorithm for position tracking control of a single-link flexible manipulator system. The proposed algorithms are further used to optimise the control parameters of position tracking control of human arm and lower limb exoskeleton model.

Chapter 7 This chapter summarises the research work that has been presented throughout the thesis. Further improvement of the current research findings are suggested for future works.

Chapter 2

Firefly and Invasive Weed Optimisation Algorithms: An Overview

2.1 Introduction

In recent years, bio-inspired optimisation techniques have been widely used in solving various engineering optimisation problems. They have also been developed and implemented to solve various problems in economics and other applications. The most pre-dominant classes of metaheuristics algorithms are evolutionary algorithms (EAs) and swarm-intelligence based algorithms that are based on natural evolution and collective behaviour living species. There also exist other metaheuristics algorithms that are based on natural ecosystems. This chapter provides an exploration of FA which is one of the swarm-intelligence algorithms and IWO, a powerful natural ecosystems algorithms. Brief concepts, recent modifications and applications of firefly and invasive weed optimisation are explored and over-viewed. The target of this chapter is to investigate the characteristics of the original FA and IWO for further improvement. The concept of parameters modifications of the attractiveness and randomness in FA and seeds distribution in IWO are the areas of focus. Recent implementation of the FA and IWO variants for constrained and multi-objective optimisation problems are also highlighted.

2.2 The Firefly Algorithm

Firefly algorithm is a metaheuristic algorithm inspired by the social behaviour of a group of fireflies. It was introduced by Yang (2010d). During the optimisation process, the algorithm attempts to move the particles or fireflies as inspired by the interaction of real fireflies. As each firefly produces light based on the phenomenon of bio-luminescence, certain suggestions are made in the algorithm. In principle, each firefly will be exploring and searching for other fireflies and preys randomly. Yang (2010a) suggests that each firefly will produce its own light intensity based on its body-flashing pattern, which also determines the brightness of the firefly. The firefly has the tendency to be always attracted to brighter ones. The brightness of each firefly is determined by the landscape of the objective function. Therefore, the variation of light intensity produced by each firefly in the search region is associated with the encoded objective function. Hence, the development of the original algorithm introduced by Yang (2010d) is based on the following assumptions:

- All the fireflies are assumed to be unisex. Therefore, each firefly will be attracted to another regardless of its sex.
- The attraction between fireflies is determined by the brightness of each firefly and in a proportional manner. Therefore, if one firefly is brighter than others, it will attract others to it. Otherwise, if found none, the firefly will move randomly in the search space.
- The brightness of a firefly is assumed based on the landscape of the objective function.

For most practical implementations, Yang (2010a) has suggested that each firefly will produce its own light intensity that determines the brightness of the firefly. The variation of light intensity produced is associated with the encoded objective function. Hence, in finding solution in an optimisation problem, the light intensity at location x can simply be proportional to the objective function $f(x)$ and can be chosen as $I(x) \propto f(x)$. For any distance r, the light intensity $I(r)$ varies exponentially as:

$$I = I_0 e^{-\gamma r} \tag{2.1}$$

where I_0 is the original light intensity coefficient at $r = 0$ and γ is predetermined parameter of light absorption coefficient. The value implies that the strength of the light intensity produced will attract other firefly members. As the attractiveness of firefly is proportional to the light intensity produced by each firefly, the distance, r could be defined as the distance between any two fireflies, and the variation of attractiveness, β as:

$$\beta = \beta_0 e^{-\gamma r} \tag{2.2}$$

where β_0 is the parameter value of attractiveness coefficient at $r = 0$. For a firefly to move to another brighter firefly, assuming that a firefly j is more attractive than firefly i, Yang suggests that the movement of firefly i, towards firefly j is determined by

$$x_{i+1} = x_i + \beta_0 \exp^{-\gamma r^2}(x_j - x_i) + \alpha \epsilon_i \tag{2.3}$$

where the third term is the randomization term which consists of randomization coefficient, α with the vector of random variable, ϵ_i from Gaussian distribution. For most practical implementations, the following has been suggested (Yang, 2010a):

- The distance between any two fireflies i and j at $x(i)$ and $x(j)$ is in the Cartesian distance $r_{ij}^2 = (x_i - x_j)^2$.

- The randomization coefficient α is replaced by αS_k where $\alpha \in [0, 1]$ and the scaling parameter $S_k (k = 1, \ldots, d)$ is in the d dimensions of the actual search space of the optimisation problem.

- The light absorption coefficient γ will determine the variation of attractiveness β and $\gamma \in [0, \infty]$. However, in practice, it is suggested to assume γ = 1 and $\beta_0 = 1$.

- The population size of the fireflies n is proposed to be from 15 to 100, although for practical purpose, the best range is $n = 15\ to\ 40$.

For the FA, the randomization parameter is used for exploration. The right tuning of the randomization factor could help the algorithm control the performance by balancing the search of local and global optima. On the other hand, the attractiveness parameter critically acts as exploitation of local knowledge in the search-space concentrating the local region especially if the optimality is very close. However, this optimal may or may not be the global optimal. Hence, Yang and He (2013) stresses out that a strong local knowledge, the movement shape of the algorithm and the capability of storing history / memory of local information in the search space will be helpful to the exploitation components.

Since the algorithm introduced in 2009, many complex problems have been solved using FA. Yang and He (2013) explained that this algorithm shows better potential together with other bio-inspired algorithms such as PSO, bat algorithm (BA) and cuckoo search (CS) in solving complex engineering problems. Therefore, based on the literature, some of the FA applications in the domain of engineering, computer science and solving other complex optimisation problems are shown in Table 2.1:

2.2.1 The Variants of Firefly Algorithm

Although Yang (2010c) mentioned that the FA has shown to be superior over many other bio-inspired optimisation algorithms, the need for improvement of the algorithm does not stop there. One of the ways to enhance the search capability and improve the convergence of the FA is by modifying its parameters. As mentioned by Yang (2009), the control of convergence is based on the attractiveness factor such as attractive coefficient, β and light absorption coefficient, γ and also its randomization factor, α. Łukasik and Żak (2009) improved this section by putting a synergy local search by customizing the attractiveness factor based on the "characteristic length" of the optimised area and variation of attractiveness with increasing distance of communicated firefly. Numerical benchmark functions showed that the proposed algorithm is comparable with PSO. Farahani et al. (2011) presented an improved FA variant by using the concept of normal Gaussian distribution of the randomization factor and adaptive step length of the firefly movement called Gaussian distribution firefly algorithm (GD-FF).

Lui et al. (2012) introduced adaptation for both absorption and random parameter, resulting in an adaptive firefly algorithm (AFFA). These changes improve the global search

Table 2.1: A selection of FA applications

Problem		Reference
Engineering	Steel structure	Gholizadeh (2015)
	Control system	Debbarma et al. (2014)
	Mathematical modelling	Klausen et al. (2014)
	Control system	Reddy et al. (2016)
	Power electronics	Sundari et al. (2016)
		Sundaram et al. (2016)
	Wind turbine	Wagan et al. (2015)
		Younes et al. (2014)
	Smart grid	Chandrasekaran et al. (2014)
	Electrical engineering	Chandrasekaran et al. (2012)
		Gokhale and Kale (2016).
		Setiadi and Jones (2016).
		Naidu et al. (2014)
		Farhoodnea et al. (2014)
		Shareef et al. (2014)
		Mohammadi et al. (2013)
	Economic power dispatch	Liang et al. (2015)
		Apostolopoulos and Vlachos (2010).
		Abedinia et al. (2012)
		Niknam et al. (2012)
		Yang et al. (2012)
	Forecasting	Ch et al. (2014)
		Xiong et al. (2014)
	Image processing	Horng and Liou (2011)
		Vishwakarma and Yerpude (2014)
		Chen et al. (2016)
		Kanimozhi and Latha (2015).
		Mishraet et al. (2014)
	Fingerprint	Al-Ta'i and Al-Hameed (2013).
	Manufacturing	Sayadi et al. (2013)
		Singh and Shukla (2016).
		Li and Ye (2012)
	Geo-magnetic information	Ma et al. (2015)
	Chemical	Fateen et al. (2012)
		Roeva and Slavov (2012).
Computer science	Machine learning	Krawczyk (2015).
	Networking	Kim and Kim (2016).
		Rubio-Largoet et al. (2014)
		Bojic et al. (2012)
	Database system	Wozniak et al. (2016)
	Molecular computing	Chaves-González and Vega-Rodríguez (2014)
	Data mining	Banati and Bajaj (2011)
	Multimedia processing	Kanimozhi and Latha (2015)
Optimisation problem	Dynamic optimisation problems	Ozsoydan and Baykasoglu (2015)
	Flow-shop scheduling problems	Marichelvam et al. (2014)
		Sayadi et al. (2010)
		Khadwilard et al. (2011)

and local search capability by changing the parameter linearly during the iterations period. Coelho et al. (2011) proposed a modified firefly algorithm (MFA) by linearly changing the parameter values of α and γ, either in decrement or increment. They introduced chaotic firefly algorithm (CFA) that combined the chaotic sequence into the original algorithm. Coelho et al (2011) used Logistic map to solve the reliability-redundancy allocation problem. Later, they introduced Tinkerbell map (Coelho et al, 2012) to enhance the algorithm and applied to tune and optimise control variables of Wood and Berry column model and industrial-scale polymerization reactor model.

Tilahun and Ong (2012) presented a modified firefly algorithm by random movement according to the attractiveness. The algorithm helped to magnify the brighter firefly and improve the algorithm. Shafaati and Mojallali (2012) proposed a modified firefly algorithm (MFA) that gradually reduced the randomness and introduced global best term in the firefly movement to converge more quickly. They applied the algorithm to development of the learning rule for identification of three IIR benchmark and nonlinear plants. Tian et al. (2012) introduced an inertia weight on the updating of firefly giving rise to so called inertia weight firefly optimisation (IWFA). The inertia weight decreases linearly during the iteration process.

Later, Gandomi et al. (2013) proposed use of chaotic method with a list of chaotic variables called chaotic optimisation algorithm (COA) where each chaotic variable tuned both parameters and a comparison of performance was made. Yan et al. (2012) employed three adaptive mechanisms and proposed an algorithm referred to as new adaptive firefly (AFA). The simulation results with different dimensions showed that AFA was comparable with DE and superior to PSO and FA.

Yu et al. (2014) proposed a new adaptation strategy on the randomisation parameter called wise step strategy for FA (WSSFA). In WSSFA, the wise step of the randomisation is considered by taking the absolute distance between the firefly's global best position and best positions during the iteration process. Wang et al. (2016) proposed FA with random attraction (RaFA), which employs a randomly attracted model. In RaFA, each firefly is attracted to another randomly selected firefly. In order to enhance the global search ability of FA, a concept of Cauchy jump is utilised.

Recently, Wang et al. (2016) presented a new adaptation mechanism for FAs' parameter called adaptive control parameters (ApFA). Comparative assessment in simulations of ApFA with standard FA and other variants of FA on benchmark functions have shown that ApFA outperformed those algorithms. In addition, Wang et al. (2016) also proposed NSRaFA in which three neighborhood search and a new randomization model are employed to improve the exploration and exploitation abilities. The algorithm proposed is also capable of adjusting the control parameters automatically during the search process.

2.2.2 Hybridization of Firefly Algorithm

Method of hybridization are also used to improve the optimisation method. Yang (2010b) formulated a new hybrid algorithm by combining firefly with Lévy flights. The Levy flights is used to enhance the search strategy of the algorithm and showed superior results compared to the firefly algorithm. Hassanzadeh and Meybodi (2012b) proposed a hybrid approach that combined firefly algorithm with K-means called K-FA. This new approach is used to solve data clustering problems. Hassanzadeh and Meybodi (2012a) presented a new hybrid model CLA-FA, which combined cellular learning automata (CLA) and FA. CLA is used to adapt the FA parameter and improve the algorithm in terms of global search and local search processes. Farahani et al. (2011) employed learning automata to adjust the firefly behaviour and GA to enhance global search.

Abdullah et al. (2012) proposed a new hybrid optimisation method called Hybrid Evolutionary Firefly Algorithm (HEFA). The method combines the standard FA with evolutionary operations of differential evolution (DE) method to improve the searching accuracy and information sharing among the fireflies. The HEFA method is used to estimate the parameters in a complex and nonlinear biological model to address its effectiveness in high dimensional and nonlinear problems. Fister et al. (2012) combined memetic algorithm and firefly algorithm and proposed memetic firefly algorithm (MFFA). The firefly algorithm hybridized with local search heuristic, memetic algorithm to solve combinatorial optimisation problems. El-Sawy et al. (2012) presented a new hybrid mechanism by incorporating concepts from ant colony optimisation (ACO) and FA. It is named ant colony-firefly algorithm. The methodology of the proposed algorithm used parallel mechanism of ACO and FA for updating the solutions.

Guo et al. (2013) proposed a hybrid metaheuristic approach by hybridizing harmony search (HS) and FA, namely, HS/FA. HS/FA is used to solve function optimisation. HS/FA combines exploration of HS with exploitation of FA. Riz-Allah et al. (2013) presented hybridization between ant colony and firefly algorithm, named ACO-FA. The FA worked as a local search and the randomization parameter in FA is decreased gradually during the iteration process. A novel hybrid FA with Pattern Search algorithm, called h-FAPS technique has been proposed by Mahapatra et al. (2014). The algorithm has been applied to design of a Static Synchronous Series Compensator (SSSC)-based power oscillation damping controller. The proposed h-FAPS technique is employed to search for optimal controller parameters.

Later, Rahmani and MirHassani (2014) presented a hybridization with GA to solve discrete optimisation problem. The algorithm is applied to capacitate facility location problem (CFLP) which is a well-known combinatorial optimisation problem. Tuba and Bacanin (2014b) employed an improved seeker optimisation algorithm (SOA) with firefly algorithm to build new hybrid FA algorithm. The approach uses either SOA or FA to enhance the exploitation search of the algorithm. Tuba and Bacanin (2014a) proposed a new hybridization with ABC for application to the cardinality constrained mean-variance (CCMV) problem, in the field of portfolio optimisation model. The ABC algorithm improved as the FA is incorporated to enhance the process of exploitation.

Rajan and Malakar (2015) presented a new hybrid algorithm combining Nelder–Mead (NM) simplex and Firefly Algorithm (FA). The NM simplex method is used to improve the exploitation section of FA and avoid premature convergence of FA. This algorithm is applied and demonstrated in solving power system ORPD problems. Sahu et al. (2015) combined Pattern Search (PS) and built a new hybrid method called hybrid Firefly Algorithm and Pattern Search (hFA-PS). The global exploration is done by FA and PS algorithm is used to enhance the local search. hFA-PS algorithm is used to optimise the scaling factors and PID controller gains for fuzzy PID controller of Load Frequency Control (LFC) of multi area power systems. The results outperform DE and a PSO variant. George and Parthiban (2015) proposed a new hybridized optimisation technique which employed firefly algorithm with Group Search Optimiser (GSO). The FA algorithm is used to update the worst fitness value from GSO to improve the performance. The algorithm solved multi-objective problem of clustering application.

Recently, Gupta and Arora (2016) presented a new hybrid algorithm formulated by combining FA and social spider algorithm (SSA). The proposed algorithm is tested on various standard benchmark problems and then compared with FA and SSA. Nekouie and Yaghoobi (2016) proposed a new method to enhance firefly algorithm to solve multimodal optimisation problems. The technique evolves in sub-population and utilises a simulated annealing local optimisation algorithm to increase search power, accuracy and speed of the algorithm. Table 2.2 shows a list of hybridization approaches of FA with other search and metaheuristic algorithms.

2.3 The Invasive Weed Optimisation

Another algorithm used in this research is the IWO algorithm. The algorithm was proposed by Mehrabian and Lucas in 2006. 'Survival of the fittest' is the phase that could easily explain the concept of IWO. The IWO algorithm is inspired by natural ecological phenomena and mimics the behaviour and survival of weeds occupying suitable place to grow, reproduce and colonize the area. In nature, weeds are unwanted plants. Invasive weeds are robust and vigorous able to adapt and change in the environment to survive, hence, they may pose as threat to agriculture. By imitating these natural phenomena of the invasive weeds, the IWO algorithm imitates these elements so that it has the robustness, adaptation, and randomness features. The algorithm is simple but effective and has a good exploration capability (Mehrabian and Lucas, 2006; Yılmaz and Küçüksille, 2015).

Based on the overall process of weed's behaviour, a general IWO can be represented in four steps, namely initialization of population, spacial dispersal, reproduction and competitive exclusion. It begins with initializing the initial plant in the search area. This is the initialization stage. The plant is spread randomly in the search place. Each plant is able to

Table 2.2: A selection of hybridization approaches of FA

Algorithm	Reference
Levy flight	Yang (2010b).
Differential evolution (DE)	Abdullah et al. (2012)
Ant colony optimisation (ACO)	El-Sawy et al. (2012).
	Rizk-Allah, et al. (2013).
Genetic algorithm (GA)	Farahani, Abshouri, Nasiri, and Meybodi (2012).
	Rahmani and MirHassani (2014).
Memetic algorithm (MA)	Fister Jr et al. (2012).
Group search optimisation (GSO)	George and Parthiban (2015).
Harmony search (HS)	Guo et al. (2013).
Social spider algorithm	Gupta and Arora (2016).
Cellular learning automata	Hassanzadeh and Meybodi (2012a).
K-means	Hassanzadeh and Meybodi (2012b).
Pattern search (PS)	Mahapatra et al. (2014).
Simulated annealing	Nekouie and Yaghoobi (2016).
	Sahu et al. (2015).
Nelder-Mead simplex	Rajan and Malakar (2015).
Artificial bee colony (ABC)	Tuba and Bacanin (2014a).
Seeker optimisation	Tuba and Bacanin (2014b).

produce seeds. In this reproduction stage, however, production of seeds depends on their relative fitness in the population. The worst member produces minimum number of seeds (s_{min}) and the best produces maximum number of seeds (s_{max}) where the weeds production of each member is linearly increased. The next stage is called spatial dispersal where the seeds are randomly scattered over the search space near to their parent plant. The scattering process uses normally distributed random number with standard deviation (SD) as;

$$\sigma_{iter} = \left[\frac{iter_{max} - iter}{iter_{max}}\right]^n (\sigma_{max} - \sigma_{min}) + \sigma_{min} \tag{2.4}$$

where $iter_{max}$ is the maximum iterations, $iter$ is current iteration, n is the nonlinear modulation index, σ_{max} is usually initial SD and σ_{min} is the final SD in the optimisation process. The seeds with their respective parent plants are considered as potential solutions for subsequent generations. This step provides nonlinearly decreasing and dropping of weeds to local minima position. In order to maintain the size of population in the search area, the algorithm conducts a competitive exclusion, where an elimination mechanism is employed; if the population exceeds its maximum size only the plants with better fitness can survive. Those with better fitness produce more seeds and with high possibility of survival and become reproductive. The process continues until the maximum number of iterations is reached and the plant with best fitness is closest to the optimal solution.

There are certain properties of IWO that can be considered as sophisticated tools for solving complex optimisation problems. IWO algorithm allows all the plants to be involved in the reproduction process as each plant is capable of producing seeds. The fitter plants produce more seeds, impacting on the convergence capability of the algorithm. Although the less fit plants show lower fitness level, they also produce seeds that may carry valuable information around its area. Thus, if at later generation / iteration they have good fitness, they can survive (Mehrabian and Lucas, 2006) and this is one of the properties that prevents pre-mature convergence. Another property of IWO algorithm is that the plants produce seeds independently without a mating process. This feature could add a new attribute to the algorithm that each weed / plant may have a different number of variable during the optimisation process (Yilmaz and Kucuksille, 2015). These variables can be chosen as one of the IWO optimisation parameters (Yilmaz and Kucuksille, 2015). The IWO algorithm shows continuous and normal distribution of the seed dispersal over the search space. As the variance parameter is centred on each parent plant, the algorithm has better chance to avoid local minima points as compared to GA and PSO (Rad and Lucas, 2007). Since introduced in 2006, IWO has been used in solving a lot of complex problems. Table 2.3 shows a selection of applications of IWO in the domain of engineering, computer science, mathematics and solving other complex optimisation problems:

2.3.1 The Variants of Invasive Weed Algorithm

Since Mehrabian and Lucas introduced IWO in 2006, a lot of IWO variants have been proposed by researchers to enhance and improve the algorithm. The variants use properties of IWO as sophisticated tools in efforts to enhance the algorithm. Suresh et al. (2009) proposed a new adaptive IWO, namely IWO with increased deviation and stochastic selection (IWO-ID-SS). The variant increased the standard deviation value linearly and equipped the improved algorithm with better exploration power. Simulation results on noisy functions have shown competitive results as compared to state-of-the-art algorithms.

Roy et al. (2011) presented a modified IWO by improving the standard deviation of seed distribution and applied it to the design of antenna arrays. Simulation results compared with state-of-the-art algorithm, showed competitive results. Ghosh et al. (2011) studied standard deviation during seed distribution process and tried to improve IWO by reducing the standard deviation per iteration. The variant has been used for solving optimal control problems by using Bezier control parameterization (BCP). Ahmadi and Mojallali (2012) proposed chaotic invasive weed optimisation (CIWO) that incorporates capability of chaotic search methods. The newly generated seeds are chaotically distribution in the search space to improve the search process. Zaharis et al. (2013) presented a modified adaptive dispersion IWO (MADIWO) to optimise data sets of neural network training for an antenna design problem. The mechanism of adaptive seed dispersions is implemented to explore better position during

Table 2.3: A selection of IWO applications

Problem		Reference
Engineering	Neural network training	Giri, et al. (2010b)
	Mathematical modelling	Abu-Al-Nadi et al (2013)
	Control system	Roy et al. (2010)
		Chen et al. (2011)
	Robot trajectory	Sengupta et al. (2011)
Engineering	Image processing	Pourjafari and Mojallali (2012)
	combustion at a coal-fired utility boiler	Zhao et al. (2009)
	Electrical engineering	Saravanan et al. (2014)
		Ghasemi et al. (2014)
	Power electronics	Wang et al. (2015)
	Economic power dispatch	Ghasemi et al. (2014)
		Nayak et al. (2011)
		Jayabarathi et al. (2012)
	Electromagnetics	Karimkashi and Kishk (2010).
	Communication	Hung et al. (2010)
	Antenna design	Roy et al. (2011)
		Roshanaei et al. (2009)
	Flexible structure	Mehrabian and Yousefi-Koma (2007).
	UAV	Ghalenoei et al. (2009)
Computer science	DNA computing	Zhang et al. (2009)
	Networking	Rakshit et al. (2012)
	Recommender system	Rad and Lucas (2007).
	Web selection	Su et al. (2014)
	Binary encoding	Veenhuis (2010).
Mathematic	Chaotic system	Ahmadi and Mojallali (2012).
Optimisation problem	Clustering	Pal et al. (2010a)
	Flow-shop scheduling	Chen et al. (2013)
		Zhou et al. (2014)
	Traveling salesman	Zhou et al. (2014)
	Path problem	Pahlavani et al. (2012)

the search process.

Recently, Peng et al. (2015) employed an adaptive invasive weed optimisation (AIWO) algorithm. They improved the global search by adding adaptive step size of the parameter in the algorithm. Li et al. (2015) proposed a modified hybrid invasive weed optimisation (MHIWO) and applied it to antenna design. MHIWO exhibits piecewise standard deviation and implements t-distribution function in the spatial distribution to enhance the exploration ability. Ouyang et al (2016) presented an improved IWO (IIWO) algorithm to solve large global optimisation (LGO) problems (CEC'2010 high-dimensional benchmark functions). This variant also applied in GPU platform improving the algorithm by adaptive and concrete adjustment of newborn seeds setting during the iteration process.

2.3.2 Hybridization of Invasive Weed Algorithm

Many researchers tend to hybridise two or more algorithms to complement one another. In this manner, IWO has been hybridized with other search and metaheuristic algorithms to improve its performance capability. Zhang et al. (2008) proposed a new hybrid algorithm referred to as cultural IWO. The IWO algorithm is embedded into cultural framework as a population space of a Cultural Algorithm (CA). Zhao et al. (2009) presented a hybrid particle swarm optimisation with invasive weed (IW-PSO). It was proposed to optimise the constrained objective functions of combustion at a coal-fired utility boiler. The hybridization is done through parallel search process between PSO and IWO. Roy et al. (2010) employed a hybrid algorithm related with IWO algorithm, where the principle of optimal foraging theory (OFT) is incorporated in IWO to improve the search mechanism.

Das et al. (2011) proposed a hybrid algorithm by combing differential evolution and IWO. The algorithm is used to solve economic dispatch problem (EDP). The IWO and DE algorithms are applied simultaneously on two sub-populations and population exchange is incorporated to refine the quality of the population after every generation. Haider et al. (2011) introduced a hybrid approach with self-adaptive cluster based and weed inspired differential evolution algorithm (SACWIDE). The hybrid algorithm divides the total population into several clusters based on the positions of individuals and the cluster number is dynamically changed by a suitable learning strategy during the evolution. The IWO is used as a local search technique of the algorithm. Bhattacharya et al. (2011) presented a hybrid algorithm by improved IWO algorithm with roulette wheel based reproduction. The proposed roulette wheel IWO (RWIWO) algorithm used roulette wheel to decide on the number of seeds generated by each plant.

Yin et al. (2012) employed hybrid genetic to build a new hybrid algorithm to improve IWO, referred as HGIWO. The algorithm aimed to reduce the likelihood on getting into local optima. The hybridization is done by letting the weeds multiply by the selection and hybridization of genetic arithmetic. Sengupta et al. (2012) proposed a new hybrid algorithm by improving IWO with memetic approach, named Intelligent Invasive Weed Optimisation (IIWO). The hybrid approach is done by employing temporal difference Q-learning as constriction factor in the seed dispersal stage. Rakshit et al. (2012) presented a hybrid algorithm by combining ABC and IWO referred as IWO-ABC search algorithm. The hybrid IWO-ANC was applied to the Gene Regulatory Network (GRN) identification problem.

Roy et al. (2013) proposed a hybrid algorithm combining IWO and a modified group search optimiser (GSO). The hybrid algorithm is enhancing the IWO by using GSO for intensifying the candidate solution to solve multimodal optimisation problem. Zhou et al. (2013) combined differential evolution (DE) algorithm and presented a hybrid method of differential evolution invasive weed optimisation (DEIWO) algorithm. In the algorithm, global exploration ability of invasive weed optimisation algorithm is used to provide effective search area for differential evolution and the heuristic search ability of differential evolution algorithm provides a reliable guide for IWO. Basak et al. (2013) employed an efficient hybrid algorithm by embedding difference vector based mutation scheme of Differential Evolution (DE) into IWO referred as Differential Invasive Weed Optimisation (DIWO). Ghasemi et al. (2014) proposed a new hybrid algorithm based on modified imperialist competitive algorithm (MICA) and IWO for solving the optimal reactive power dispatch problem.

Barisal and Prusty (2015) presented a new hybrid method by hybridizing the oppositional based learning (OBL) and implemented in IWO algorithm, namely oppositional invasive weed optimisation (OIWO). OIWO was applied to minimize the total generation cost by satisfying several constraints in the economic dispatch problems. Yılmaz and Küçüksille (2015) tried to improve and modify bat algorithm, and the algorithm was referred to as enhanced bat algorithm (EBA). The enhanced algorithm used IWO algorithm and other two methods to improve local and global search capabilities of the algorithm. Simulation results showed that the algorithm was better than the original algorithm in solving well known benchmark function and engineering design problems. Naidu and Ojha (2015a) proposed a hybrid version of IWO with quadratic programming (QA) operator, referred to as QAIWO. The hybrid method used QA operator in guiding the parent seeds during the distribution process. Simulations were shown on a real-life optimisation problems and well-known benchmark problem. Shi et al. (2015) presented an effective hybrid IWO algorithm with simulated annealing (SA) to solve quadratic assignment problem (QAP). Ojha and Naidu (2015) combined cat swarm optimisation (CSO) with IWO algorithm and proposed a hybrid algorithm. In the hybrid algorithm, IWO is used to enhance the intensification capability of the CSO to obtain better solution.

Recently, Chifu et al. (2016) proposed a hybrid invasive weed optimisation method for generating healthy meals starting from a given user profile, a diet recommendation, and a set of food offering. The method proposed is based on a hybrid model which consists of a core component and two hybridization components. The core component is based on the invasive weed optimisation algorithm, and the hybridization components rely on PSO-based path relinking as well as on tabu search and reinforcement learning. Mahto and Choubey (2016) presented a hybridization method by combining IWO and wind driven optimisation (WDO). The hybrid algorithm is verified with six standard benchmark functions and the nulling pattern synthesis of a uniform linear array (ULA) and non-linear circular array (NUCA) antenna having a minimum side lobe level (SLL), and beam width to minimize the interference effect by optimising the parameters of array elements. Table 2.4 shows a selected list of hybridization strategies involving IWO with other search and metaheuristic algorithms in the literature.

Table 2.4: A selection of hybridization strategies of IWO

Algorithm	Reference
Oppositional based learning (OBL)	Barisal and Prusty (2015)
Bat algorithm	Yılmaz and Küçüksille (2015)
Imperialist competitive algorithm (ICA)	Ghasemi et al. (2014)
Differential evolution (DE)	Basak et al. (2013)
	Das et al. (2011)
	Haider et al. (2011)
	Zhou, Luo, and Chen (2013)
Group search optimiser (GSO)	S. Roy et al. (2013)
Genetic algorithm	Yin et al. (2012)
Memetic algorithm	Sengupta et al. (2012)
Artificial bee colony (ABC)	Rakshit et al. (2012)
Roulette wheel	Bhattacharya et al. (2011)
Optimal foraging theory (OFT)	Roy et al. (2010)
Particle swarm optimisation (PSO)	Zhao et al. (2009)
	Chifu et al. (2016)
Cultural algorithm (CA).	X. Zhang et al. (2008)
Wind driven optimisation (WDO)	Mahto and Choubey (2016)
Simulated annealing (SA)	Shi et al. (2015)
Cat swarm optimisation (CSO)	Ojha and Naidu (2015)
Quadratic approximation (QA)	Naidu and Ojha (2015a)

2.4 Solving Constrained Optimisation Problem

In solving real-world optimisation problems, constraint conditions should also be prioritized. Therefore, in the process of getting the optimal solution, huge consideration should be focused on how to satisfy the constraints involved. The optimisation has to be formularised to deal with this aspect in constrained optimisation problems. Constrained optimisation problems arise in numerous applications especially in practical engineering design, structural design, economics and location optimisation problems. There has been favourable attention given to development of algorithms for solving constrained optimisation problems over the past several years.

Mezura-Montes et al. (2003; 2010) used an evolutionary algorithm to solve constrained problems. Artificial bee colony (ABC) has been used by several researchers to solve numerical constrained problems (Akay and Karaboga, 2012; Karaboga and Akay, 2011; Li and Yin, 2014). He and Wang (2007) used a particle swarm optimisation approach named CPSO and Huang et al. (2007) used differential evolution based on co-evolutionary mechanism to solve constrained problems. Various bio-inspired methods have also been attempted by researchers to solve such problems, and these include differential search (Liu et al., 2015), firefly algorithm (Gandomi et al., 2011), cuckoo search (Gandomi et al., 2013; Bulatovic, 2014), harmony search (Mahdavi et al., 2007, Mun and Cho, 2012), artificial immune system (Zhang et al., 2014), ant colony optimisation (Kaveh and Talahari, 2010) and bacterial-inspired algorithm (Niu et al., 2015).

Hybrid algorithms by combining two or more metaheuristic algorithms have further been attempted by researchers for solving numerical constrained optimisation problems. These include PSO-ACO (Kaveh and Talahari, 2009), charge system search and PSO (Kaveh and Talahari, 2011) and glowworm swarm optimisation (Zhou et al., 2013). In the literature, a number of formulae based on FA and IWO have also been proposed.

2.4.1 Constrained Problem Approaches with Firefly Algorithm

Gandomi et al. (2011) proposed to apply the firefly algorithm to solve constrained optimisation problems that deal with mixed continuous / discrete structural optimisation problems. Yang et al. (2012) applied FA to determine feasible optimal solution of constrained economic dispatch (ED) problems. Penalty function is used to handle the constraints. El-Sawy et al. (2012) presented a hybrid concept by incorporating the concepts from ACO and FA. The algorithm was named ant colony-firefly algorithm (ACO-FFA) algorithm. The methodology of the proposed algorithm is used in a parallel mechanism of ACO and FA for updating the solutions. ACO-FFA was applied to well-known benchmark constrained optimisation problems.

Talahari et al., (2014) proposed an adaptive FA that utilizes the feasible-based method to handle constrained large-scale structure problems. Kazemzadeh-Parsi (2014) developed a modified firefly algorithm to solve classical engineering design optimisation problems and truss structures. The algorithm is modified by adding memory, mutation and proposing a new updating formula. Penalty function is used to handle the constraints of the system. Tuba and Bacanin (2014b) presented a hybrid algorithm that improved seeker optimisation algorithm (SOA) with FA. The approach was called SOA with firefly search (SOA-FS) as it used either SOA or FA to enhance the exploitation search of the algorithm. SOA-FS was applied to well-known benchmark constrained optimisation problems.

2.4.2 Constrained Problem Approaches with Invasive Weed Algorithm

Pal et al. (2009) implemented IWO for solving constrained real parameter optimisation problems by using penalty function. The simulation results showed that the algorithm was better than state-of-the-art PSO variant. Hu et al. (2014) proposed a hybrid algorithm by using memetic algorithm and used IWO algorithm as the local refinement procedure of the hybrid algorithm, referred as DE-IWO. The hybrid algorithm was used to solve constrained optimisation problems by using multi-objective method. Naidu and Ojha (2014) presented a modified IWO in solving constrained optimisation problems. The modified algorithm used simulated annealing to improve the penalty function approach. Naidu and Ojha (2015b) applied IWO to solve constrained optimisation problem by using multi-stage penalty approach.

2.5 Solving Multi-objective Optimisation Problems

In real engineering applications, design problems often involve many design variables and multiple objectives. A significant different method should be formulated in order to find solutions to the problems. In the literature, a number of metaheuristics have been proposed to deal with such optimisation problems (Abbass and Sarker, 2002; Deb and Goel, 1999; Rangaiah, 2009). Firefly and invasive weed algorithm have also been formulated to deal with multi-objective problems. In solving these problems which may have conflicting objective functions, several proposed methods have been proposed by researchers. The related literature involving FA and IWO is discussed and the approaches for solving multi-objective problems are reviewed.

2.5.1 Multi-objective Approaches with Firefly Algorithm

Apostolopoulos and Vlachos (2010) applied FA by extending the algorithm to solve multi-objective optimisation problems relating to economic emissions load dispatch problem. Kumar and Phani (2011) applied multi-objective firefly algorithm (MOFA) to solve combined economic and emission dispatch (CEED) problem in thermal power station. The aggregate approach is used to solve the multi-objective problem by formulating the Pareto optimal front of three different load demands.

Abedinia et al. (2012) proposed a MOFA to solve environmental / economic power dispatch (EED) problems. The MOFA was used to obtain a more accurate solution in solving a nonlinear constrained optimisation problem with competing objectives of fuel cost, emission and system loss which are conflicting with one another. MOFA has also been used to solve practical dynamic economic emission dispatch problems (Niknam et al., 2012). Enhanced FA with chaotic mechanism and novel self-adaptive probabilistic mutation strategy was used to improve the performance and achieve a set of non-dominated solutions of the problem.

Yang (2013) extended the firefly optimisation technique to solve multi-objective problems especially in continuous problem cases and formulated a MOFA. Well-known benchmark functions and practical engineering applications were used to test the MOFA. El-Sawy et al. (2013) presented a hybrid ant colony and firefly algorithm to solve numerical multi-objective optimisation problems. The algorithm used vector evaluated ant colony optimisation (VEACO) and firefly algorithm in solving the benchmark problems. Chandrasekaran and Simon (2013) implemented FA in solving a multi-objective unit commitment problem (MOUCP). They developed a novel methodology to employ optimal deviation based firefly algorithm tuned fuzzy member function. Marichelvam et al. (2014) applied MOFA by extending discrete firefly algorithm to solve hybrid flow-shop scheduling problems with two objectives by minimizing the sum of make span and mean flow time of the system.

2.5.2 Multi-objective Approaches with Invasive Weed Algorithm

Kundu et al. (2011) proposed IWO to solve multi-objective optimisation problem by using fuzzy dominance sorting to get the non-dominated Pareto optimal solution. The improved IWO to handle the multi-objective problems was referred to as IWO-MO, which modified the schedule to decrease the number of the seed population using the concept of fuzzy dominance for choosing the best maximum number of population members to survive in the next generation. The simulation results on some well-known benchmark functions showed competitive result as compared with state-of-the-art metaheuristic algorithms.

Hu and Cai (2012) presented an improved IWO to solve multi-objective optimisation problems. The intra and inter-communities of weeds method is used to exchange the information in the IWO algorithm to solve the multi-objective problems. Nikoofard et al. (2012) proposed an integrated approach of fast non-dominated sorting in NSGA-II and IWO, namely NSIWO to solve well known benchmark problems and complex electricity markets problems. Lui et al. (2012) employed and utilized IWO to optimise two fuzzy clustering objective functions simultaneously in solving clustering problems. For this approach, non-dominated sorting of NSGA-II is used to solve the multi-objective problems.

Recently, Pouya et al. (2016) applied IWO to solve a multi-objective portfolio optimisation problem. In the algorithm, penalty function is used to handle the constraints in solving the problem. Maghsoudlou et al. (2016) implemented IWO to solve multi-objective multi-mode resource constrained project scheduling problem. In the problem, fuzzy dominance based sorting is used to determine the non-dominated Pareto solutions.

2.6 Summary

A review the literature related to FA and IWO algorithm has been presented. The origin and the way each algorithm works, the overview of variants of FA and IWO as well as their applications have been highlighted. The variants of FA and IWO in solving more complex optimisation problems such as constrained and multi-objective problems have been briefly reviewed.

In solving unconstrained, constrained and multi-objective problems, a lot of improvements in convergence accuracy and speed have been reported in the literature using FA and IWO. Various adaptation mechanisms have been used to vary parameters of the algorithms. However, an adaptation strategy using local information during the optimisation process is one of the areas that could be explored further and used to improve the search process. Furthermore, it is worth exploring approaches that hybridise these algorithms as at present such approaches have not been explored with firefly and invasive weed optimisation algorithms.

In the next chapter, new approaches to enhance FA and IWO will be proposed and elaborated. The proposed algorithms will create new variants involving FA and IWO for applications to single, constrained and multi-objective optimisation problems. Numerical tests on the optimisation conditions and parameters of the proposed algorithm will also be explored.

Chapter 3

Adaptive Firefly and Invasive Weed Optimisation Algorithms

3.1 Introduction

This chapter presents new optimisation algorithms developed based on FA and IWO. The proposed new algorithms are established from modification of firefly and invasive weed optimisation algorithms. New hybridization algorithms combining FA and IWO algorithm are also described. Some modifications to the original FA and IWO algorithms are proposed in order to improve their optimisation capability such as convergence, optimum value, computation time and optimum solution accuracy. The potential to use local knowledge could be utilized in improving the original algorithm. The parameters of each of the algorithm are adaptively changed with the use of local knowledge during the optimisation process. Investigations on the size of population and iteration parameters are carried out to determine the best competitive condition of the proposed algorithms to be evaluated in optimisation problems. In this research, the problems are set to be continuous optimisation problems.

3.2 Adaptive Mechanism for Firefly Algorithm

It is worth mentioning that FA could solve complex non-linear optimisation problems better than evolutionary algorithms as it is easier to implement and has higher stability mechanism as well as less execution time (Nikman et al., 2012). Although the original FA algorithm is proved to perform better than other metaheuristic algorithms, it also has its limitations and weaknesses. The algorithm parameters are set fixed (Coelho and Mariani, 2012; Yang, 2009) where they do not change in time or iterations. Yan et al. (2012) have highlighted that FA with predetermined parameters works well on the functions with low dimension and narrow variable range. If the problems are more complex, the variable range and / or dimension may increases, these parameters might not be suitable and performance may be dropped (Coelho and Mariani, 2012). Therefore, modifications are needed to improve the diversification area and the speed of the algorithm to avoid premature convergence of the algorithm. FA also does not have any memorization mechanism to remember historical information, that may be useful. Coelho and Mariani (2012) have stressed that in early simulation; as the optimum value is approaching, the solutions are still changing due to fixed randomness value. Hence, the need to improve this aspect is an active research potential.

The proposed method here is based on the countermeasure of these weaknesses of the basic FA. One way to enhance the algorithm performance is to improve the exploration and intensification of the search by affecting the fixed parameter values. In the original FA, most of the parameter values in the equation of renewal movement are fixed and predetermined. In the proposed algorithms, the attractiveness and randomization parameters are pre-set at the initialization and adaptively changed during the optimisation process. The potential to use local knowledge during the iteration process is a further aspect that could be utilized in improving the original algorithm. So far, FA is memory-less and the tendency to jump out of the extreme in early iteration could be an issue as the algorithm still has big randomization factor, which means that it is always in moving mode. The objectives with the changes made are to enhance the search and local exploration, avoid unnecessary pace at any local extreme point and at the end of the search process and to enable the algorithm to move fast to the global best optimum point.

The proposed FA algorithms focus on improvement in the area of firefly movement. The mathematical expression for improved movement of the firefly in the proposed algorithms is given as:

$$x_{i+1} = \omega(t)x_i + \beta(t)(x_j - x_i) + \alpha(t)\varepsilon_i \tag{3.1}$$

where $\omega(t)$ is the inertia weight, $\beta(t)$ is the attractiveness coefficient, $\alpha(t)$ is the randomness coefficient at time t and ε_i is a random number.

In this movement equation, the randomization and attractiveness coefficients are proposed to adaptively change in nonlinear and exponential forms. In both terms, the attractiveness parameter is adaptively increased and randomization is adaptively decreased over time in the search process.

During optimisation and search process, at the early stage, the diversification phase could be increased by letting the randomization at high value and attractiveness in low value. This action would let all the fireflies to randomly move around at the early stage to fine tune the global searching process. By letting the parameter adaptively change over time, it will help extend the balance and capability of strong global search and local search of the firefly movement so that it would not let the algorithm get trapped into local extreme point at the early stage.

At the end stage of optimisation, the attractiveness is set at high value to increase the light intensity so that the firefly with extreme point produces more light to trigger other fireflies to move towards it. The randomization is set in low value so that the fireflies are forced to move gradually to the nearer local or global optimum point, relatively increase the intensity of the fireflies. These movements are targeted so that the algorithm is able to get better accuracy to of the optimum value. The adaptively nonlinear equation used in the proposed algorithm is given as

$$\sigma_{iter} = \left[\frac{iter_{max} - iter_i}{iter_{max}}\right]^n (\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.2}$$

On the other hand, the equation below shows the adaptively exponential equation used

$$\sigma_{iter} = \left(exp\left[\tau\left[\frac{iter_i}{iter_{max}}\right]^m\right]\right)(\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.3}$$

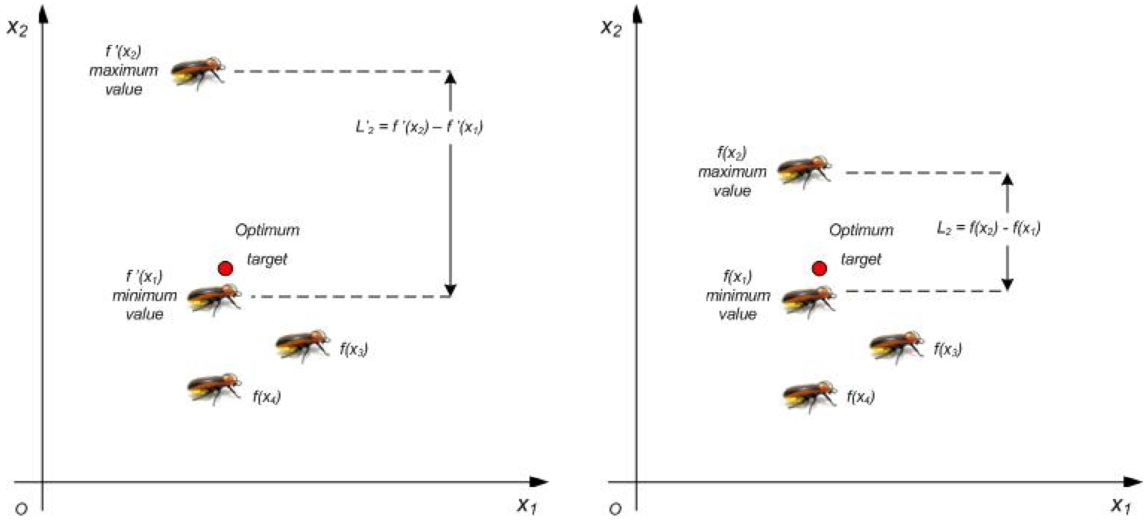
where σ_{iter} is the coefficient value at the iteration during search process with σ_{ini} as its initialized value and σ_f is the final coefficient value. Hence, in search process, σ_{iter} is in the range $[\sigma_{ini}, \sigma_f]$. $iter_{max}$ is the maximum iteration value and $iter_i$ is the present value of the iteration. The coefficient is shown in increment shape when putting small value of σ_{ini} and high value σ_f during initialization process of the algorithm. If the σ_f value is set low and σ_{ini} is set high, the decrement shape is shown. For equation (3.3), the values of τ and m are predetermined. For both equation (3.2) and (3.3), the coefficient is shown in increment shape when putting small value of σ_{ini} and high value σ_f during initialization process of the algorithm. If the σ_f value is set low and σ_{ini} is set high, the decrement shape is shown.

Another strategy used in the proposed algorithm is using local knowledge to improve the firefly movement. The synergy of using local knowledge is done by exploiting the light intensity of the neighbourhood condition during the search process. The process uses local neighbourhood knowledge at each iteration through the fitness value of the highest light intensity of firefly and the lowest value to generate a normalized factor, referred to as spread factor (SF). This factor is used in the FA coefficients to re-adjust the value and improve the movement of all the fireflies during the search process. The SF is evaluated in the proposed algorithm as:

$$SF_{iter} = \left(\frac{|f_i| - f_{worst}}{f_{best} - f_{worst}}\right)(q_i' - q_f') + q_f' \tag{3.4}$$

where SF_{iter} is the new coefficient value at the iteration during the search process with q_i' as the lowest factor value and q_f' as the highest factor value set at the initialization stage. f_i is the fitness value of present firefly and f_{max} is the highest fitness value or the firefly that produced light with lowest intensity value. f_{min} is the brightest firefly in the current iteration.

During iteration process, the SF mechanism helps the algorithm to re-adjust the parameter value. Based on Figure 3.1, if the position of the lowest intensity firefly and highest intensity is near, $L2$ the factor is in the high value and not affected by the current value. Otherwise, if it is in the opposite case, $L2'$ value is low and the SF factor will have some effect on the parameter. This mechanism is applied on firefly movement during updating the firefly after



(a) Adaptive A (b) Adaptive B

Figure 3.1: The spread factor mechanism during iteration process.

each iteration. Thus, a full description of the proposed firefly algorithms is given below.

3.2.1 Firefly Algorithm with Nonlinear Spread Factor

Firefly algorithm with nonlinear spread factor, FA-NSF is proposed by using equation (3.2) as the randomization and attraction coefficient. It will also combine with equation (3.4) of the SF to enhance adaptive mechanism of the coefficient. The adaptively nonlinear coefficient with normalized spread factor used in the proposed algorithm is thus given as

$$\sigma_{iter\,NSF} = SF_{iter}\left[\frac{iter_{max} - iter_i}{iter_{max}}\right]^n (\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.5}$$

where $\sigma_{iter\,NSF}$ is the coefficient value of both γ and α during the movement process in the proposed FA. The new mathematical expression of the improved movement of the firefly in the FA-NSF algorithm is as follows:

$$x_{i+1} = \omega(t)x_i + \beta_{NSF}(t)(x_j - x_i) + \alpha_{NSF}(t)\varepsilon_i \tag{3.6}$$

where $\omega(t)$ is the inertia weight, $\beta_{NSF}(t)$ is the nonlinearly adaptive attractiveness coefficient and $\alpha_{NSF}(t)$ is the randomness coefficient with nonlinearly adaptive mechanism at time t.

3.2.2 Firefly Algorithm with Exponential Spread Factor

Firefly algorithm with exponential spread factor, FA-eSF, uses exponential form of randomization and attraction parameters as illustrated in equation (3.3). The proposed algorithm is also combined with Equation (3.4) to enhance the adaptive aspect of the parameters. Thus, the new randomization (α) and attractiveness (γ) coefficient with exponentially adaptive SF used in the algorithm is given as

$$\sigma_{itereSF} = SF_{iter}\left(exp\left[\tau\left[\frac{iter_i}{iter_{max}}\right]^m\right]\right)(\sigma_{ini}-\sigma_f)+\sigma_f \quad (3.7)$$

where $\sigma_{itereSF}$ is the new coefficient value of both γ and α during the movement process in the proposed FA algorithm. The new mathematical expression of improved movement of firefly in the FA-eSF algorithm is as follows:

$$x_{i+1} = \omega(t)x_i + \beta_{eSF}(t)(x_j - x_i) + \alpha_{eSF}(t)\varepsilon_i \quad (3.8)$$

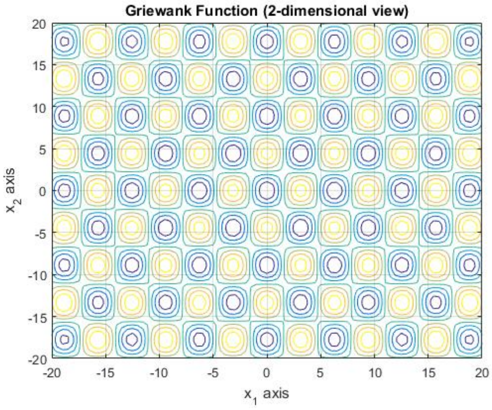
where $\beta_{eSF}(t)$ is the exponentially adaptive attractiveness coefficient and $\alpha_{eSF}(t)$ is the randomness coefficient with exponentially adaptive mechanism at time t.

With these nonlinear and exponential adjustments, the modified firefly algorithm will have better balance between global and local search capabilities, and thus will avoid getting trapped in local optimum, and this will increase the speed of convergence to better optimum solution.

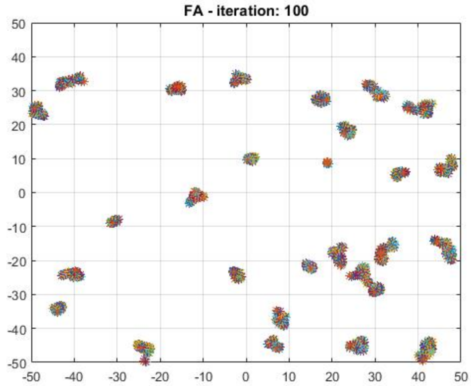
To illustrate the changes made based on the mechanism above, a simple example of comparison is made on these proposed algorithms with the original algorithm by running them on two dimensional Griewank function in the range [-30, 30]. Griewank function is a well-known single optimum multi-modal function. Thus, the function has many local optima and one global optimum at the origin in (x_1, x_2) coordinate system as shown in Figure 3.2a.

Using the same predetermined parameter and iteration of 100, Figure 3.2 shows the end result of movements of fireflies during the optimisation process of the three algorithms. In Figure 3.2b, the fireflies based on the original algorithm got trapped at the local optima. Meanwhile, the FA-NSF and FA-eSF managed to jump out of those local optima and produced better value as shown in 3.2c and 3.2d, respectively. Figure 3.3 shows the rate of change of the parameters for both proposed algorithms, FA-NSF and FA-eSF.

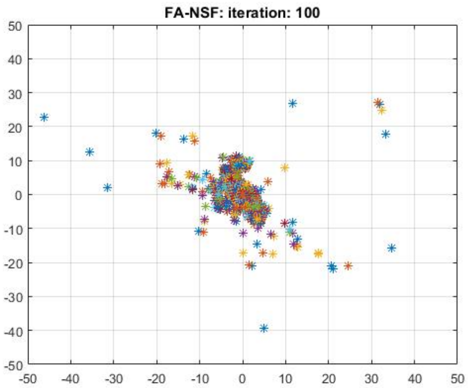
The adaptive step measurements are intended to make the algorithm balanced between exploration and exploitation and improve the algorithm whenever the search condition is in high dimension, very complex and large space. By taking all the considerations and the proposed changes, the original FA is improved and can be presented in the following pseudo-code of Algorithm 1;



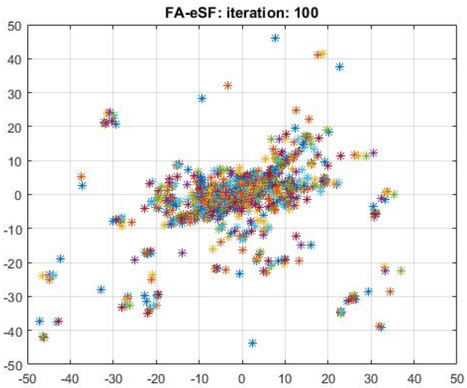
(a) Griewank in 2-D view



(b) FA

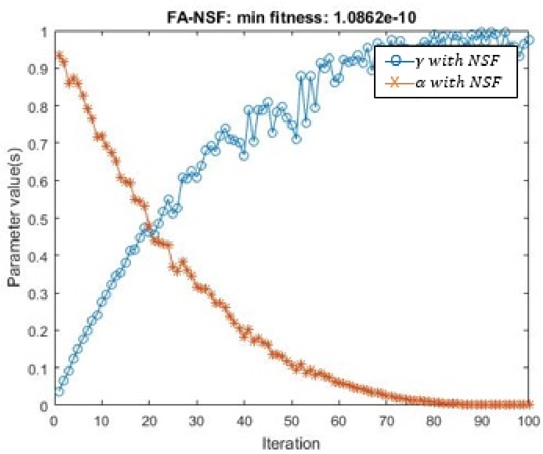


(c) FA-NSF

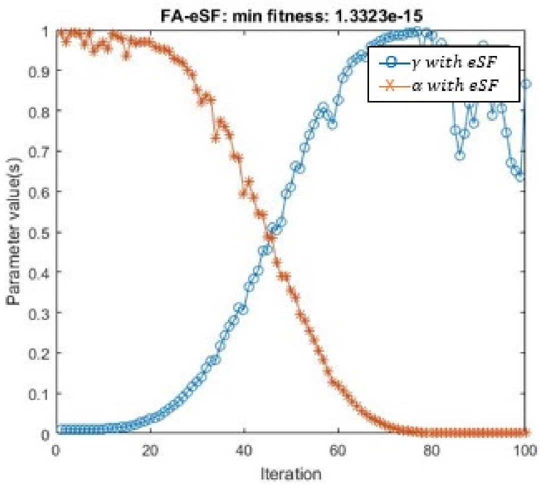


(d) FA-eSF

Figure 3.2: The fireflies movement toward global optimum [0, 0]



(a) FA-NSF



(b) FA-eSF

Figure 3.3: The rate of change of parameters per iteration.

Algorithm 1 Pseudo code of the FA-NSF and FA-eSF algorithm

Input: Objective function of $f(x_d)$, Pre-determined parameter; β_0, γ, α, variable boundary and population size N.
Output: Global minimum, elapsed time.
Generate initial population $x_i, (i = 1, \cdots, n)$ randomly, determine the light intensity, $I(x_d)$ based on individual fitness, $f(x_i)$.
while t, current iteration $t \leq$ maximum iteration **do**
  Determine the value of adaptive parameter (γ_{SF}, α_{SF});
  FA-NSF, referred as (γ_{NSF}, α_{NSF});
  FA-eSF, referred as (γ_{eSF}, α_{eSF});
  for all i to n **do**
    for all j to n **do**
      Evaluate the distance, r between two units (x_i, x_j)
      Evaluate the attractiveness, $e^{-\gamma r^2}$
      if $I_j > I_i$), move i towards j **then**
        Update value of γ_{SF}, α_{SF};
        Evaluate new solution x_{i+1};
      end if
    end for
  end for
  if x_{i+1} exceeds boundary **then**
    Set to its boundary
  end if
  Update light intensity, $I(x_d)$ based on the update location;
  Rank the fireflies and find the current best;
  Export global minimum and elapsed time;
end while

3.3 Adaptive Mechanism for Invasive Weed Algorithm

Based on the introduction of IWO in the previous section, the modification of the metaheuristic algorithm is described in this section. The modification to the IWO algorithm is aimed at achieving a more robust optimisation technique, especially to compensate for deficiencies in the original algorithm. In the original IWO, each weed updates its position through the process of reproduction, elimination and spatial dispersion which are the key steps in IWO. The modified algorithm is expected to provide better balance between global and local search as well as achieve in more accurate result during the iteration process.

In the original IWO, the spatial dispersion uses nonlinear decrement equation of its standard deviation, SD of seeds spreading. Hence, the proposed algorithms use an exponential decrement equation that is aimed at improving the algorithm. The variation in the SD is made exponential in the spatial dispersion process. The new SD is given as

$$\sigma_{iterSD} = \left(exp \left[\tau \left[\frac{iter_i}{iter_{max}} \right]^m \right] \right) (\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.9}$$

where the values of $\sigma_i ni$, σ_f, $iter_{max}$ are as described in the initial SD and maximum iteration parameters setting of the algorithm. The values of τ and δ are pre-set to determine the shape of the exponential slope changes of the SD during the iteration process. It is assumed that τ = 2 and m = 4, which are found as competitive values for the proposed IWO algorithm.

In the original algorithm, it does not use any local knowledge or memory to help seeds to spread over the search space. As a result, many researchers have found that the search accuracy is low and the weed gets stuck in local optima if converged too early (Yin et al., 2012, Gandomi et al., 2013). Yin et al. (2012) stressed that the drawbacks of IWO are specifically low solution precision, getting stuck in local optima and premature convergence. Hence, in order to overcome the shortcomings of IWO, local search knowledge is integrated in the proposed IWO algorithm. This strategy could enable the algorithm to escape from local optima; therefore, the improved version of IWO algorithm has a lesser chance of pre-mature convergence compared to IWO.

A normalized factor to adjust the standard deviation of the seed spreading is proposed to improve its convergence without any major changes in its original structure or additional requirements in the number of evaluations. Two novel variants of the original IWO are proposed, namely IWO-eSSF which employs a seeds spread factor (SSF) on its original standard deviation and MIWO-eSSF with exponential SSF variation in the spatial dispersion process.

3.3.1 IWO with Exponential Seeds-spread Factor

In this new variant of IWO, a simple factor is implemented as the number of iterations increase. This simple factor is called the rate of seeds-spreading evolution factor (SSF). It involves the use of local knowledge in previous iterations to improve the SD property. The mathematical representation of the SSF factor is given as follows:

$$k = \left[\frac{|f_i| - f_{worst}}{f_{best} - f_{worst}} \right] (k'_i - k'_f) + k'_f \tag{3.10}$$

where k is the normalized spread factor of SSF; f_i is the fitness value of selected current weed; f_{worst} and f_{best} are the least and best fitnesses in the current iteration, respectively; k'_i and k'_f are the initial and final values of the seeds-spread factor for the process. The rate of seeds-spreading evolution factor changes according to the values obtained during the iteration process. It will quickly respond to the reaction speed of the seeds during the evolution period to search for better value. Under the assumption and definition above, it can be shown that $0 < k < 1$. This parameter takes the run history of value of each weed into account, and reflects the capability of speed of spreading evolution of each seed, that is, the smaller the value of k, the faster the speed. The normalized SSF mechanism could also be explained as illustrated in Figure 3.1 by using the weed and its fitness value instead of using firefly and its fitness value.

Therefore, the new variants of IWO are proposed by using a factor that benefits from local knowledge after each iteration. In the IWO-eSSF algorithm, the SD in equation (3.9) is improved by incorporating an SSF factor, in equation (3.10) into the SD as

$$\sigma_{iter eSSF} = k \left(exp \left[\tau \left[\frac{iter_i}{iter_{max}} \right]^m \right] \right) (\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.11}$$

where $\sigma_{iter eSSF}$ is the new formulated SD of IWO-eSSF and k is the SSF value. This factor could justify whether the decrement of exponential SD could be adjusted. As the iteration process proceeds, the SD is decreased in an exponential manner. By adding the factor, if the current best and best value fitness show large difference, the factor could be high and help the selection of SD to become large and vice versa. The addition factor could help diversification and intensification of the algorithm. It also helps the algorithm to achieve robust and accurate values.

3.3.2 Modified IWO with Exponential Seeds-spread Factor

In this proposed algorithm, an exponential distribution factor is adopted. The factor k of SSF mechanism in equation 3.10 is modified in order to control changes of SD;

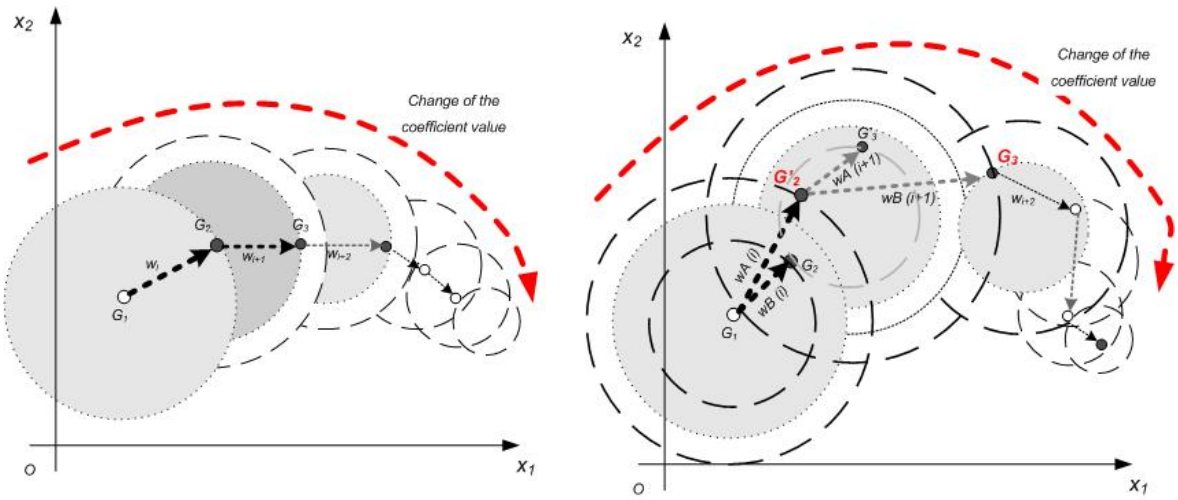
$$k_{new} = \left[\frac{1}{exp\left(\frac{|f_i| - f_{worst}}{f_{best} - f_{worst}}\right)} \right] (k'_i - k'_f) + k'_f \tag{3.12}$$

where k_{new} is the modified SSF; f_i is the fitness value of selected current weed; f_{worst} and f_{best} are the least and best fitness in the current iteration, respectively; k'_i and k'_f are the initial and final values of the seeds-spread factor for the process. Using equation (3.12), the SD will vary in the range $[k'_f, k'_i]$ at each iteration. In the MIWO-eSSF algorithm, the SD in equation (3.9) is improved by incorporating an improved factor into the SD as

$$\sigma_{iter\,MIWO} = k_{new}\left(exp\left[\tau\left[\frac{iter_i}{iter_{max}}\right]^m\right]\right)(\sigma_{ini}-\sigma_f)+\sigma_f \tag{3.13}$$

where $\sigma_{iter\,MIWO}$ is the enhanced SD of MIWO-eSSF and k_{new} is the improved SSF value. In this manner, the exploration ability of weeds located closer to the best weed increases and allows searching around the optimum solution.

Figure 3.4 illustrates how the weeds search for optimum value in the iteration process of IWO and proposed IWO algorithms. Figure 3.4a shows the original movement based on IWO algorithm. The round shape decreases nonlinearly during the search process and the radius of the round shape, w represent the original SD. However, for the improve IWO-eSSF and MIWO-eSSF, as shown in 3.4b the SD will vary as either $wA(i)$ or $wB(i)$ and also as either $wA(i+1)$ or $wB(i+1)$. The value of SD depends on the value of factor k for IWO-eSSF and k_{new} for MIWO-eSSF.



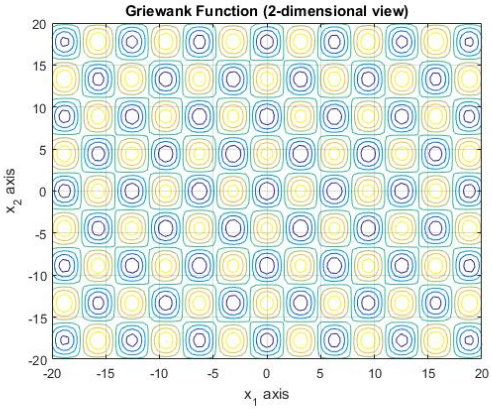
(a) Nonlinear variation in the original algorithm

(b) Adaptive changes of the weeds

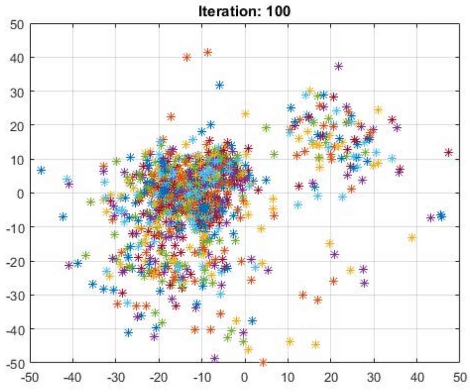
Figure 3.4: The movement of weeds during iteration process.

As a further example, a simple comparison is made based on the proposed algorithms and the original algorithm by running them on two dimensional Griewank function in the range [-30, 30]. This well-known function is an example of single optimisation problem with many local optima as shown in Figure 3.5a.

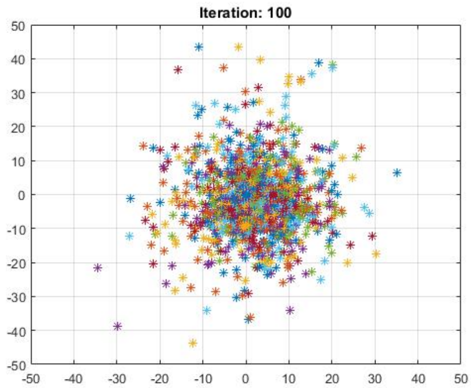
By using the same predetermined parameters and 100 iterations, Figure 3.5 shows the end result of the three algorithms showing the movement of weeds during the optimisation process. In Figure 3.5b, the weeds based on the original algorithm got stuck at a local optimum value and the weeds spread around it. Meanwhile, as shown in Figures 3.5c and 3.5e, the IWO-eSSF and MIWO-eSSF managed to jump out of local optima and concentrated on the global optimum area to produce better value. Figure 3.6 shows the rate of change of the parameters for both proposed algorithms, IWO-eSSF and MIWO-eSSF.



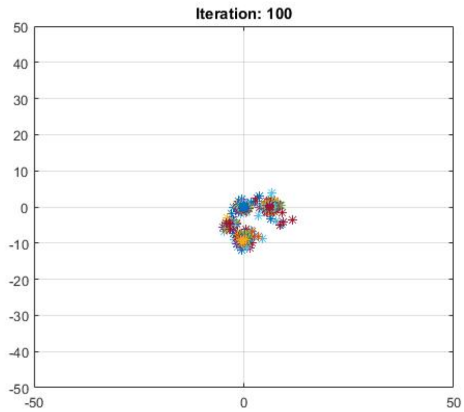
(a) Griewank in 2-D view



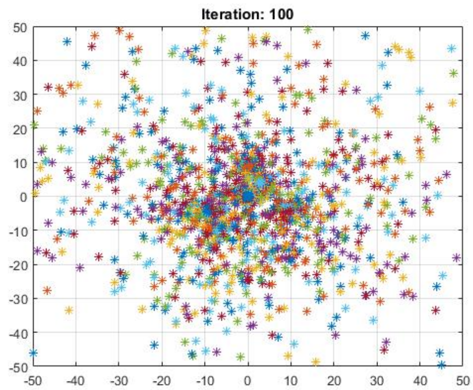
(b) IWO



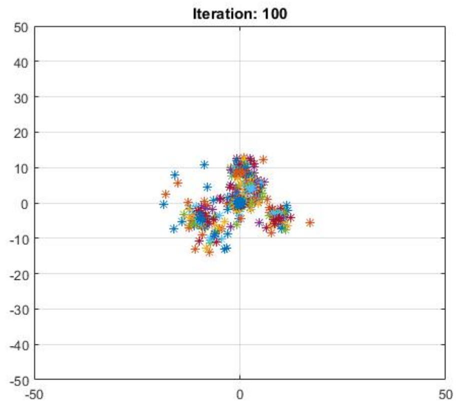
(c) IWO-eSSF



(d) IWO-eSSF (viewed after 70 iterations)



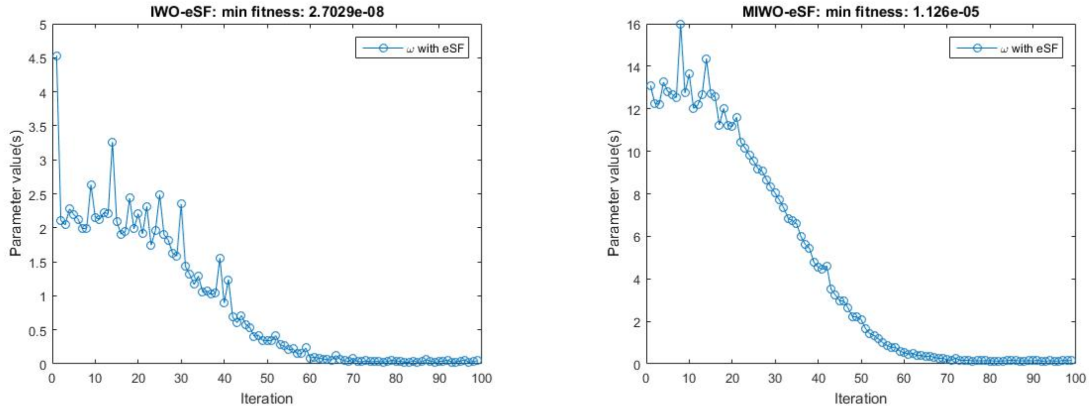
(e) MIWO-eSSF



(f) MIWO-eSSF (viewed after 70 iterations)

Figure 3.5: The weeds movement toward global optimum [0, 0]

IWO-eSSF and MIWO-eSSF algorithms utilized a new strategy in the spatial dispersion of IWO to exploit the weed population and therefore, it can overcome the lack of exploration and improve the solution precision of the IWO. The strategy helps the spatial dispersion process in the algorithm to improve the population diversity to avoid premature convergence and make the algorithm more robust. The strategy could improve the capability of optimisation



(a) IWO-eSSF

(b) MIWO-eSSF

Figure 3.6: The rate of change of the SD parameters per iteration.

procedure by updating the solution to accelerate the convergence speed for more accurate fitness values with less computational time. Algorithm 2 shows the pseudo-code of the proposed IWO variants:

Algorithm 2 Pseudo code of IWO-eSSF and MIWO-eSSF algorithms

Input: Objective function of $f(x_d)$, Pre-determined parameter; s_{min}, s_{max}, σ_{min}, σ_{max}, variable boundary and population size n.
Output: Global minimum, elapsed time.
Generate initial population $x_i, (i = 1, \cdots, n)$ randomly,
Determine the initial population by random search, $f(x_d)$,
Rank the initial weeds and calculate the number of seeds for each plant,
while t current iteration $t \leq$ maximum iteration **do**
  Determine the value of normalized factor, k,
  Update the value of SD_{SSF},
  Spatial dispersion
  Distribute seeds based on SD_{SSF} and generate seeds over the search space,
  Competitive exclusion
  if the number of weeds and seeds $>$ maximum population **then**
    Eliminate the plant;
  end if
  if x_{i+1} exceeds boundary **then**
    Set to its boundary
  end if
  Find the current best individual and its fitness, $f(x_i)$;
  Rank the weeds based on their fitnesses, $f(x)$ and determine the number of seeds produced by each weed;
  Export global minimum and elapsed time;
end while

3.4 Hybrid Strategies of Firefly and Invasive Weed Optimisation

Instead of improving the algorithm, combining two or more algorithms is another alternative to compliment one another for a better optimisation algorithm. The combination strategy is simply called hybridization method, which refers to the process of combining the best features of two or more algorithms to form a new optimisation algorithm. The resulting hybridization algorithm is expected to outperform its predecessor algorithms over general problems or application specific problems. Therefore, in this section, new hybridization algorithms by using FA are developed to achieve improved performance in terms of search capabilities and better accuracy.

Based on the introduction of IWO and FA in the previous section, the combination of the two approaches is described in this section. The FA is effective in local search, but can easily get trapped in local optima. The IWO algorithm, on the other hand, is effective in accurate global search. Therefore, the idea of hybridization between IWO and FA is to achieve a more robust optimisation technique, especially to compensate for the deficiencies of the individual algorithms. The biggest advantage of IWO algorithm is its capability of global exploration and diverse search. In the proposed algorithm, the firefly method is embedded into IWO to enhance the local search capability of IWO algorithm that already has very good exploration capability.

The strategy utilizes the spatial dispersion of IWO and firefly movement to explore new areas in the search space and exploit the population, respectively. Therefore, it can overcome the lack of exploration of the original FA and improve the low solution precision of the IWO. In other words, hybridization not only improves the performance, it also improves the accuracy of the constituent algorithms. This combination improves the capability of optimisation procedure by updating the solution to accelerate the convergence for more accurate fitness values with less computational time.

3.4.1 Hybrid Invasive Weed-Firefly Optimisation

In this work, a hybrid algorithm is proposed by inducing FA into IWO referred to as hybrid invasive weed firefly optimisation (HIWFO) algorithm. In HIWFO algorithm, the initialization of both FA and IWO is done by pre-determining the initial parameters. Table 3.1 shows description of the parameters used in the HIWFO algorithm.

During the initialization process, diversification strategy of IWO algorithm is used. Here, a random dispersion of the initial population takes place. In this section, an objective function is set up and determined. Each population produces new seed(s) according to their fitness level. Another stage of diversification of IWO is carried out, by dispersal of the seeds randomly based on the number of seeds given to each population. The range of distribution or

Table 3.1: Parameters used in HIWFO algorithm

Parameters used	**Symbol**
Initial population size	n_{ini}
Maximum population size	n
Minimum number of seeds	s_{min}
Maximum number of seeds	s_{max}
Initial value of standard deviation, SD	σ_{ini}
Final value of SD	σ_f
Attractiveness coefficient	γ
Randomization coefficient	α
Attractiveness coefficient	β

the standard deviation of dispersal is based on the SD value. Early stage of iteration has a large SD value as it decreases nonlinearly. This spatial dispersion process in the algorithm strives to improve the population diversity to avoid premature convergence and makes the algorithm more robust.

After spatial dispersion, the process of competitive exclusion process is performed. The process continues using the FA movement. At this stage, the cooperation of FA is done by trying to improve the position so that the current population can move towards the best individual in the current iteration. Hence, the enhanced algorithm not only ensure individual diversity by IWO, but also improves the optimisation accuracy and the speed of convergence of the algorithm.

The boundary re-adjustment scheme is placed after the movement process at the end of the iteration to ensure the population is within the search space. The action also helps each member of the population to stay within the boundary and ready for the next iteration. Therefore, the steps of the proposed HIWFO algorithm are best described as follows:

(Step 1) Initialization

[Sub-step a] Initialize the parameters of invasive weed and firefly algorithm, the dimension and boundary limits of the search space.

[Sub-step b] Initialize the population of the hybrid algorithm. A population of initial seeds of plant is dispersed over a search space randomly. By using the designated objective function, each seed's fitness value could be calculated based on its initial position.

(Step 2) Update the following parameters:

[Sub-step a] The production and distribution of weed(s) by plant. Each plant produces seeds and this increases linearly from minimum to its maximum possible seeds production.

$$weed_{x_i} = \left[\frac{f_{x_i} - f_{min}}{f_{max} - f_{min}}\right](s_{max} - s_{min}) + s_{min} \tag{3.14}$$

where f_{x_i} is the weed's fitness at current population, f_{max} is the maximum fitness of the current population, f_{min} is the minimum fitness of the same population, s_{max} and s_{min} respectively represent the maximum and the minimum values of a seed.

[Sub-step b] The parameter of light absorption coefficient, γ, attraction coefficient, β and randomization coefficient, α remain constant as suggested by Yang (2009).

(Step 3) Reproduction loop: Iteration = iteration + 1

Each seed grows into plant in the population capable of reproducing seeds but according to its fitness, where the fitter plants produce more seeds.

(Step 4) Spatial dispersion

The seeds generation is randomly distributed in the search area according to normal distribution with zero mean and SD. The normalized SD per iteration, σ_{iterSD} is given as

$$\sigma_{iterSD} = \left[\frac{iter_{max} - iter}{iter_{max}}\right]^n (\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.15}$$

where σ_{iterSD} is the coefficient value at the iteration during the search process with σ_{ini} as its initialized value and σ_f as the final coefficient value. $iter_{max}$ is the maximum iteration value and $iter_i$ is the present value of the iteration.

(Step 5) Competitive exclusion

The population of plants is controlled by the fitness of the plants. If the population has reached its maximum size, the elimination process runs on the poor fitness plants where only plants with better fitness are allowed to survive. This elimination process or competitive exclusion is employed from generation to generation until it reaches its maximum number of generations / iterations of the algorithm. At the end of the algorithm, the seeds and their respective parents are ranked together and have chance to grow in the search area and reproduce seeds as mentioned in Step (2). Those with better fitness produce more seeds and have high possibility of survival and become reproductive. The processes continue until the maximum number of iterations is reached and the plant with best fitness is expectedly closest to the optimum solution.

(Step 6) Improve the local search by firefly localization

[Sub-step a] The fitness value of each plant is equal to the light intensity of the firefly algorithm. Therefore, the firefly algorithm's mechanism is started.

[Sub-step b] The position of the plant, $x_{(i+1)}$is updated using equation (3.1) in a highly random manner. The plant with lower fitness value essentially has low light intensity, and will approach and move towards higher light intensity.

(Step 7) Boundary checking mechanism

With the random movement in Step 6 members of the population will have tendency to move beyond the boundary. The boundary checking mechanism is used to avoid any member of the population jump out of the boundary of the problem.

(Step 8) The result of the algorithm for the iteration is updated and if the maximum number of iterations has not been reached, the next generation of the plant starts in the loop.

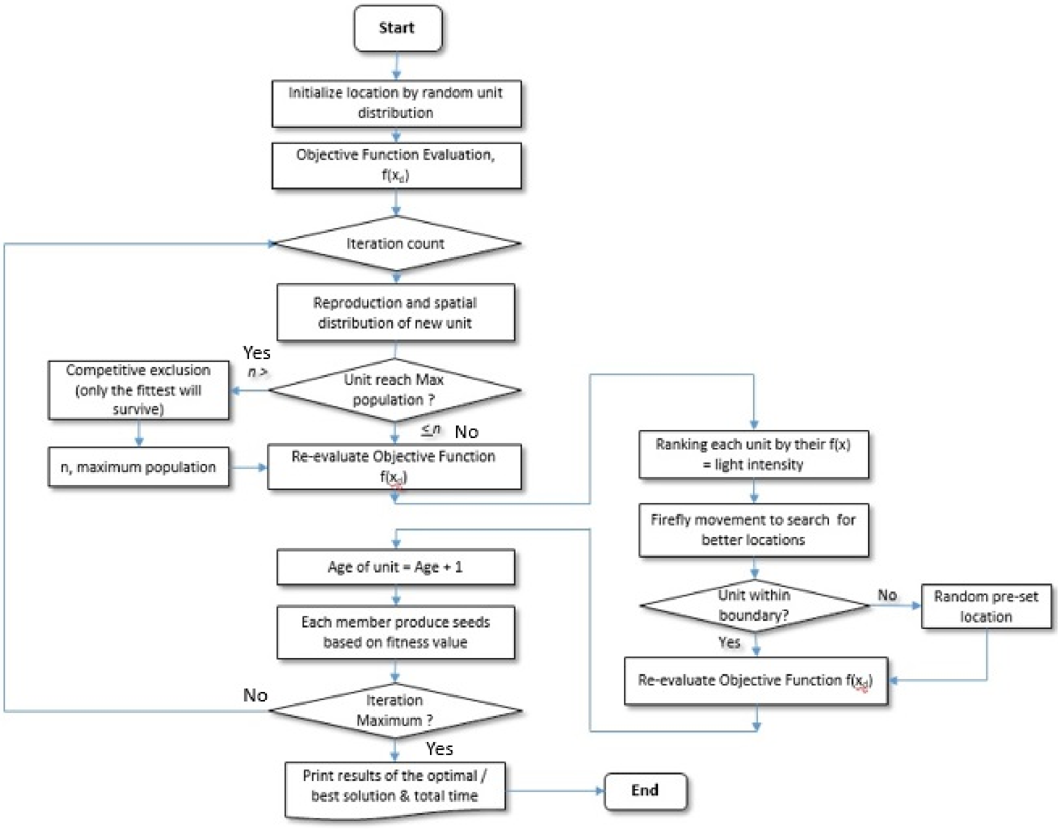


Figure 3.7: The flow-chart of HIWFO algorithm.

Figure 3.7 shows the flow chart of HIWFO algorithm. Each and every step of the algorithm is illustrated in the figure and as explained in the step stage above. The proposed HIWFO approach can be summarized in pseudo code as in Algorithm 3.

Algorithm 3 Pseudo code of HIWFO algorithm

Input: Objective function of $f(x_d)$, Pre-determined parameters of IWO and FA.
Output: Global minimum, elapsed time.
Random initial population $x_i, (i = 1, \cdots, n)$, evaluate fitness $f(x_i)$, rank based on its fitness and number of seeds produced.
while t current iteration $t \leq$ maximum iteration **do**
  Competitive exclusion
  Distribute seeds based on SD_{new} and generate seeds over the search space,
  if the number of weeds and seeds $>$ maximum population **then**
    Eliminate the plant
  end if
  Improve the locations by using firefly localization
  for all i to n **do**
    for all j to n **do**
      Evaluate r between two units (x_i, x_j) and their attractiveness via $e^{-\gamma r^2}$
      if $I_j > I_i$), move i towards j **then**
        Evaluate new solution x_{i+1};
      end if
    end for
  end for
  if x_{i+1} exceeds boundary **then**
    Set to its boundary
  end if
  Update light intensity, $I(x_d)$ based on the updated location;
  Export global minimum and elapsed time;
end while

3.4.2 Hybrid Invasive Weed-Firefly Optimisation with Spread Factor

Another hybridization algorithm based on FA and IWO proposed is named hybrid invasive weed firefly optimisation with spread factor (HIWFO-SF) algorithm. In HIWFO-SF algorithm, the same framework of HIWFO is used. However, some modifications are made on the parameters of the algorithm. Table 3.2 shows the parameters used in the HIWFO-SF algorithm.

Table 3.2: Parameters used in HIWFO-SF algorithm

Parameters used	**Symbol**
Initial population size	n_{ini}
Maximum population size	n
Minimum number of seeds	s_{min}
Maximum number of seeds	s_{max}
Initial value of standard deviation, SD	σ_{ini}
Final value of SD	σ_f
Attractiveness coefficient	γ
Randomization coefficient	α
Randomization coefficient	α
Attractiveness coefficient	β

The modification is made at the spatial dispersion section of HIWFO-SF. At this stage, as stated in equation (3.10), normalized factor of the rate of seeds-spreading evolution factor (SSF) is introduced. It involves the use of local knowledge in previous iterations to improve the SD property. The SD is changed nonlinearly, hence the mathematical representation of a new equation of SD with SSF, referred to as SF_1 is given as

$$SF_1 = k_{SF}\left[\frac{iter_{max} - iter}{iter_{max}}\right]^n (\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.16}$$

where SF_1 is the new coefficient value of SD at the present iteration during the search process with σ_{ini} as its initialized SD value and σ_f as the final SD value. k is the normalized SF factor stated in equation 3.10, $iter_{max}$ is the maximum iteration value and $iter_i$ is the present value of the iteration. The SF_1 value is nonlinearly decreased and aimed to further improve the diversification process of the algorithm.

In the firefly section, randomization parameter is adaptively changed in the search process. In this work, randomization coefficient (α) is decreased exponentially with normalized factor of SF introduced, called α_{SF}. This new coefficient is given as

$$\alpha_{SF} = k_{SF}\left(exp\left[\tau\left[\frac{iter_i}{iter_{max}}\right]^m\right]\right)(\sigma_{ini} - \sigma_f) + \sigma_f \tag{3.17}$$

where α_{SF} is the randomization value at the iteration during the search process with α_{ini} as its initialized value and α_f as the final coefficient value. $iter_{max}$ is the maximum iteration value and $iter_i$ is the present value of the iteration.

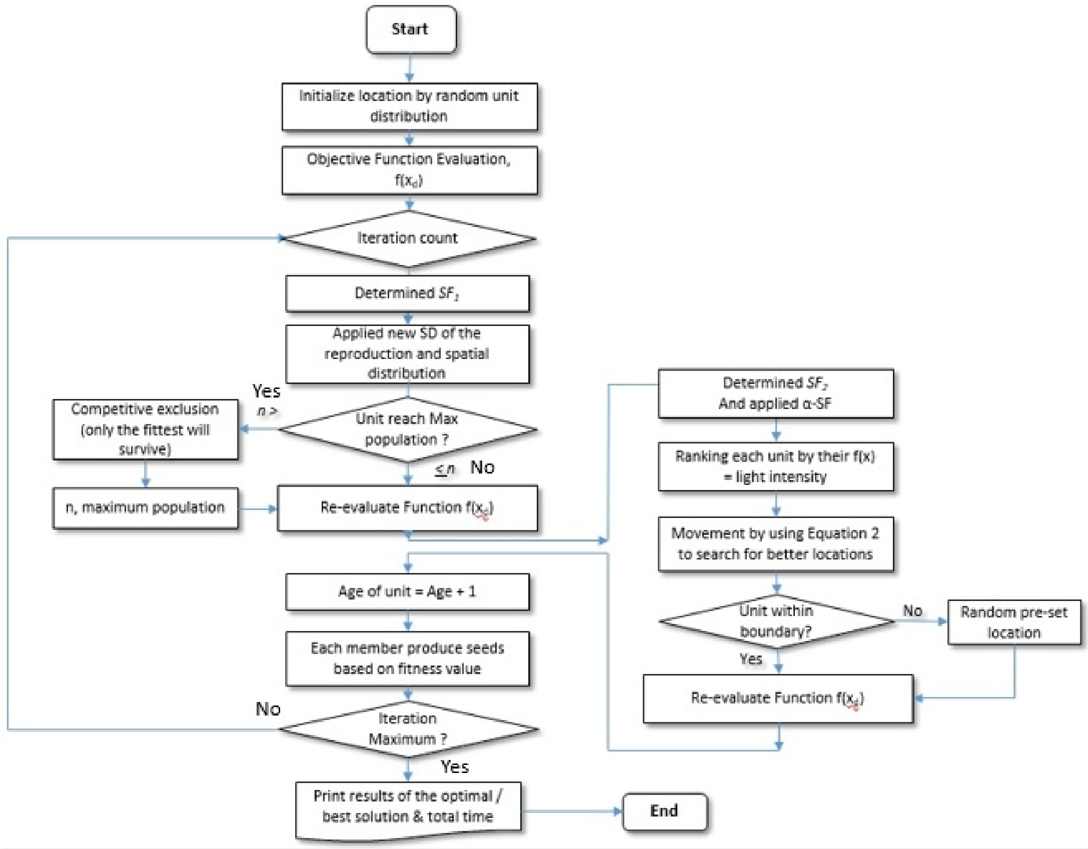


Figure 3.8: The flow-chart of HIWFO-SF algorithm.

The adaptive coefficient factor, α_{SF} could justify whether the exponential decrement of randomization value could be adjusted. As the iteration process proceeds, the coefficient decreases in an exponential manner. By adding the factor, if the current best and best value fitness show large difference, the factor could be high and help the random value to become large and vice versa. The additional factor could help diversification and intensification of the algorithm. It also helps the algorithm to achieve more robust and accurate values.

Figure 3.8 shows the flow chart of HIWFO-SF algorithm. The main steps of the proposed HIWFO-SF approach can be summarized in pseudo code as in Algorithm 4.

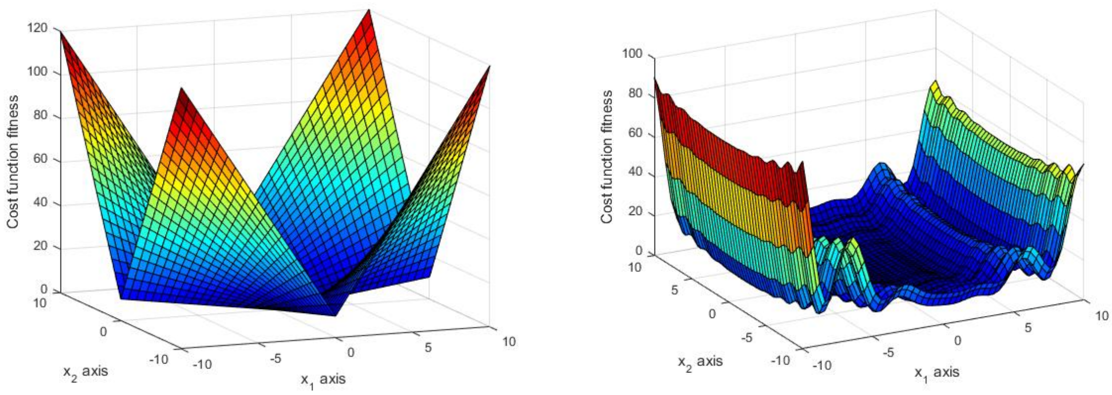
Algorithm 4 Pseudo code of HIWFO-SF algorithm

Input: Objective function of $f(x_d)$, Pre-determined parameters of IWO and FA.
Output: Global minimum, elapsed time.
Random initial population $x_i, (i = 1, \cdots, n)$, evaluate fitness $f(x_i)$, rank based on its fitness and number of seeds produced.
while t current iteration $t \leq$ maximum iteration **do**
  Determine the value of normalized factor, k,
  Determine the value of adaptive parameter, α_{SF};
  Update the value of SD_{SSF},
  Competitive exclusion
  Distribute seeds based on SD_{SSF} and generate seeds over the search space,
  if the number of weeds and seeds $>$ maximum population **then**
    Eliminate the plant
  end if
  Improve the locations by using firefly localization
  for all i to n **do**
    for all j to n **do**
      Evaluate r between two units (x_i, x_j) and their attractiveness via $e^{-\gamma r^2}$
      if $I_j > I_i$), move i towards j **then**
        Update value of α_{SF};
        Evaluate new solution x_{i+1};
      end if
    end for
  end for
  if x_{i+1} exceeds boundary **then**
    Set to its boundary
  end if
  Update light intensity, $I(x_d)$ based on the update location;
  Export global minimum and elapsed time;
end while

3.5 Parameters and Their Impact on Accuracy and Convergence

In this section, the effect of tuning and selecting the population and iteration parameters of the algorithms used are studied. The impact on the accuracy of the optimal solution and convergence characteristics is observed and investigated. This is because, in solving any optimisation problem, these computational parameters of bio-inspired algorithms have to be determined and chosen properly in order to get the optimum results. The simulation in this section compares between all the proposed algorithms and their predecessors. The study is concerned with the impact of the size of population and number of iterations in different problem dimensions.

The benchmark functions used in this study consist of two unconstrained single optimisation problems. The functions constitute two type of optimisation problems, a single global optimum as well as local optima, comprising unimodal and multimodal types as shown in Figure 3.9.



(a) Schwefel's Problem 2.22 (b) Levy function

Figure 3.9: Benchmark functions used in the study

The experimental testing platform is implemented on a personal computer (PC) with processor CPU Intel (R) Core (TM) i5-2400 with Windows 7 Professional operating system, frequency of 3.10 GHz and memory installed of 4.00 GB RAM. The program is coded in MATLAB R2013a. For fair comparison of all the algorithms used, most of the parameters are set identical. Both benchmark function have a global optimum. Table 3.3 shows the parameter set of all the algorithms during initialization.

The performances of the algorithms are assessed with well-known benchmark functions as shown in Table 3.4. The Schwefel's Problem 2.22 represents unimodal function and Levy function is used to test the algorithms with multimodal function. Both benchmark functions have the same global optimum which is $F(x_i^*) = 0$ at $x_i^* = (0, \ldots, 0)$ where $i = 1, \ldots, D$ and D is the problem dimension.

In solving most optimisation problem with FA, Yang (2010a) advised that the number of

Table 3.3: Initial parameters of the algorithms used in the study

Parameters	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
s_{min}	-	-	-	0	0	0	0	0
s_{max}	-	-	-	5	5	5	5	5
σ_{ini}	-	-	-	5	5	5	5	5
σ_f	-	-	-	0.01	0.01	0.01	0.01	0.01
β_0	1	1	1	-	-	-	1	1
α_{ini}	0.2	1	1	-	-	-	0.2	1
α_f	-	0.001	0.001	-	-	-	-	0.001
γ_{ini}	1	0.001	0.001	-	-	-	1	1
γ_f	-	1	1	-	-	-	-	-

Table 3.4: Benchmark functions used in the study

Function	Formulation	Range
Schwefel's Problem 2.22	$f_a(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$
Levy function	$f_b(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{D-1} (\omega_i - 1)^2 [1 + 10\sin^2(\pi\omega_i + 1)]$ $+(\omega_D - 1)^2[1 + \sin^2(2\pi\omega_D)];$ $where\ \omega_i = 1 + \frac{x_i - 1}{4},\ for\ all\ i = 1, ..., D$	$[-10, 10]^D$

fireflies, n, is sufficient if the value is within between 15 to 50. For more complex problems, unless there is no other alternative, large n could be considered (Yang, 2010a). However, with excessively large n, extensive computational burden should be expected. The parameters chosen for the study are tabulated in Table 3.5:

Table 3.5: The parameters to be studied for all the algorithm

No.	Parameter	Symbol	Value
1	Size of population	n	6, 30 and 100
2	Number of iteration	it_{max}	50 and 500
3	Number of problem dimension	Dim	2 and 30

As the size of population and number of iterations are fixed at certain value, the number of function evaluations (NFE) can be defined as;

$$NFE = n \times it_{max} \tag{3.18}$$

where n is the size of population and it_{max} is the maximum iteration during optimisation process. The performance results are specified by the best solution value obtained and the time taken after allowable it_{max} was reached. In each simulation, the value it_{max} signifies the stopping criterion.

3.5.1 Unimodal Function

In this section, studies are conducted for understanding the effects of tuning parameters of improved FA, IWO and the proposed hybrid algorithms on convergence and solution accuracy for unimodal functions. The simulations with Schwefel's Problem 2.22, $f_a(x)$ are carried out using the proposed algorithms, FA-NSF, FA-eSF, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF. The results are compared and conclusions drawn on the best parameter condition for the algorithm.

Table 3.6: Results of FA variants for Schwefel's Problem 2.22 test

			FA		FA-NSF		FA-eSF	
n	Dim	it_{max}	$f(x)$	$t,(sec)$	$f(x)$	$t,(sec)$	$f(x)$	$t,(sec)$
6	2	50	1.85E+00	0.034	1.35E-04	0.012	**4.03E-05**	**0.011**
		500	3.86E-01	0.065	1.72E-07	**0.064**	**1.52E-08**	0.085
	30	50	8.83E+11	0.063	1.15E-02	**0.011**	**9.68E-03**	0.014
		500	3.06E+09	0.061	9.97E-06	**0.092**	**5.82E-06**	0.099
30	2	50	4.36E-01	**0.17**	**2.03E-05**	0.189	5.80E-05	0.225
		500	7.31E-02	**1.182**	2.33E-08	1.829	**1.44E-08**	1.963
	30	50	2.54E+09	**0.216**	7.62E-03	0.234	**3.86E-03**	0.3
		500	1.68E+07	**1.437**	1.22E-05	1.993	5.40E-06	1.558
100	2	50	2.77E-01	**1.541**	5.25E-05	1.827	**4.92E-05**	1.598
		500	5.48E-02	**12.983**	**7.84E-09**	19.835	1.01E-08	13.719
	30	50	1.05E+09	**1.253**	9.64E-03	3.077	**2.44E-03**	1.65
		500	1.21E+07	19.876	1.20E-05	**17.167**	**6.29E-06**	21.319

Table 3.6 shows the numerical simulation results for original FA and two proposed FA variants, FA-NSF and FA-eSF for unimodal problem. Classified by the size of population as 5, 30 and 100, the FA-eSF showed better solution accuracy. The results showed that the use of population size of 100, only slightly improvement as compared with population size of 30. However, the computational burden is clearly shown using high population size especially when extending the iteration to 500 and increasing the problem dimension to 30. For all algorithms, using low size in population, did not helped the algorithm in getting better optimal solution.

The numerical simulation results for original IWO and the two proposed IWO variants, IWO-eSSF and MIWO-eSSF for Schwefel's Problem 2.22 are shown in Table 3.7. The simulated results are tabulated based on the size of population, different dimensions and iterations. Most of the results showed that the proposed IWO variants showed better solution accuracy. The results also showed that while the higher size of population has a significant improvement to the solution quality, but it increased the time taken to solve the problem. This computational burden became heavy especially when using a larger number of iterations and larger dimension. The use of small population in this problem, already showed a good result, however, not as competitive as population sizes 30 and 100.

Figure 3.8 shows the simulation results of HIWFO and HIWFO-SF for unimodal op-

Table 3.7: Results for IWO variants for Schwefel's Problem 2.22 test

			IWO		IWO-eSSF		MIWO-eSSF	
n	Dim	it_{max}	$f(x)$	$t, (sec)$	$f(x)$	$t, (sec)$	$f(x)$	$t, (sec)$
6	2	50	5.93E-03	0.021	**8.66E-04**	**0.017**	3.36E-03	0.029
		500	1.45E-03	0.206	**1.05E-03**	0.204	1.10E-03	**0.123**
	30	50	2.60E+06	0.022	5.82E+01	**0.021**	**4.78E+01**	0.033
		500	8.70E+05	**0.158**	**2.38E+00**	0.234	3.72E+00	0.2
30	2	50	2.93E-03	0.084	**7.88E-04**	**0.059**	1.15E-03	0.099
		500	6.13E-04	**0.602**	**2.88E-04**	0.657	4.41E-04	0.827
	30	50	1.25E+02	0.125	4.30E+01	**0.09**	**3.76E+01**	0.229
		500	5.21E+01	0.734	**2.44E-01**	**0.719**	6.38E-01	1.2
100	2	50	8.21E-04	0.28	**2.53E-04**	0.323	7.59E-04	**0.185**
		500	2.77E-04	**1.498**	**7.59E-05**	2.097	2.24E-04	2.547
	30	50	9.16E+01	0.382	3.25E+01	0.352	**2.41E+01**	**0.283**
		500	2.02E+01	**2.299**	**1.90E-01**	2.649	4.91E-01	4.198

Table 3.8: Results of HIWFO and HIWFO-SF for Schwefel's Problem 2.22 test

			HIWFO		HIWFO-SF	
n	Dim	it_{max}	$f(x)$	$t, (sec)$	$f(x)$	$t, (sec)$
6	2	50	4.62E-03	**0.045**	**5.74E-04**	0.104
		500	2.13E-03	**0.177**	**6.94E-08**	0.197
	30	50	**1.14E+00**	0.084	1.17E-01	**0.036**
		500	7.05E-01	0.421	**9.72E-03**	**0.252**
30	2	50	2.39E-03	**0.121**	**6.21E-04**	0.278
		500	5.49E-04	0.943	**2.75E-08**	**0.895**
	30	50	1.27E+00	**0.107**	**9.85E-01**	0.154
		500	5.87E-01	**1.365**	**3.13E-01**	0.847
100	2	50	1.76E-03	**0.487**	**5.02E-04**	0.718
		500	6.50E-04	**5.849**	**7.36E-09**	6.416
	30	50	1.19E+00	**0.7**	**8.95E-01**	0.889
		500	3.58E-01	**7.027**	**2.93E-01**	9.849

timisation problem. These newly proposed hybrid algorithms have shown their competitive characteristics with better solution quality as compared to FA (referred to Table 3.6) and IWO (referred to Table 3.7). In most of the cases, HIWFO-SF showed better solution compared to HIWFO. For both algorithms, high population size resulted better solution compared to lower ones. The increment of iteration also helped the algorithm to gain better result. For both algorithms, simulation procedure that used 100 maximum population size showed slightly better than using population size 30, but with higher computational time.

3.5.2 Multimodal Function

The simulations with Levy function, $f_b(x)$ as an example of multimodal function are performed to study the effect of parameters on the convergence and solution accuracy of the proposed algorithms, FA-NSF, FA-eSF, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF. The results are compared and conclusions drawn on best parameter condition for the algorithms.

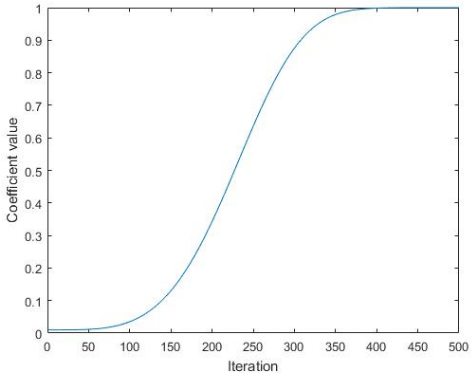
Modification of attractiveness and randomization of both FA-NSF and FA-eSF are tested using multimodal function, Levy function. To illustrate the adaptive change of both coefficient, Figure 3.10 shows the change of attraction and randomization parameters in FA-eSF. The attractive coefficient is increased exponentially whereas the randomization is decreased in exponential shape. Figures 3.10c and 3.10d show the effect of SF on both coefficient during the iteration process.

On the other hand, Figure 3.11 shows an example of the effect of the adaptive mechanism of both improved IWO algorithm, IWO-eSSF and MIWO-eSSF. Both Figures 3.10 and 3.11 are based on simulation of the proposed algorithms on Levy function.

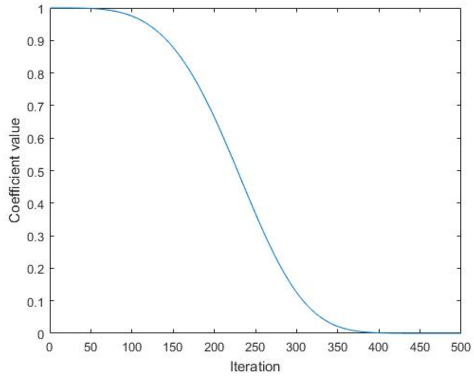
Table 3.9: Results of FA variants for Levy function test

			FA		FA-NSF		FA-eSF	
n	Dim	it_{max}	$f(x)$	$t, (sec)$	$f(x)$	$t, (sec)$	$f(x)$	$t, (sec)$
5	2	50	2.45E-01	0.01	4.09E-02	0.011	**3.76E-03**	**0.01**
		500	2.76E-02	**0.063**	2.41E-02	0.079	**3.39E-03**	0.112
	30	50	2.32E+02	**0.016**	2.99E+00	0.022	**2.77E+00**	0.035
		500	2.15E+02	0.152	2.64E+00	**0.104**	**2.31E+00**	0.15
30	2	50	3.64E-02	**0.133**	**1.23E-03**	0.169	2.50E-03	0.167
		500	7.68E-04	**1.35**	8.05E-04	1.659	**2.33E-04**	1.974
	30	50	2.06E+02	**0.137**	2.65E+00	0.287	**2.59E+00**	0.231
		500	1.56E+02	**1.54**	2.55E+00	2	**2.25E+00**	2.131
100	2	50	4.73E-03	**1.354**	**4.79E-04**	2.117	7.80E-04	1.896
		500	8.84E-04	**10.828**	2.50E-04	19.248	**3.31E-05**	14.805
	30	50	2.22E+02	**1.745**	**2.58E+00**	2.369	2.66E+00	2.088
		500	1.33E+02	**12.867**	**2.34E+00**	20.599	2.46E+00	18.585

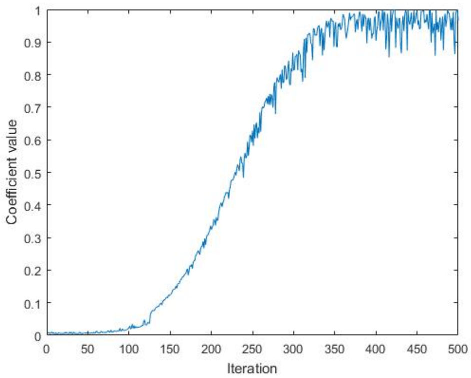
Based on the multimodal problem of Levy functions, the numerical simulation results for original FA and two proposed FA variants, FA-NSF and FA-eSF are shown in Table 3.9.



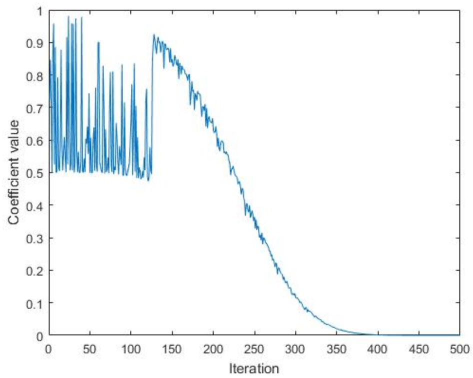
(a) Exponential changes of gamma value



(b) Exponential changes of alpha value

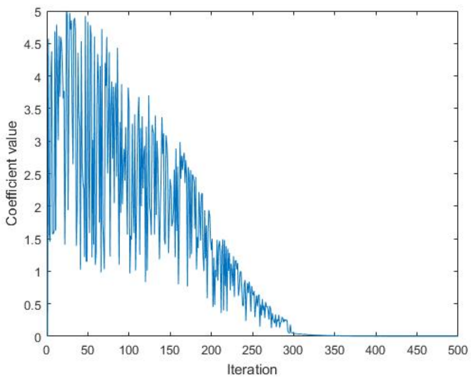


(c) Changes of gamma with SD mechanism

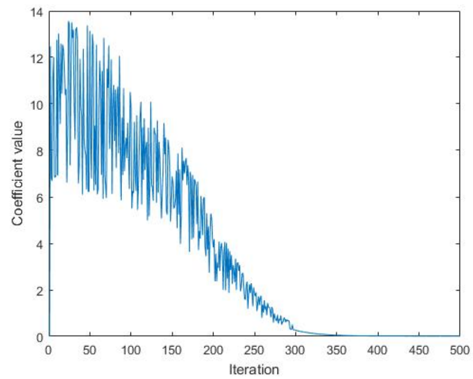


(d) Changes of alpha with SD mechanism

Figure 3.10: Variation of parameters for FA-eSF during iteration process



(a) IWO-eSSF



(b) MIWO-eSSF

Figure 3.11: The rate of change of SD for adaptive mechanism of IWO variants

Classified by three different population sizes; 5, 30 and 100, the FA-eSF showed slightly better solution accuracy than FA-NSF, but both proposed algorithms produced better results than FA. The results show that with higher population size, only slight improvement was observed especially between population sizes 30 and 100. In addition, the computational time clearly increased with population size. For all algorithms, low population size did not help the algorithm to achieve better optimal solution.

Table 3.10: Results of IWO variants for Levy function test

			IWO		IWO-eSSF		MIWO-eSSF	
n	Dim	it_{max}	$f(x)$	$t,(sec)$	$f(x)$	$t,(sec)$	$f(x)$	$t,(sec)$
5	2	50	7.33E-06	0.021	9.74E-07	0.032	**3.32E-07**	0.026
		500	1.25E-06	0.136	3.76E-08	**0.11**	8.19E-07	0.132
	30	50	1.57E+02	0.024	4.88E+01	0.158	**4.38E+01**	0.019
		500	8.34E+01	0.292	1.34E+01	0.185	**7.27E+00**	0.161
30	2	50	9.30E-07	**0.049**	**6.95E-08**	0.114	1.94E-07	0.101
		500	1.46E-07	0.848	4.88E-09	0.691	**3.61E-09**	**0.614**
	30	50	6.73E+01	0.132	2.48E+01	**0.094**	**2.05E+01**	0.178
		500	7.72E+00	1.012	1.91E+00	**0.849**	**1.60E+00**	0.987
100	2	50	1.38E-07	0.331	**2.08E-08**	0.287	2.13E-08	**0.207**
		500	5.42E-08	2.199	**4.85E-11**	2.061	9.50E-09	**1.892**
	30	50	4.07E+01	0.62	1.72E+01	0.484	**1.41E+01**	**0.283**
		500	4.56E-01	3.672	**1.33E-03**	3.113	1.86E-01	3.287

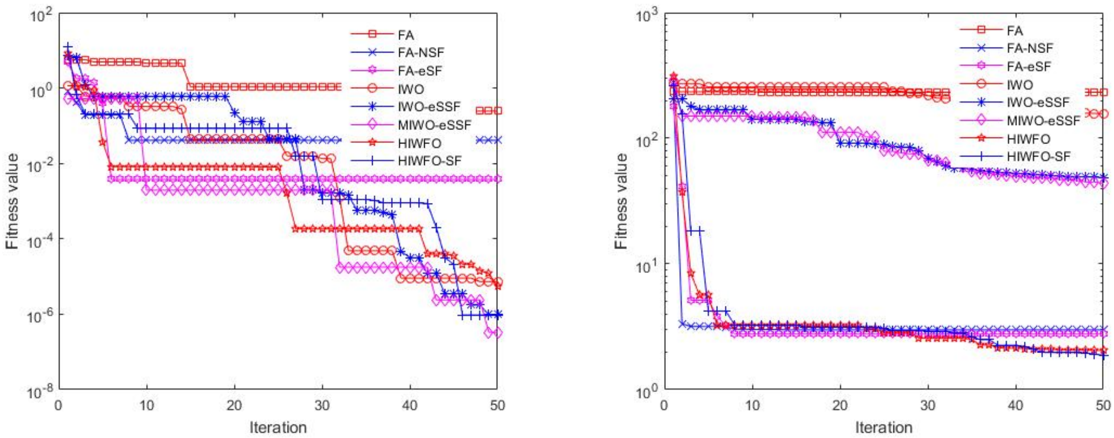
Table 3.10 shows the simulation results of original IWO and the proposed algorithms, IWO-eSSF and MIWO-eSSF for multimodal optimisation problem. The proposed algorithms have shown competitive results with better solution quality as compared to their original predecessor algorithm. There are some mixed performance among both proposed algorithms, but MIWO-eSSF showed a slightly better solution compared to IWO-eSSF for multimodal problem. For all algorithms, higher population size resulted better solution as compared with lower ones. The increment of iterations also helped the algorithm to gain better result. For both algorithms, simulations with maximum population size of 100 showed slightly better result than population size 30. The high computational time and NFE with higher population size is also evident.

Table 3.11: Result of HIWFO and HIWFO-SF for Levy function test

			HIWFO		HIWFO-SF	
n	Dim	it_{max}	$f(x)$	$t,(sec)$	$f(x)$	$t,(sec)$
5	2	50	5.58E-06	0.511	**9.53E-07**	**0.02**
		500	**8.37E-08**	0.244	2.92E-07	**0.185**
	30	50	2.05E+00	**0.028**	1.88E+00	0.11
		500	**5.89E-01**	**0.259**	6.81E-01	0.308
30	2	50	2.35E-07	**0.222**	**2.55E-08**	0.226
		500	2.35E-08	2.113	**2.35E-16**	**0.736**
	30	50	**1.22E+00**	0.264	1.54E+00	**0.133**
		500	1.85E-01	2.512	**1.12E-01**	**1.259**
100	2	50	3.59E-08	1.853	**3.37E-08**	**0.595**
		500	4.19E-09	15.937	**3.49E-17**	**4.717**
	30	50	1.36E+00	2.307	1.36E+00	**0.617**
		500	**9.57E-02**	21.126	9.67E-02	**7.06**

The numerical simulation results for HIWFO and HIWFO-SF algorithms are shown in Table 3.11. Most of the results show that the proposed HIWFO-SF produced better solution accuracy. Both algorithms also gave better solution as compared to their predecessors, IWO and FA. The results also show that higher population size had a significant improved solution quality, but it also increased the time taken to solve the problem. This showed the heavy computational burden especially when using a larger number of iteration and larger dimension. The use of low population size in this problem, already showed a good result, however, not as competitive as with sizes of 30 and 100.

In terms of convergence quality, from the results of using multimodal function, an example of convergence graph is shown in Figure 3.12. Figure 3.12a shows the convergence results of all the algorithms for dimension 2 and Figure 3.12b is for dimension 30. Based on Figure 3.12, all the proposed algorithms have improved the solution quality as compared to FA and IWO especially in the case of higher dimension.



(a) Results in 2 dimensions (b) Results in 30 dimensions

Figure 3.12: The convergence plot in solving Levy Function

In Figure 3.12a, IWO-eSSF, MIWO-eSSF and HIWFO-SF showed competitive results among the algorithms. These algorithms showed further promise as they are converging and the potential to obtain better solution quality if the iteration is extended. On the other hand, as shown in Figure 3.12b, by increasing the problem dimension to 30, HIWFO-SF and HIWFO showed the potential of hybridization quality as their convergence and accuracy were better quality as compared with the rest of the algorithms.

The results obtained in this section concluded the study on understanding the effects of tuning parameters of adaptive FA, IWO and the proposed hybrid algorithms on their convergence and solution accuracy. The simulations were performed with unimodal Schwefel's Problem 2.22 and multimodal Levy function problems. It was shown that larger number of iterations produce better quality solution. A need for large iteration is essential to compare the algorithms analysed. This is because all the proposed algorithms have shown great potential to get better global optimum value by extending the iteration. Increasing size of population may also increase the solution quality, but it will also increase the computational time and higher NFE. Hence, in comparing all the proposed algorithms, a competitive size of population is used in order to be fair and also able to get the optimal result.

3.6 Summary

In this chapter, investigations of new proposed optimisation based on FA and IWO have been elaborated. The studies present new variants of FA and IWO algorithm. New hybridization strategy referred as hybrid invasive weed firefly optimisation (HIWFO) and hybrid invasive weed optimisation with spread factor (HIWFO-SF) algorithms have also been proposed to solve global optimisation problems. The hybridization of the algorithms has been achieved by embedding the FA method into IWO algorithm structure to enhance the local search capability of IWO that already has very good exploration capability. Moreover, incorporating suitable adaptive parameters of the algorithm could further improve the diversity mechanism in the HIWFO algorithm to further balance the exploration and exploitation abilities to achieve better performance. Based on the results obtained,

- In the unimodal study, as the population size and dimension of the problem increases, FA-NSF, FA-eSF and HIWFO-SF show better solution accuracy. However, IWO-eSSF, MIWO-eSSF and HIWFO have shown slightly better results as compared with FA and IWO. All the proposed algorithms have achieved significant improvement with increased iteration.

- In the multimodal study, only HIWFO-SF achieved significant improvement in solution accuracy with increase in population size and dimension. The other proposed algorithms have achieved slight improvements compared to FA and IWO algorithms.

The algorithms will be further tested with single unconstrained and constrained optimisation problems with continuous design variables. The multi-objective optimisation problems and selected engineering optimisation problem are also used. Simulation and comparative assessments of performance of the proposed algorithms with the original FA and IWO are also carried out on the mentioned optimisation problems to illustrate their effectiveness and robustness.

Chapter 4

Single-objective Adaptive Firefly and Invasive Weed Algorithms

4.1 Introduction

This chapter presents performance analyses of the proposed adaptive firefly and invasive weed algorithms with single-objective optimisation problems. Two types of problems are considered, namely unconstrained and constrained optimisation to test and evaluate the algorithms. For solving single-objective unconstrained optimisation problems, standard well-known benchmark functions and CEC 2014 test functions are used. The functions used have different landscapes, dimensions and complexities with either no or several local optima. Performance measurements are set to measure and compare the performances of the algorithms. On the other hand, well-known benchmark and CEC 2006 test functions are used to evaluate the algorithms to solve constrained optimisation problems. Further tests are conducted with practical engineering design problems which deal with continuous variables in constrained optimisation environment. In each case, graphical and numerical results are presented to carry out out comparative performance assessment of the proposed algorithms with their predecessor, FA and IWO algorithms.

The experimental testing hardware platform comprises a personal computer (PC) with processor CPU Intel (R) Core (TM) i5-2400 Window 7 Professional operating system, 3.10 GHz frequency and 4.00 GB RAM. The program is coded in MATLAB R2013a. Each problem is tested with 30 independent runs with a minimum number of function evaluations of 30,000 per run.

4.2 Unconstrained Optimisation Problems

In general, if the problem is to minimize $f(x)$ over all x. A general unconstrained optimisation problem can be represented as (Simon, 2013)

$$x^* = arg\ \underset{x}{min} f(x) \tag{4.1}$$

where x^* is used to represent the optimising value of x, and $f(x^*)$ is the minimum value of $f(x)$. Details of unconstrained optimisation problems can be found in (Ali et al., 2005; Simon, 2013; Yang, 2010a). In order to analyse the proposed algorithms, benchmark functions or also called test functions are used to obtain comparative results among metaheuristic algorithms (Simon, 2013).

The benchmark function that has single optimum is called unimodal whereas if it has more than one optimum, it is called multimodal. Multimodal functions are used to test the ability of the algorithm to escape from local optima and locate a good near-global optimum. Therefore, for the case of multimodal functions especially in high dimensions, the final results are more important than the convergence rate. The experiment also looks at how effectively could the algorithm be extended for higher dimension problems, although this also will involve increased computational complexity.

4.2.1 Experiments on Unconstrained Optimisation Problems

This section examines the algorithms in solving unconstrained optimisation problems. The experiments are aimed to investigate the performance of FA-NSF, FA-eSF, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF algorithms. Ten standard benchmark functions and 16 CEC 2014 test functions are used in the experiments. All functions used are minimization problems and the dimensions of search space (Dim) are 2, 10, 30 and 50. The performances of the proposed algorithms are also compared with those of FA and IWO algorithm.

Standard Benchmark Functions

The benchmark functions used in this study are adopted from Jamil and Yang (2013), Surjanovic and Bingham (2013) and these are described well by Simon (2013). The functions are also used in the literature to analyse the performance of bio-inspired algorithms on the unconstrained problems. (Wang et al., 2016; Yilmaz and Kucuksille, 2015).

Ten benchmark functions featured with unimodal and multimodal properties are used to evaluate the algorithms. Table 4.1 shows the benchmark functions used for the analysis of the performance tests. Table 4.2 shows a brief summary of the benchmark functions properties and conditions used in this section.

Note: D represents the number of dimensions for $i = 1, \ldots, D$.

CEC 2014 Test Functions

Sixteen different global optimisation problems $(f_{11} - f_{26})$ are additionally used and collected from CEC 2014 single objective real-parameter numerical optimisation (Liang et al., 2013). The mathematical formulations of the CEC 2014 test functions are listed in Appendix B. All functions are minimization problems and have global optimum value. The functions $f_{11} - f_{13}$

Table 4.1: Benchmark functions used

Function	Formulation	Range
Sphere	$f_1(x) = \sum_{i=1}^{D} {x_i}^2$	$[-10, 10]^D$
Rosenbrock	$f_2(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - {x_i}^2)^2 + (x_i - 1)^2]$	$[-5, 10]^D$
Dixon & Price	$f_3(x) = (x_1 - 1)^2 + \sum_{i=1}^{D} i(2{x_i}^2 - x_{i-1})^2$	$[-10, 10]^D$
Schwefel's Problem 1.2	$f_4(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} {x_j}^2 \right)$	$[-10, 10]^D$
Schwefel's Problem 2.22	$f_5(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10, 10]^D$
Zakharov	$f_6(x) = \sum_{i=1}^{D} {x_i}^2 + \left(\sum_{i=1}^{D} 0.5 i x_i \right)^2 + \left(\sum_{i=1}^{D} 0.5 i x_i \right)^4$	$[-5, 10]^D$
Rastrigin	$f_7(x) = 10D + \sum_{i=1}^{D} [{x_i}^2 - 10\cos(2\pi x_i)]$	$[-5.12, 5.12]^D$
Ackley	$f_8(x) = -20\, exp \left(-0.2 \sqrt{\frac{1}{D} \sum_{i=i}^{D} {x_i}^2} \right)$ $-exp \left(\frac{1}{D} \sum_{i=1}^{D} \cos 2\pi x_i \right) + 20 + e$	$[-32, 32]^D$
Griewank	$f_9(x) = \frac{1}{4000} \sum_{i=1}^{D} {x_i}^2 - \prod_{i=1}^{D} \frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]^D$
Levy	$f_{10}(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{D-1} (\omega_i - 1)^2[1 + 10\sin^2(\pi\omega_i + 1)]$ $+(\omega_D - 1)^2[1 + \sin^2(2\pi\omega_D)];$ $where\ \omega_i = 1 + \frac{x_i - 1}{4},\ for\ all\ i = 1, ..., D$	$[-10, 10]^D$

Table 4.2: Brief summary of properties of the benchmark functions

Functions		Properties
$f_1(x)$	Unimodal	Continuous, convex and no local minimum except the global one
$f_2(x)$	Unimodal	The global minimum lies in a narrow, banana-shaped valley
$f_3(x)$	Unimodal	Continuous, differentiable, non-Separable, Scalable
$f_4(x)$	Unimodal	Continuous, differentiable, non-separable, scalable,
$f_5(x)$	Unimodal	Continuous and non-differentiable
$f_6(x)$	Unimodal	Continuous, differentiable, non-separable, scalable
$f_7(x)$	Multimodal	Many local minima locations and regularly distributed
$f_8(x)$	Multimodal	Continuous, differentiable, non-separable, scalable, many local minima
$f_9(x)$	Multimodal	Many local minima locations and regularly distributed
$f_{10}(x)$	Multimodal	Several local minima

are unimodal functions and the rest of the functions are multimodal functions. A summary of their properties and common condition used in this analysis is shown in Table 4.3.

Table 4.3: Summary of the CEC 2014 test functions

No	Functions		Properties	$f^* = f(x_i^*)$
$f_{11}(x)$	Rotated High Conditioned Elliptic Function	Unimodal	Unimodal, non-separable, quadratic ill-conditioned	100
$f_{12}(x)$	Rotated Bent Cigar Function	Unimodal	Unimodal, non-separable, smooth but narrow ridge	200
$f_{13}(x)$	Rotated Discus Function	Unimodal	Unimodal, non-separable, with one sensitive direction	300
$f_{14}(x)$	Shifted and Rotated Rosenbrock's Function	Multimodal	Multimodal, non-separable, having a very narrow valley from local optimum to global optimum	400
$f_{15}(x)$	Shifted and Rotated Ackley's Function	Multimodal	Multimodal, non-separable	500
$f_{16}(x)$	Shifted and Rotated Weierstrass Function	Multimodal	Multimodal, non-separable, continuous but differentiable only on a set of points	600
$f_{17}(x)$	Shifted and Rotated Griewank's Function	Multimodal	Multimodal, non-separable, rotated	700
$f_{18}(x)$	Shifted Rastrigin's Function	Multimodal	Multimodal, separable, local optima's number is huge	800
$f_{19}(x)$	Shifted and Rotated Rastrigin's Function	Multimodal	Multimodal, non-separable, local optima's number is huge	900
$f_{20}(x)$	Shifted Schwefel's Function	Multimodal	Multimodal, separable, local optima's number is huge and second better local optimum is far from the global optimum	1000
$f_{21}(x)$	Shifted and Rotated Schwefel's Function	Multimodal	Multimodal, non-separable, Local optima's number is huge and second better local optimum is far from the global optimum	1100
$f_{22}(x)$	Shifted and Rotated Katsuura Function	Multimodal	Multimodal, non-separable, Continuous everywhere yet differentiable nowhere	1200
$f_{23}(x)$	Shifted and Rotated HappyCat Function	Multimodal	Multimodal, non-separable	1300
$f_{24}(x)$	Shifted and Rotated HGBat Function	Multimodal	Multimodal, non-separable	1400
$f_{25}(x)$	Shifted and Rotated Expanded Griewank's plus Rosenbrock's Function	Multimodal	Multimodal, non-separable	1500
$f_{26}(x)$	Shifted and Rotated Expanded Scaffer's F6 Function	Multimodal	Multimodal, non-separable	1600

Note: n represents the number of dimensions and for all functions $f^* = f(x_i^*)$ where $x_i \in [-100, 100]$ for $i = 1, \ldots, n$.

4.2.2 Performance Measurement

Numerical results from the benchmark functions and CEC 2014 function tests are used to evaluate performances of the proposed algorithms (FA-NSF, FA-eSF, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF) in solving unconstrained optimisation problems.

Performance measurement tools are described in this section to compare, evaluate and analyse the results of all the algorithms. In all the unconstrained problem tests, the same population size, n and the maximum number of iterations are used for a fair comparative evaluation of the algorithms. This basic criteria used in this research are as follows:

- Maximum size of population, $n_{max} = 30$.
- Maximum number of iterations, $it_{max} = 1,000$ (NFE = 30, 000).
- The problems are tested in 2, 10 and 30 and 50 dimensions.

The number of function evaluations (NFE) is also used in the experiments as measure of computational time instead of number of generations. The algorithms are terminated when the criterion NFE = 30,000 is met.

Table 4.4: The initial parameters used in the study

Parameters	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
s_{min}	-	-	-	0	0	0	0	0
s_{max}	-	-	-	5	5	5	5	5
σ_{ini}	-	-	-	5	5	5	5	5
σ_f	-	-	-	0.01	0.01	0.01	0.01	0.01
β_0	1	1	1	-	-	-	1	1
α_{ini}	0.2	1	1	-	-	-	0.2	1
α_f	-	0.001	0.001	-	-	-	-	0.001
γ_{ini}	1	0.001	0.001	-	-	-	1	1
γ_f	-	1	1	-	-	-	-	-

Table 4.4 shows the parameter sets used in the tests where σ_{ini} and σ_f, represent the initial and final values of SD respectively, s_{max} and s_{min}, represent the maximum and minimum values of a seed respectively, γ, light absorption coefficient, β_0, attraction coefficient, and α_{ini}, randomization coefficient used in the algorithms. For implementing the adaptive mechanism of the proposed algorithms (FA-NSF, FA-eSF and HIWFO-SF), γ_{ini} and γ_f represent the initial and final values of light absorption coefficient, and α_{ini} and α_f, are the initial and final values of randomization coefficient used in the algorithms.

The performance evaluation measurement used for the comparison study includes the quality of final solution, the convergence speed towards optimum solution, the success rate (reliability of hitting the optimum threshold) and statistical significance test.

Performance of Global Optimum Solution

During the initialization, the initial population is randomly scattered in the search space. In the tests, 30 independent runs of the algorithms are carried out on each function. The best solution, the average of final solutions of each run, and their respective standard deviations are noted.

Comparison of the results on basis of quality of optimum solution for the algorithms in different dimension ranges are presented in tabulated form. Samples of convergence graphs are also provided to show the performances on the basis of convergence quality and speed towards optimum solution.

Reliability Performance Test

The reliability of an optimisation algorithm is very important to solve a given problem. For this purpose, reliability test is carried out using the data obtained from the experiments. The performance criterion is set on how reliably the algorithm reaches the average fitness threshold (success criterion) after a predefined NFE. For the experiments, the success criterion or accuracy threshold is set to be 10^{-4} as proposed by Gandomi et al. (2013) and Akbari and Ziarati (2011) for each function in different dimensions. The percentage rate of successful runs is determined by calculating the success rate, SR in the experiment, where:

$$SR = \frac{NSR}{NR} \times 100\% \tag{4.2}$$

where NSR is the number of successful runs and NR is the number of runs. The SR is evaluated if the minimum value across the threshold, the value of NFE and time (t) taken for the algorithm to converge to the specific threshold are specified. The SR value shows the robustness of the algorithm in solving optimisation problems. After reaching maximum NFE if the minimum value achieved by the algorithm has not reached the threshold, the run is considered to be unsuccessful and notation '–' is indicated to signify that the algorithm run did not converge to the accuracy threshold. The average SR for an algorithm can be computed using

$$Avg_{SR} = \frac{\Sigma_{k=1}^{f_i(x)} SR}{f(x)} \tag{4.3}$$

where f(x) is number of functions used to evaluate the algorithm and SR is the success rate value of the algorithm. The total time (t) taken as the algorithm converged to the threshold and the respective NFE is also noted.

Statistical Significant Test

In this research, the significant performance tests of the algorithm are also carried out. Kruskal-Wallis non-parametric statistical test is chosen. This is because, although the results of parametric and nonparametric analyses are nearly similar, conditions of parametric tests used in the metaheuristics algorithm analysis are not usually fulfilled, as studied by Garcia et al. (2009). Therefore, the use of non-parametric tests is encouraged and preferred by researchers noted by García et al. (2009). Some suggestions of non-parametric tests for analysis of optimisation algorithms include Kruskal-Wallis, Wilcoxon, Friedman, Iman-Davenport, Bonferroni-Dunn, and Holm.

Kruskal-Wallis test is a non-parametric statistical test to evaluate more than two groups. It is a non-parametric of one-way ANOVA test and an extension of Wilcoxon rank sum test. The test assumes that all samples come from populations having the same continuous distribution, and all observations are mutually independent. The data captured is ranked in the Kruskal-

Wallis test by ordering the data from smallest to largest across all groups. In this case, it takes numeric index of this ordering. In this research, all 30 solutions of each algorithm are ranked and $p - value$ is used to measure significance of chi-square statistic of Kruskal-Wallis test.

In adopting Kruskal-Wallis one-way variance test, the hypothesis is set by assuming that all algorithms performed equally and showed the same median. The result of the test will show the p-value and mean rank for each comparison group / algorithm. If this hypothesis is rejected at $95\%(p - value < 0.05)$ confidence interval, the Kruskal-Wallis test suggests that at least one algorithm involved in the comparison is different from others. In short, at least one of the algorithm's median appear to be different from those of the other algorithms. As most of the problems are minimisation type, the lowest mean rank show that the algorithms have more tendency and accuracy in reaching the minimum global optimum value.

4.2.3 Experimental Results and Performance Analyses

In this section, the results of all the algorithms used in the experiments are analysed and evaluated. Different analyses based on defined performance measurements listed in the previous section are presented. The comparative study will show the ability of the proposed algorithms in solving unconstrained optimisation problems. Following the experiments on the benchmark functions as well as with CEC 2014 test functions, the overall performance of the proposed algorithms are compared together with FA and IWO.

Standard Benchmark Functions

The performance and analysis of the proposed algorithms are shown in this section for the standard benchmark function tests. The numerical results comparing the mean, standard deviation and the best optimum value after 30 runs with dimensions 2, 10, 30 and 50 are presented in Tables 4.5, 4.6, 4.7 and 4.8, respectively. The mean value is defined as the mean best fitness value by averaging of 30 simulation runs. The functions f_1–f_6, are unimodal whereas functions f_7–f_{10} are multimodal problems. The measurements are based on the optimum value for reaching the predefined maximum iterations for each function.

Table 4.5 represents comparison results of eight algorithms for ten standard benchmark functions in 2 dimension problem. The numerical results show the solution quality of each algorithm in solving these problems. The highlighted bold font signifies the best obtained values. The results mostly show that the algorithms successfully optimised the functions as they obtained near to zero value of global optimum. In Table 4.5, the statistical results of FA-NSF, FA-eSF and HIWFO-SF demonstrate a far better solution quality than other algorithms. Furthermore, HIWFO-SF showed more precise optimum value and significantly outperformed other algorithms.

Table 4.6 tabulates the experimental results in 10 dimensions. As the dimension increased, the complexity of the functions also increased. As noted FA-eSF, FA-NSF and HIWFO-

Table 4.5: Results for benchmark functions in 2 dimensions

		FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_1	Mean	2.28E-06	5.37E-15	1.05E-15	1.05E-07	4.57E-08	2.70E-07	1.26E-06	**5.47E-16**
	Std Dev	2.59E-06	5.58E-15	1.19E-15	8.72E-08	3.97E-08	2.05E-07	1.51E-06	**6.78E-16**
	Best	5.22E-08	9.54E-17	1.28E-18	5.04E-09	1.85E-10	2.49E-08	4.70E-08	**2.26E-18**
f_2	Mean	1.88E-05	2.32E-03	1.68E-03	9.22E-07	3.36E-07	3.61E-06	2.53E-05	**5.61E-07**
	Std Dev	1.57E-05	2.87E-03	1.47E-03	6.51E-07	3.14E-07	2.96E-06	2.64E-05	**1.52E-06**
	Best	5.28E-07	6.28E-05	3.88E-05	9.94E-08	1.19E-09	3.27E-07	2.28E-08	**1.45E-14**
f_3	Mean	7.39E-06	4.73E-04	7.10E-04	3.15E-07	2.49E-07	1.72E-06	5.38E-06	**4.75E-15**
	Std Dev	8.33E-06	4.21E-04	5.28E-04	2.89E-07	2.02E-07	1.51E-06	5.35E-06	**3.79E-15**
	Best	6.88E-08	3.46E-05	4.10E-07	8.54E-09	2.51E-09	3.91E-08	7.55E-08	**3.94E-16**
f_4	Mean	1.77E-06	6.33E-15	1.45E-15	1.01E-07	4.62E-08	3.61E-07	1.50E-06	**1.24E-15**
	Std Dev	1.37E-06	5.83E-15	1.09E-15	1.18E-07	4.10E-08	3.03E-07	1.28E-06	**1.36E-15**
	Best	1.45E-07	9.74E-17	8.26E-17	8.31E-11	5.00E-09	1.28E-08	9.57E-08	**1.39E-17**
f_5	Mean	3.38E-03	7.87E-08	3.70E-08	3.99E-04	2.35E-04	5.41E-04	1.02E-03	**3.27E-08**
	Std Dev	8.10E-04	6.56E-08	1.95E-08	1.99E-04	1.27E-04	2.26E-04	5.19E-04	**1.34E-08**
	Best	3.05E-03	1.43E-08	3.84E-09	6.58E-05	1.41E-05	8.16E-05	2.27E-04	**1.27E-08**
f_6	Mean	2.89E-06	9.84E-15	1.53E-15	1.40E-07	6.27E-08	4.93E-07	1.89E-06	**2.71E-15**
	Std Dev	2.95E-06	1.35E-14	1.34E-15	1.22E-07	6.33E-08	5.11E-07	2.86E-06	**2.64E-15**
	Best	1.35E-07	2.20E-16	1.23E-17	2.18E-08	5.94E-10	5.96E-09	5.31E-08	**4.02E-17**
f_7	Mean	1.87E+00	1.12E-12	2.75E-13	2.08E-05	1.09E-05	6.31E-05	1.32E-04	**2.00E-13**
	Std Dev	1.68E+00	1.22E-12	2.15E-13	1.98E-05	7.40E-06	7.05E-05	1.89E-04	**5.28E-13**
	Best	1.14E-04	1.78E-14	7.11E-15	1.13E-07	5.21E-07	2.10E-06	2.43E-06	**3.55E-15**
f_8	Mean	2.19E+00	5.21E-09	1.22E-09	9.18E-06	8.05E-06	1.26E-05	1.73E-04	**1.23E-08**
	Std Dev	2.42E+00	4.28E-09	1.04E-09	1.07E-05	9.05E-06	1.44E-05	1.53E-04	**2.08E-08**
	Best	1.13E-05	6.17E-11	9.39E-12	8.23E-07	7.39E-08	5.85E-08	3.24E-06	**5.34E-10**
f_9	Mean	4.16E+00	2.25E-15	3.85E-16	2.47E-04	4.93E-04	1.48E-03	6.99E-03	**4.07E-16**
	Std Dev	3.44E+00	2.10E-15	3.74E-16	1.35E-03	1.88E-03	3.01E-03	9.18E-03	**3.62E-16**
	Best	7.88E-01	1.11E-16	1.11E-16	8.49E-10	6.60E-10	2.38E-08	9.37E-09	**1.11E-16**
f_{10}	Mean	2.33E-01	3.69E-03	9.75E-04	2.41E-08	1.18E-08	1.02E-07	3.70E-07	**2.18E-16**
	Std Dev	4.12E-01	4.37E-03	9.44E-04	2.24E-08	1.04E-08	8.33E-08	3.39E-07	**1.62E-16**
	Best	5.54E-08	2.75E-05	4.85E-05	1.22E-10	2.74E-10	4.59E-09	2.50E-10	**5.17E-19**

SF algorithms outperformed other algorithms and the proposed FA variants achieved more precise solutions. The statistical results of IWO-eSSF show competitive result and better than IWO.

Table 4.7 shows the comparison results among the eight algorithms for 30 dimensional standard benchmark problems and Table 4.8 shows the results in 50 dimension problems. The pattern of results is consistent with Table 4.6. The FA-eSF, FA-NSF, IWO-eSSF and HIWFO-SF showed far better average convergence value than other algorithms. The SF mechanism adopted was enable to help the proposed algorithms to jump out of the local optima in the higher dimensional problems. Hence, this adaptive mechanism implemented in those algorithms can effectively prevent premature convergence and enhance the solution quality of the algorithms.

Rosenbrock function, f_2 is one of the problems which is hard to optimise especially if

Table 4.6: Results for benchmark functions in 10 dimensions

		FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_1	Mean	1.22E+02	1.05E-12	**3.38E-13**	1.66E-04	1.30E-04	6.86E-04	4.42E-04	2.34E-09
	Std Dev	2.05E+01	7.47E-13	**1.80E-13**	4.16E-05	3.20E-05	3.84E-04	1.59E-04	5.49E-09
	Best	7.97E+01	2.46E-13	**1.66E-13**	8.03E-05	5.40E-05	5.18E-06	2.65E-04	1.81E-15
f_2	Mean	7.85E+04	8.77E+00	8.82E+00	1.88E+01	**4.33E+00**	6.23E+00	1.43E+01	7.73E+00
	Std Dev	2.94E+04	1.42E-01	**1.01E-01**	7.14E+01	7.30E-01	5.94E-01	2.15E+01	4.90E-01
	Best	1.94E+04	8.15E+00	8.56E+00	3.87E+00	**3.22E+00**	5.18E+00	8.17E-01	6.90E+00
f_3	Mean	5.17E+04	7.21E-01	7.40E-01	6.67E-01	**6.67E-01**	6.70E-01	6.70E-01	**6.67E-01**
	Std Dev	2.06E+04	2.21E-02	2.36E-02	3.10E-04	**1.23E-04**	1.02E-03	1.83E-01	2.59E-04
	Best	1.28E+04	6.89E-01	6.88E-01	6.67E-01	6.67E-01	6.68E-01	**1.66E-02**	6.67E-01
f_4	Mean	1.12E+02	2.94E-12	**5.82E-13**	2.75E-04	1.97E-04	1.30E-03	2.15E-03	1.70E-04
	Std Dev	2.24E+01	4.21E-12	**2.23E-13**	8.78E-05	5.42E-05	4.52E-04	1.47E-03	1.70E-04
	Best	6.80E+01	**1.45E-13**	1.54E-13	1.35E-04	1.02E-04	1.76E-05	7.42E-04	1.97E-06
f_5	Mean	2.83E+01	2.63E-06	**1.46E-06**	3.51E-02	2.75E-02	6.84E-02	5.85E-02	3.65E-03
	Std Dev	2.90E+00	1.00E-06	**3.08E-07**	3.39E-03	4.16E-03	1.64E-02	1.79E-02	8.37E-03
	Best	2.22E+01	1.29E-06	**9.52E-07**	2.91E-02	1.79E-02	5.70E-03	3.11E-02	7.08E-06
f_6	Mean	1.61E+02	3.63E-12	**8.13E-13**	3.55E-04	2.34E-04	1.64E-03	1.46E-03	3.59E-05
	Std Dev	3.17E+02	3.86E-12	**4.19E-13**	1.19E-04	4.81E-05	6.16E-04	7.22E-04	4.62E-05
	Best	2.72E+01	6.56E-13	**2.94E-13**	1.48E-04	1.16E-04	3.21E-05	4.86E-04	5.64E-07
f_7	Mean	5.98E+01	1.99E-10	**5.94E-11**	1.04E+01	8.53E-01	1.11E-01	1.30E+01	1.07E-06
	Std Dev	7.28E+00	1.15E-10	**1.68E-11**	5.76E+00	1.89E+00	7.72E-02	7.41E+00	2.69E-06
	Best	4.01E+01	3.45E-11	**2.89E-11**	3.01E+00	1.04E-02	1.93E-04	4.08E+00	1.15E-11
f_8	Mean	1.95E+00	5.09E-09	**2.60E-09**	1.75E-05	7.79E-06	1.78E-05	1.11E-04	8.84E-07
	Std Dev	2.47E+00	5.29E-09	**2.86E-09**	1.84E-05	8.03E-06	2.31E-05	8.62E-05	4.72E-06
	Best	1.79E-05	5.60E-11	**1.01E-11**	9.72E-07	1.72E-07	8.81E-07	1.08E-05	9.16E-11
f_9	Mean	1.39E+02	1.95E-13	**4.53E-14**	9.00E-02	8.23E-02	9.03E-02	1.90E-01	3.94E-09
	Std Dev	2.36E+01	1.82E-13	**2.25E-14**	3.76E-02	4.06E-02	5.60E-02	1.73E-01	8.01E-09
	Best	7.02E+01	2.89E-14	**1.08E-14**	3.20E-02	1.73E-02	9.98E-03	4.19E-02	5.37E-13
f_{10}	Mean	1.85E+01	7.09E-01	5.71E-01	8.70E-05	**5.93E-05**	4.36E-04	1.52E-02	2.73E-02
	Std Dev	3.10E+00	9.27E-02	9.67E-02	2.26E-05	**1.64E-05**	1.02E-04	4.13E-02	6.28E-02
	Best	1.23E+01	5.22E-01	3.52E-01	4.84E-05	**2.92E-05**	2.54E-04	8.99E-05	1.62E-04

Table 4.7: Results for benchmark functions in 30 dimensions

		FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_1	Mean	6.20E+02	4.13E-12	**1.79E-12**	3.28E-03	2.09E-03	1.54E-02	1.05E-02	1.94E-04
	Std Dev	5.36E+01	4.34E-12	**3.96E-13**	4.19E-04	2.71E-04	1.98E-03	2.74E-03	2.85E-04
	Best	4.57E+02	1.54E-12	**1.09E-12**	2.57E-03	1.59E-03	1.03E-02	6.35E-03	2.20E-05
f_2	Mean	1.24E+06	2.89E+01	2.89E+01	1.70E+02	4.02E+01	2.95E+01	1.56E+02	**2.85E+01**
	Std Dev	2.07E+05	**2.83E-02**	3.14E-02	3.34E+02	3.22E+01	1.20E+00	1.90E+02	2.14E-01
	Best	7.13E+05	2.88E+01	2.88E+01	**2.12E+01**	2.31E+01	2.66E+01	2.45E+01	2.80E+01
f_3	Mean	1.40E+06	9.49E-01	9.57E-01	7.14E+01	8.09E-01	1.06E+00	1.97E+01	**7.35E-01**
	Std Dev	1.62E+05	**1.74E-02**	1.80E-02	1.52E+02	2.53E-01	3.17E-01	1.55E+01	5.93E-02
	Best	1.03E+06	9.01E-01	9.04E-01	7.18E-01	6.85E-01	8.21E-01	8.24E-01	**6.68E-01**
f_4	Mean	8.72E+02	**4.63E-11**	5.42E-12	8.69E-01	1.69E+00	3.22E-01	2.33E+01	2.22E-02
	Std Dev	1.99E+02	1.06E-10	**1.98E-12**	3.77E-01	1.46E+00	1.97E-01	8.24E+00	6.52E-03
	Best	5.78E+02	5.45E-12	**2.13E-12**	2.89E-01	6.12E-02	1.11E-01	8.53E+00	9.52E-03
f_5	Mean	5.29E+06	1.14E-05	**5.89E-06**	7.92E+00	1.95E-01	8.54E-01	7.02E+00	2.86E-01
	Std Dev	1.03E+07	7.57E-06	**6.38E-07**	1.28E+01	1.45E-02	6.29E-01	1.52E+01	8.20E-02
	Best	1.71E+02	4.78E-06	**4.61E-06**	2.13E-01	1.63E-01	4.67E-01	6.03E-01	1.68E-01
f_6	Mean	5.71E+08	3.44E-11	**8.65E-12**	6.04E+01	6.75E-01	1.22E-01	3.90E+01	2.05E-02
	Std Dev	8.11E+08	2.31E-11	**3.92E-12**	6.26E+01	6.66E-01	5.69E-02	1.86E+01	7.13E-03
	Best	7.72E+02	3.82E-12	**2.20E-12**	1.51E+00	4.62E-02	5.47E-02	9.02E+00	7.11E-03
f_7	Mean	3.32E+02	1.75E-09	**3.88E-10**	8.03E+01	2.98E+01	5.61E+00	7.06E+01	2.52E-02
	Std Dev	1.11E+01	2.99E-09	**1.33E-10**	1.69E+01	1.47E+01	4.22E+00	1.88E+01	2.50E-02
	Best	3.11E+02	2.93E-10	**1.88E-10**	4.55E+01	3.46E+00	1.69E-02	3.64E+01	9.57E-04
f_8	Mean	2.15E+00	6.71E-09	**1.30E-09**	1.26E-05	5.94E-06	2.16E-05	1.66E-04	7.66E-09
	Std Dev	3.20E+00	6.33E-09	**1.24E-09**	1.78E-05	6.07E-06	2.76E-05	1.78E-04	9.56E-09
	Best	7.24E-05	1.73E-10	**3.30E-12**	3.86E-07	2.49E-07	3.24E-07	6.10E-06	1.78E-10
f_9	Mean	6.06E+02	2.70E-13	**9.74E-14**	6.91E-03	7.43E-03	4.20E-03	1.30E-02	3.45E-05
	Std Dev	7.37E+01	2.02E-13	**2.82E-14**	7.21E-03	6.64E-03	6.26E-03	1.15E-02	3.61E-05
	Best	3.94E+02	7.09E-14	**5.17E-14**	1.34E-04	8.68E-05	6.86E-04	5.36E-04	1.52E-06
f_{10}	Mean	1.49E+02	**1.03E-01**	2.48E+00	4.29E+00	2.64E+00	1.23E+00	1.23E+01	4.50E-01
	Std Dev	1.19E+01	2.46E+00	**1.21E-01**	4.31E+00	2.16E+00	1.16E+00	5.25E+00	5.49E-01
	Best	1.28E+02	2.70E+00	2.22E+00	**1.58E-03**	9.05E-02	9.69E-03	2.51E+00	2.91E-02

the dimension is increased. As shown in Tables 4.6, 4.7 and 4.8, all the algorithms struggled to obtain global optimum value especially as the problem dimension increased from 10 to 50 dimensions. However, for these functions, IWO-eSFF and HIWFO-SF showed some potential as they significantly outperformed other algorithms in the Rosenbrock functions across different dimensions. The multimodal Levy function, f_{10} is also another benchmark function that is hard to optimise. For dimension increased to 30 and 50, all the proposed FA and IWO as well as HIWFO-SF outperformed other algorithms in tackling this problem. The adaptive SF mechanism helps the proposed algorithm to improve the diversification and intensification of the algorithms during the evolutionary process.

From dimension 10 and above, FA found it hard to converge for Rosenbrock and Levy function. For other multimodal functions, f_7–f_{10}, FA method was easily trapped in local optima. The performance of FA also showed the same result for the other functions in higher

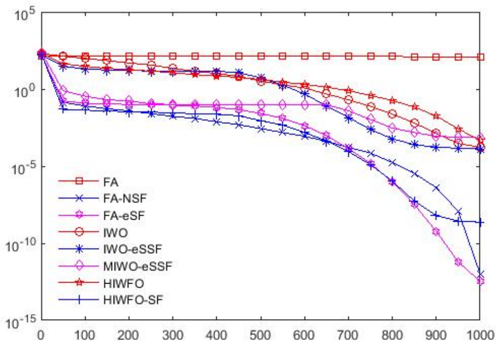
Table 4.8: Results for benchmark functions in 50 dimensions

		FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_1	Mean	1.19E+03	2.18E-11	**3.25E-12**	1.20E-02	7.07E-03	5.16E-02	9.68E-02	7.85E-04
	Std Dev	5.06E+01	5.74E-11	**8.18E-13**	1.32E-03	7.34E-04	5.99E-03	2.84E-02	1.14E-03
	Best	1.08E+03	2.56E-12	**1.89E-12**	9.83E-03	5.74E-03	4.18E-02	5.74E-02	5.00E-05
f_2	Mean	2.84E+06	4.89E+01	4.89E+01	2.81E+02	8.26E+01	7.37E+01	3.38E+02	**4.88E+01**
	Std Dev	3.86E+05	**2.99E-02**	3.03E-02	5.68E+02	5.25E+01	4.09E+01	2.40E+02	2.03E-01
	Best	1.14E+06	4.88E+01	4.88E+01	4.61E+01	**4.59E+01**	5.03E+01	1.09E+02	4.84E+01
f_3	Mean	5.09E+06	9.86E-01	9.87E-01	1.86E+02	6.31E+00	6.37E+00	4.32E+01	**9.43E-01**
	Std Dev	3.33E+05	**4.74E-03**	4.85E-03	3.25E+02	7.76E+00	6.30E+00	3.90E+01	4.49E-02
	Best	4.40E+06	9.75E-01	9.74E-01	1.28E+00	8.75E-01	2.11E+00	9.90E+00	**8.47E-01**
f_4	Mean	2.41E+03	7.27E-11	**1.47E-11**	1.22E+02	5.42E+01	1.89E+01	2.14E+02	3.58E-02
	Std Dev	4.00E+02	1.24E-10	**6.28E-12**	3.75E+01	1.31E+01	8.29E+00	4.96E+01	1.14E-02
	Best	1.31E+03	6.55E-12	**3.33E-12**	2.40E+01	3.49E+01	7.94E+00	8.97E+01	6.83E-03
f_5	Mean	4.94E+15	1.51E-05	**1.04E-05**	1.24E+02	2.08E+00	5.93E+00	7.83E+02	5.43E-01
	Std Dev	1.71E+16	5.23E-06	**1.52E-06**	6.10E+01	1.44E+00	3.46E+00	2.66E+03	8.79E-02
	Best	5.61E+12	9.10E-06	**7.54E-06**	2.76E+01	4.63E-01	1.24E+00	2.34E+01	3.68E-01
f_6	Mean	2.23E+11	1.57E-10	**2.15E-11**	5.61E+03	8.13E+01	3.08E+01	4.48E+02	4.09E-02
	Std Dev	2.00E+11	1.93E-10	**1.09E-11**	2.36E+03	2.60E+01	2.62E+01	8.28E+01	9.83E-03
	Best	5.66E+09	1.84E-11	**5.33E-12**	2.12E+03	4.44E+01	6.32E+00	2.35E+02	2.06E-02
f_7	Mean	6.43E+02	1.98E-09	**7.33E-10**	1.88E+02	7.41E+01	2.15E+01	1.50E+02	2.87E-01
	Std Dev	1.29E+01	1.50E-09	**2.38E-10**	3.09E+01	2.11E+01	1.29E+01	3.16E+01	5.82E-01
	Best	6.19E+02	4.36E-10	**3.87E-10**	1.40E+02	2.55E+01	3.08E+00	1.01E+02	3.33E-03
f_8	Mean	2.10E+00	4.73E-09	**1.33E-09**	1.93E-05	7.58E-06	1.60E-05	1.23E-04	1.15E-08
	Std Dev	2.54E+00	5.00E-09	**1.03E-09**	1.89E-05	1.11E-05	1.79E-05	9.60E-05	1.60E-08
	Best	2.69E-04	1.23E-10	**2.69E-11**	1.34E-07	8.39E-09	2.23E-07	1.95E-06	2.26E-10
f_9	Mean	1.14E+03	3.75E-13	**1.34E-13**	8.74E-03	5.16E-03	6.03E-03	2.99E-02	4.38E-05
	Std Dev	6.38E+01	3.14E-13	**3.76E-14**	9.13E-03	6.37E-03	6.84E-03	1.36E-02	8.36E-05
	Best	9.97E+02	**6.00E-14**	7.90E-14	4.91E-04	2.29E-04	1.65E-03	5.27E-03	2.04E-07
f_{10}	Mean	3.22E+02	4.60E+00	4.36E+00	3.58E+01	1.61E+01	7.44E+00	4.37E+01	**1.86E+00**
	Std Dev	1.59E+01	**8.37E-02**	1.54E-01	2.07E+01	5.06E+00	3.42E+00	1.22E+01	9.54E-01
	Best	2.81E+02	4.30E+00	3.91E+00	4.66E+00	7.64E+00	**4.84E-01**	2.20E+01	8.03E-01

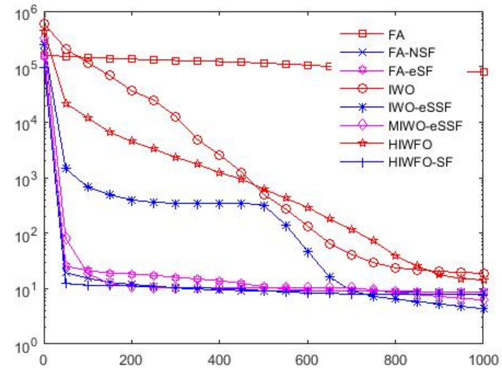
dimension problems. However, adoption of the adaptive mechanism in FA-NSF and FA-eSF, has improved and enhanced the evolutionary search of the algorithm.

Benchmark functions f_7–f_{10} represent multimodal problems. As noted in Tables 4.6, 4.7 and 4.8, multimodal functions with many local optima were successfully optimised by FA-NSF, FA-eSF, IWO-eSF and HIWFO-SF algorithms. The adaptive mechanism also helped the algorithms to significantly enhance the convergence accuracy during the iteration process.

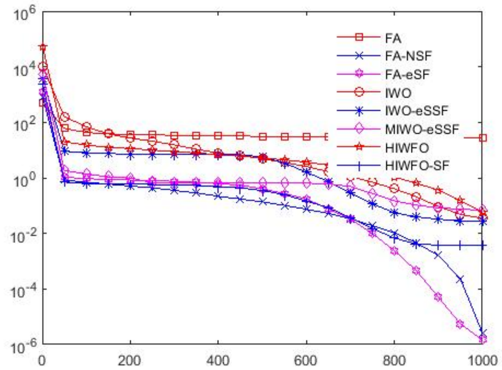
Figures 4.1 and 4.2 show the convergence quality of the algorithms for some samples from benchmark problems. This illustrates the convergence plot of unimodal functions, f_1, f_2 and f_5 as well as samples from multimodal functions, f_7, f_8 and f_9. The convergence lines show the evolution behaviour of the algorithms throughout the iteration process. Each value on the graph represents the best mean fitness value of the algorithm during the optimisation process. It implied whether the algorithm may or may not provide better performance as



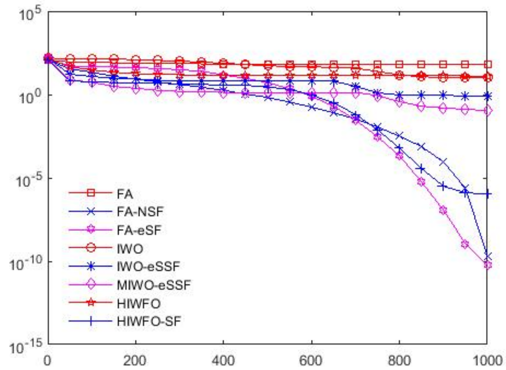
(a) f1



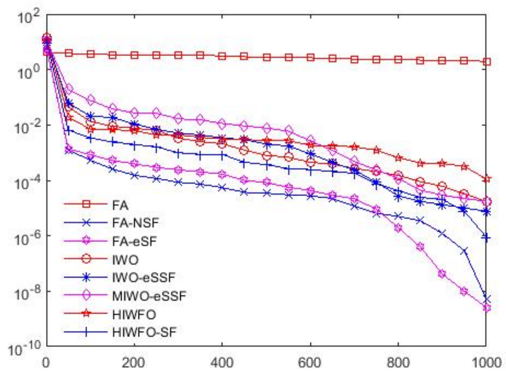
(b) f_2



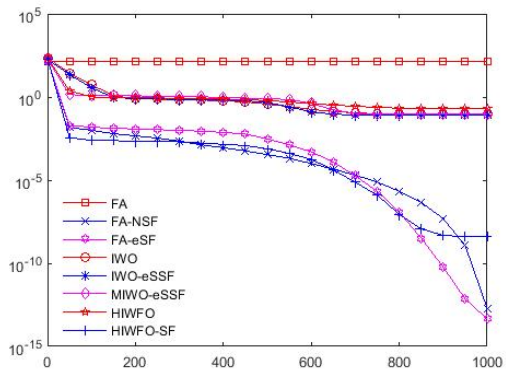
(c) f_5



(d) f_7



(e) f_8



(f) f_9

Figure 4.1: Convergence plots of 10-dimensional basic benchmark problems

the number of iterations is extended. It also shows the speed of the convergence during the optimisation process. Hence, the solution accuracy and fast convergence can be determined based on the obtained graph.

Figure 4.1 shows six convergence plots of all the algorithms simulated on 10-dimensional of six selected standard benchmark functions. On the other hand, Figure 4.2 shows the same functions used, however, on 50-dimensional problems. Each graph shows the best mean fitness value of each algorithm based on 30 simulation runs in log-10 scale over 1000 iterations. Function f_1, f_2 and f_5 represent unimodal problems and other functions, f_7, f_8 and f_9

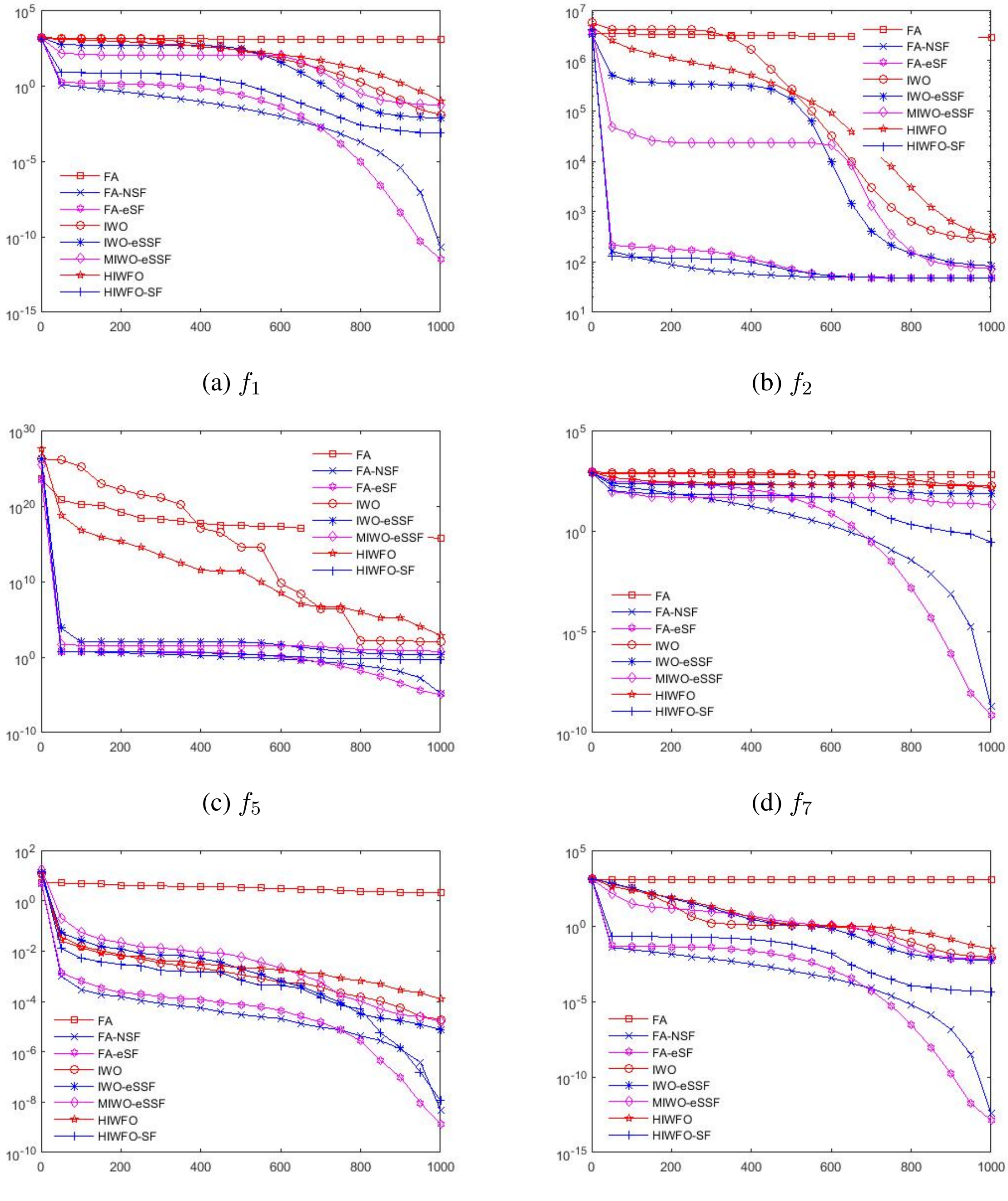
represent multimodal problems. The maximum NFE is set to 30,000.

From Figure 4.1, it can be seen that each algorithm kept converging as the iterations increased. Hence, their optimal solutions achieve better quality as the iterations are extended. Based on the observation, FA-eSF, FA-NSF, MIWO-eSSF and HIWFO-SF converged faster than other algorithms in less than 100 iterations. Not only giving the best optimal value, both FA-NSF and FA eSF methods also converged rapidly to the optimum criteria in most of the benchmark problems. IWO-eSSF and HIWFO-SF also converged fast. As the iteration passed 500, HIWFO struggled to converge and the results were not improved as compared to FA and IWO variants.

Based on the observation in Figures 4.1a – 4.1d, FA-NSF, FA-eSF and HIWFO-SF showed faster convergence as compared with other algorithms at less than 100 iterations. These proposed algorithms still kept converging as the iterations is extended to 1000. MIWO-eSSF also performed fast convergence for the unimodal functions. Although HIWFO performed better at the final optimum value compared to IWO and FA, but HIWFO showed slower convergence as compared to IWO. It was also slower in convergence than MIWO-eSSF. However, as the iteration increased, it started to improve and slowly converged and showed competitive solution quality compared to HIWFO-SF. As noted FA showed the worst convergence as it got stuck at the local optimum point and seemed hard to improve as the iterations increased.

Figure 4.2 shows algorithm convergence for more complex problems as the dimensions are increased to 50. The same pattern is seen in solving this high dimension of standard benchmark problems. Based on the observation from Figure 4.2, HIWFO-SF, FA-NSF and FA-eSF showed faster convergence on all the functions. As the iterations increased beyond 100 iteration, they also continued to converge more for better solution quality. MIWO-eSSF and IWO-eSF were also able to convergence fast at the early stage of iterations, however, after more than 100 iterations, the convergence became slower. On the other hand, IWO converged steadily and slightly gave better solution quality than HIWFO. FA also showed the same pattern as seen as in Figure 4.1, where it failed to converge in all the functions.

The analysis of convergence shows that the proposed FA variants, FA-NSF and FA-eSF outperformed the FA. IWO-eSSF and MIWO-eSSF algorithms also outperformed their predecessor algorithm, IWO. Furthermore, HIWFO-SF algorithm showed faster convergence and competitive solution accuracy compared to all the algorithms used in the experiment.



(a) f_1 (b) f_2

(c) f_5 (d) f_7

(e) f_8 (f) f9

Figure 4.2: Convergence plots of 50-dimensional basic benchmark problems

Reliability Performance Test Results

In this section, the results to measure reliability of each algorithm are presented. Tables 4.9, 4.10 and 4.11 show the success rate (SR) results in 2 and 30 dimensions for FA variants, IWO variants and hybrid algorithms, respectively. Table 4.12 summarises the SR values for the functions in different dimensions.

Table 4.9 shows a comparison based on NFE and SR shown by FA, FA-NSF and FA-eSF. The results show that optima of f_1, f_4, and f_6 were easily achieved by all the FA algorithms in 2 dimensional problems with original FA showing shorter time and NFE values. However, as the problem dimension increased to 30, the proposed FA variants improved the condition and managed to achieved optima of f_1 and f_4–f_9 with 100% success rate shown (Noted as 1.0 in the table). It is also noted that FA-eSF was efficient and reliable over FA-NSF as it achieved shorter time and lower NFE values especially dealing with problems in higher dimension.

Table 4.9: Results of success rate for FA variants

		FA			FA-NSF			FA-eSF		
$f(x)$	Dim	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE
f_1	2	1.0	0.27	3000	1.0	0.461	4140	1.0	0.763	6870
	30	0	-	-	1.0	2.998	24060	1.0	2.718	22350
f_2	2	1.0	1.103	12300	0	-	-	0	-	-
	30	0	-	-	0	-	-	0	-	-
f_3	2	1.0	0.513	5790	0.2	-	-	0.1	-	-
	30	0	-	-	0	-	-	0	-	-
f_4	2	1.0	0.345	3690	1.0	0.515	4530	1.0	0.793	7140
	30	0	-	-	1.0	3.286	24990	1.0	2.952	23040
f_5	2	0	-	-	1.0	2.834	25560	1.0	2.607	23430
	30	0	-	-	1.0	3.696	29460	1.0	3.357	27510
f_6	2	1.0	0.227	2460	1.0	0.542	4680	1.0	0.922	8250
	30	0	-	-	1.0	3.173	25350	1.0	2.864	23280
f_7	2	0	-	-	1.0	2.325	20040	1.0	2.311	20790
	30	0	-	-	1.0	3.413	27750	1.0	3.074	24990
f_8	2	0	-	-	1.0	1.068	8970	1.0	1.671	14670
	30	0	-	-	1.0	1.127	9060	1.0	1.733	13950
f_9	2	0	-	-	1.0	0.28	2160	1.0	0.746	6510
	30	0	-	-	1.0	2.591	19770	1.0	2.586	20310
f_{10}	2	0	-	-	0.1	-	-	0.1	-	-
	30	0	-	-	0	-	-	0	-	-

Table 4.10 shows performance comparison of IWO and the proposed IWO variants, IWO-eSSF and MIWO-eSSF. In addition, Table 4.11 compares performances of proposed hybrid algorithms, HIWFO and HIWFO-SF with the original FA and IWO algorithm.

In Table 4.10, it is recorded that all IWO algorithms managed to converge to the given threshold for all the functions in 2 dimension. As seen in Table 4.11, HIWFO-SF algorithm also managed to converge for the same case. However, for this case, IWO-eSSF recorded faster convergence and lower NFE value compared to other IWO variants and HIWFO-SF algorithms. IWO algorithms managed to achieve global optima of multimodal functions f_8 and f_9 and failed to converge for other functions in higher dimension.

Table 4.10: Results of success rate for IWO variants

			IWO			**IWO-eSSF**			**MIWO-eSSF**	
$f(x)$	Dim	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE
f_1	2	1.0	0.711	20820	1.0	0.69	19410	1.0	0.714	20550
	30	0	-	-	0	-	-	0	-	-
f_2	2	1.0	0.849	24750	1.0	0.797	21270	1.0	0.865	23040
	30	0	-	-	0	-	-	0	-	-
f_3	2	1.0	0.814	23880	1.0	0.762	20580	1.0	0.815	22110
	30	0	-	-	0	-	-	0	-	-
f_4	2	1.0	0.698	20790	1.0	0.713	19320	1.0	0.734	20100
	30	0	-	-	0	-	-	0	-	-
f_5	2	1.0	0.895	27390	1.0	0.833	23370	1.0	0.919	26250
	30	0	-	-	0	-	-	0	-	-
f_6	2	1.0	0.732	21390	1.0	0.729	19140	1.0	0.794	20880
	30	0	-	-	0	-	-	0	-	-
f_7	2	1.0	0.879	27810	1.0	0.808	23910	1.0	0.916	27570
	30	0	-	-	0	-	-	0	-	-
f_8	2	1.0	0.893	25470	1.0	0.836	22710	1.0	0.837	23940
	30	1.0	1.062	25770	1.0	0.905	21630	1.0	0.775	19980
f_9	2	1.0	0.524	14640	1.0	0.625	16860	1.0	0.479	12840
	30	1.0	1.133	26160	1.0	0.985	22410	1.0	1.062	23430
f_{10}	2	1.0	0.927	19200	1.0	0.791	17640	1.0	0.872	19590
	30	0	-	-	0	-	-	0	-	-

As noted in Table 4.10, HIWFO-SF managed to optimum values for f_1, f_4, and functions, f_6–f_9 with 100% success rate. In addition, HIWFO-SF was more efficient than FA-eSF and FA-NSF as it achieved shorter time and lower NFE values.

Table 4.12 summarises the values of SR for all the algorithms for the standard benchmark function test. The listed values are the average SR value of each function after 4 different dimension tests and Avg_{SR} value shows the average SR value of all the test functions for the algorithm.

All algorithms successfullyachieved global optima of functions f_1, f_4 and f_6 as they showed SR of 1.0 in lower dimension. FA-eSF, FA-NSF and HIWFO-SF showed better result for all the functions with different dimensions as it achieved higher SR. However, HIWFO-SF was more efficient than FA-eSF and FA-NSF as it achieved shorter time and lower NFE values.

Statistical Significant Test

In this section, Kruskal-Wallis non-parametric test is used for comparative statistical analysis of the algorithms. The Kruskal-Wallis test is conducted based on 95% confidence interval and 30 simulation runs for each algorithm. The results show the mean rank, rank number

Table 4.11: Result of success rate for the proposed hybrid algorithms

		FA			**IWO**			**HIWFO**			**HIWFO-SF**		
$f(x)$	Dim	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE	SR	$t, (sec)$	NFE
f_1	2	1.0	0.27	3000	1.0	0.711	20820	1.0	1.219	25440	1.0	0.301	5880
	30	0	-	-	0	-	-	0	-	-	1.0	1.237	23160
f_2	2	1.0	1.103	12300	1.0	0.849	24750	1.0	1.408	28680	1.0	1.113	20040
	30	0	-	-	0	-	-	0	-	-	0	-	-
f_3	2	1	0.513	5790	1.0	0.814	23880	1.0	1.327	27180	1.0	0.911	16470
	30	0	-	-	0	-	-	0	-	-	0	-	-
f_4	2	1.0	0.345	3690	1.0	0.698	20790	1.0	1.082	25650	1.0	0.305	5850
	30	0	-	-	0	-	-	0	-	-	1.0	1.113	17070
f_5	2	0	-	-	1.0	0.895	27390	0	-	-	1.0	1.167	23250
	30	0	-	-	0	-	-	0	-	-	0	-	-
f_6	2	1.0	0.227	2460	1.0	0.732	21390	1.0	1.122	25170	1.0	0.468	8850
	30	0	-	-	0	-	-	0	-	-	1.0	0.942	16530
f_7	2	0	-	-	1.0	0.879	27810	1.0	1.044	25290	1.0	1.028	20670
	30	0	-	-	0	-	-	0	-	-	1.0	1.334	25230
f_8	2	0	-	-	1.0	0.893	25470	1.0	0.696	17850	1.0	1.097	21210
	30	0	-	-	1.0	1.062	25770	1.0	0.962	21510	1.0	1.097	19410
f_9	2	0	-	-	1.0	0.524	14640	0	-	-	1.0	0.138	2670
	30	0	-	-	1.0	1.133	26160	0	-	-	1.0	1.252	23670
f_{10}	2	0	-	-	1.0	0.927	19200	1.0	1.082	20400	1.0	0.164	2910
	30	0	-	-	0	-	-	0	-	-	0	-	-

Table 4.12: Overall result of the success rate

$f(x)$	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_1	30%	100%	100%	30%	30%	30%	50%	100%
f_2	30%	0	0	30%	30%	30%	30%	30%
f_3	30%	0	0	30%	30%	30%	30%	30%
f_4	30%	100%	100%	30%	30%	30%	30%	80%
f_5	0	100%	100%	30%	30%	30%	0	30%
f_6	30%	100%	100%	30%	30%	30%	30%	100%
f_7	0	100%	100%	30%	30%	30%	30%	80%
f_8	0	100%	100%	100%	100%	100%	80%	100%
f_9	0	100%	100%	0.8	100%	80%	0	100%
f_{10}	0	0	0	50%	50%	50%	30%	30%
Avg_{SR}	13%	71%	71%	40%	43%	40%	28%	65%

(bracket) and two-tailed p-value. The significant difference is considered if the probability value is less than $0.05 (p - value < 0.05)$.

Table 4.13 shows the results of the Kruskal-Wallis non-parametric test of all the algorithms for 10 standard benchmark functions in dimensions 2, 10, 30 and 50, respectively. From Table 4.13, it can be seen that all results show the two-tailed p-value score less than 0.05, which implies that there are significant median difference among all the outputs given. The test ranked all the results in ascending order which is from minimum to maximum value. Hence, it can be easily concluded that the lowest mean rank shows significantly better result than others. Based on the observation from Table 4.13, HIWFO-SF has shown the smallest value of average mean rank (58.24) and rank (2.25). On the other hand, FA-eSF had slightly lower value of the average mean rank (59.08) and rank (2.40). It can be summarized that both FA-eSF and HIWFO-SF algorithms statistically dominated the performance as compared with other algorithms. In addition, other proposed algorithms also had lower mean rank than FA and IWO algorithm.

Table 4.13: The ranking of algorithms based on statistical significant test results for benchmark functions

$f(x)$	Dim	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF	*p-value**
f_1	2	209.63 (8)	66.50 (3)	41.60 (2)	137.63 (5)	117.70 (4)	166.83 (6)	195.70 (7)	28.40 (1)	*1.07E-41*
	10	225.50 (8)	52.43 (2)	25.80 (1)	131.40 (5)	115.90 (4)	180.00 (7)	174.70 (6)	58.27 (3)	*1.07E-45*
	30	225.50 (8)	40.97 (2)	20.03 (1)	135.50 (5)	105.50 (4)	193.27 (7)	167.73 (6)	75.50 (3)	*1.11E-46*
	50	225.50 (8)	42.53 (2)	18.47 (1)	135.50 (5)	105.50 (4)	165.87 (6)	195.13 (7)	75.50 (3)	*6.56E-47*
f_2	2	140.57 (5)	210.87 (8)	209.37 (7)	75.23 (3)	47.90 (2)	104.40 (4)	144.00 (6)	31.67 (1)	*3.90E-40*
	10	225.50 (8)	162.30 (6)	171.00 (7)	67.03 (2)	22.97 (1)	71.13 (3)	128.83 (5)	115.23 (4)	*3.71E-37*
	30	225.50 (8)	98.03 (3)	109.80 (4)	121.53 (5)	78.37 (2)	128.50 (6)	148.37 (7)	53.90 (1)	*4.37E-22*
	50	225.50 (8)	61.27 (2)	77.90 (3)	122.17 (5)	103.20 (4)	142.37 (6)	188.07 (7)	43.53 (1)	*1.00E-33*
f_3	2	139.77 (5)	205.93 (7)	211.73 (8)	72.33 (3)	66.80 (2)	112.70 (4)	139.23 (6)	15.50 (1)	*2.51E-40*
	10	225.50 (8)	165.47 (6)	180.67 (7)	70.27 (3)	47.30 (2)	119.53 (4)	130.03 (5)	25.23 (1)	*9.38E-42*
	30	225.50 (8)	109.27 (3)	119.73 (5)	139.23 (6)	44.13 (2)	116.30 (4)	176.47 (7)	33.37 (1)	*2.50E-34*
	50	225.50 (8)	54.27 (2)	58.73 (3)	173.93 (6)	119.60 (4)	129.93 (5)	175.60 (7)	26.43 (1)	*3.34E-42*
f_4	2	208.30 (8)	66.13 (3)	38.40 (2)	132.23 (5)	118.73 (4)	165.27 (6)	202.97 (7)	31.97 (1)	*2.23E-42*
	10	225.50 (8)	39.40 (2)	21.60 (1)	122.97 (5)	103.67 (4)	173.27 (6)	184.70 (7)	92.90 (3)	*6.72E-44*
	30	225.50 (8)	43.93 (2)	17.07 (1)	143.70 (5)	151.10 (6)	111.70 (4)	195.50 (7)	75.50 (3)	*8.21E-46*
	50	225.50 (8)	41.20 (2)	19.80 (1)	166.03 (6)	136.67 (5)	105.87 (4)	193.43 (7)	75.50 (3)	*1.50E-46*
f_5	2	225.50 (8)	61.27 (3)	40.23 (2)	141.50 (5)	116.80 (4)	159.57 (6)	184.13 (7)	35.00 (1)	*1.51E-42*
	10	225.50 (8)	43.43 (2)	17.57 (1)	135.30 (5)	107.53 (4)	185.53 (7)	171.07 (6)	78.07 (3)	*1.78E-45*
	30	225.50 (8)	42.50 (2)	18.50 (1)	154.60 (6)	79.10 (3)	154.17 (5)	178.93 (7)	110.70 (4)	*1.71E-43*
	50	225.50 (8)	40.43 (2)	20.57 (1)	178.03 (6)	106.50 (4)	131.37 (5)	182.97 (7)	78.63 (3)	*1.07E-45*
f_6	2	210.77 (8)	64.57 (3)	30.93 (1)	138.57 (5)	117.27 (4)	167.13 (6)	193.77 (7)	41.00 (2)	*1.82E-41*
	10	225.50 (8)	43.90 (2)	17.10 (1)	131.50 (5)	111.80 (4)	180.57 (7)	177.23 (6)	76.40 (3)	*1.68E-45*
	30	225.50 (8)	41.23 (2)	19.77 (1)	181.07 (7)	133.20 (5)	108.10 (4)	179.63 (6)	75.50 (3)	*4.86E-46*
	50	225.50 (8)	43.57 (2)	17.43 (1)	195.50 (7)	133.47 (5)	107.53 (4)	165.50 (6)	75.50 (3)	*7.41E-47*
f_7	2	224.87 (8)	66.55 (3)	43.90 (2)	138.37 (5)	123.50 (4)	166.33 (6)	174.43 (7)	26.05 (1)	*4.90E-42*
	10	224.50 (8)	47.47 (2)	22.50 (1)	175.40 (6)	121.03 (5)	120.90 (4)	183.79 (7)	66.53 (3)	*1.43E-44*
	30	225.50 (8)	41.33 (2)	19.67 (1)	184.20 (7)	135.77 (5)	106.40 (4)	175.27 (6)	75.87 (3)	*5.16E-46*
	50	224.50 (8)	40.30 (2)	20.70 (1)	189.13 (7)	135.03 (5)	106.23 (4)	170.28 (6)	75.50 (3)	*3.60E-46*
f_8	2	210.37 (8)	55.93 (3)	25.60 (1)	137.40 (5)	131.43 (4)	142.97 (6)	185.72 (7)	55.07 (2)	*3.00E-39*
	10	207.27 (8)	45.47 (2)	35.17 (1)	143.10 (6)	125.20 (4)	138.67 (5)	182.77 (7)	58.47 (3)	*1.91E-37*
	30	209.43 (8)	56.13 (3)	27.27 (1)	139.83 (5)	126.53 (4)	145.67 (6)	179.07 (7)	53.10 (2)	*9.95E-39*
	50	210.43 (8)	51.07 (2)	30.43 (1)	144.23 (6)	124.60 (4)	142.03 (5)	178.27 (7)	55.57 (3)	*1.39E-38*
f_9	2	225.50 (8)	67.97 (3)	34.28 (2)	136.50 (5)	116.50 (4)	165.10 (6)	183.90 (7)	34.25 (1)	*2.10E-43*
	10	225.50 (8)	43.22 (2)	17.85 (1)	145.73 (6)	139.70 (4)	142.33 (5)	174.23 (7)	75.43 (3)	*4.82E-42*
	30	225.50 (8)	42.37 (2)	18.63 (1)	144.43 (6)	143.57 (5)	140.80 (4)	172.73 (7)	75.97 (3)	*8.86E-42*
	50	225.50 (8)	39.90 (2)	21.10 (1)	144.57 (6)	123.83 (4)	142.40 (5)	190.40 (7)	76.30 (3)	*7.35E-44*
f_{10}	2	172.90 (6)	208.27 (8)	194.73 (7)	74.07 (3)	61.50 (2)	106.00 (4)	131.03 (5)	15.50 (1)	*9.84E-41*
	10	225.50 (8)	191.03 (7)	169.97 (6)	41.73 (2)	20.20 (1)	114.40 (5)	90.87 (3)	110.30 (4)	*2.05E-44*
	30	225.50 (8)	131.87 (6)	106.33 (4)	118.23 (5)	100.20 (3)	63.07 (2)	190.10 (7)	28.70 (1)	*1.12E-34*
	50	225.50 (8)	78.37 (3)	51.07 (2)	169.97 (6)	136.73 (5)	97.93 (4)	186.53 (7)	17.90 (1)	*2.07E-44*
Average		217.05	76.22	59.08	135.74	106.4	135.55	173.08	58.24	
Average (Rank)		7.8	3.13	2.4	5.15	3.75	5.03	6.5	2.25	

Comparative results with other metaheuristics algorithms

The algorithms are also compared with six state-of-the-art metaheuristic algorithms to verify the reliability and validity of the algorithm. Yan et al. (2012) conducted a comparative study on performance of adaptive firefly algorithm (AFA), FA, PSO and DE on 10 benchmark functions with three different dimensions. The results obtained from 5 functions are compared with the proposed algorithms as shown in Table 4.14. Moreover, Yilmaz and Kucukseille (2015) compared an enhanced bat algorithm (EBA) with BA and GA. The results from 7 functions on 30 and 50 dimensions are also tabulated to compare with the best mean values obtained in this study. The best solution in each case has been marked in bold font.

From Table 4.14, it can be seen that FA-NSF and FA-eSF achieved better results in both 30 and 50 dimensions for all the benchmark functions in terms of mean search precision. MIWO-eSSF and HIWFO-SF algorithms also showed better performance compared to the mentioned state-of-the-art algorithms.

Table 4.14: Performance comparison for unconstrained optimisation problems

	D	**BA**	**EBA**	**GA**	**AFA**	**PSO**	**DE**	**FA-NSF**	**FA-eSF**	**IWO-eSSF**	**MIWO-eSSF**	**HIWFO**	**HIWFO-SF**
f_1	30	4.65E+03	2.06E-05	1.11E+03	-	-	-	**4.13E-12**	**1.79E-12**	2.09E-03	1.54E-02	1.05E-02	1.94E-04
	50	1.06E+04	2.08E-05	-		-	-	**2.18E-11**	**3.25E-12**	7.07E-03	5.16E-02	9.68E-02	7.85E-04
f_2	30	1.98E+05	**2.11E+01**	1.96E+05	2.72E+01	3.29E+01	3.21E+05	2.89E+01	2.89E+01	4.02E+01	2.95E+01	1.56E+02	2.85E+01
	50	1.04E+06	5.45E+01	-	-	-	-	4.89E+01	4.89E+01	8.26E+01	7.37E+01	3.38E+02	**4.88E+01**
f_4	30	1.26E+04	1.18E-04	7.40E+03	-	-	-	**4.63E-11**	**5.42E-12**	1.69E+00	3.22E-01	2.33E+01	2.22E-02
	50	3.95E+04	1.07E-02	-	-	-	-	**7.27E-11**	**1.47E-11**	5.42E+01	1.89E+01	2.14E+02	3.58E-02
f_5	30	4.09E+00	3.11E-01	1.10E+01	6.75E-02	3.80E-01	2.82E+01	**1.14E-05**	**5.89E-06**	1.95E-01	8.54E-01	7.02E+00	2.86E-01
	50	2.07E+01	1.34E+00	-	-	-	-	**1.51E-05**	**1.04E-05**	2.08E+00	5.93E+00	7.83E+02	5.43E-01
f_7	30	1.42E+02	3.44E+01	5.29E+01	4.67E+01	7.15E+01	3.22E+02	**1.75E-09**	**3.88E-10**	2.98E+01	5.61E+00	7.06E+01	2.52E-02
	50	2.25E+02	5.15E+01					**1.98E-09**	**7.33E-10**	7.41E+01	2.15E+01	1.50E+02	2.87E-01
f_8	30	9.17E+00	4.58E-01	1.47E+01	2.57E-02	1.92E+00	1.45E+01	**6.71E-09**	**1.30E-09**	5.94E-06	2.16E-05	1.66E-04	7.66E-09
	50	1.08E+01	1.80E+00	-	-	-	-	**4.73E-09**	**1.33E-09**	7.58E-06	1.60E-05	1.23E-04	1.15E-08
f_9	30	6.45E+01	7.30E-03	1.06E+01	4.79E-02	1.07E-01	5.41E+01	**2.70E-13**	**9.74E-14**	7.43E-03	4.20E-03	1.30E-02	3.45E-05
	50	1.16E+02	4.68E-03	-	-	-	-	**3.75E-13**	**1.34E-13**	5.16E-03	6.03E-03	2.99E-02	4.38E-05

CEC 2014 Test Functions

In this section, the CEC 2014 test functions are used to analysis the performance of the proposed algorithms. As noted in the previous section, the test functions are run with dimension 2, 10, 30 and 50. The numerical results comparing the mean error and standard deviation are presented in Tables 4.15 and 4.16, respectively. The mean error is the error between the best average fitness value, f_i with the optimal value of each function, f_i^* where f_i represents the CEC 2014 test functions as shown in Table 4.3. Unimodal problems are represented by functions $f_{11} - f_{13}$ whereas functions $f_{14} - f_{26}$ represent multimodal problems.

Table 4.15 presents the comparative results in 2 and 10 dimensions. On the other hand, Table 4.16 shows the results in 30 and 50 dimensions. The best result obtained is highlighted in bold font. It is noted in Table 4.15, that HIWFO-SF has outperformed other algorithms for CEC 2014 unimodal problems. The mean error and standard deviation of the algorithm was the lowest among them. HIWFO-SF also performed better in 2 dimensional problem for functions, f_{14}, f_{18}, f_{22}, and f_{23}. In comparison, MIWO-eSSF showed the best value for functions, f_{15}–f_{21} especially in 10 dimensional problems. IWO-eSSF demonstrated better performance in dimension 2 for functions, f_{24} and f_{25} as well as in dimension 10 for f_{25}. IWO algorithm also showed competitive results compared to the original FA algorithm.

In Table 4.16, results of higher dimensions 30 and 50 are shown. It is noted that, IWO-eSSF performed better in functions f_{11} and f_{25} for both dimensions and f_{20}, f_{21}, f_{23}, and f_{24} in dimension 30. Meanwhile, MIWO-eSSF scored better in functions f_{18}, f_{19} and f_{26} for both dimensions. MIWO-eSSF also performed better in dimension 30 for function f_{16}, and in dimension 50 for functions, f_{13}, f_{20}, f_{21} and f_{24}. It is noted that HIWFO-SF achieved competitive results in functions, f_{12}, f_{15} and f_{17} for dimensions 30 and 50 compared to the other algorithms. The results also show that the proposed variants of FA outperformed their predecessor, FA algorithm.

Overall, the results show that IWO-eSSF, MIWO-eSSF and HIWFO-SF achieved a slightly better solution quality than other algorithms. IWO-eSSF achieved more precise optimum value and can be concluded that the algorithm outperformed other algorithms in the CEC 2014 function test.

Figure 4.3 shows the convergence plot of all the algorithms on 2-dimensional CEC 2014 test functions. Functions f_{12} and f_{13} represent unimodal problems and f_{15}, f_{18}, f_{19} and f_{24} are selected plots to study multimodal problems. In addition, by using the same functions, Figure 4.4 shows those function plots in 30-dimensional problems. Each graph shows the best mean fitness value of each algorithm based on 30 simulation runs in log-10 scale. The maximum iteration and NFE are set to 1,000 and 30,000, respectively.

As noted in Figure 4.3, HIWFO, HIWFO-SF, IWO-eSSF and MIWO-eSSF achieved faster convergence at less than 100 iterations. After 100 iterations, HIWFO-SF, IWO-eSSF and MIWO-eSSF improved the solution quality and kept converging. However, HIWFO was slower and appears to have got stuck at local optimum for f_{15}, f_{18} and f_{19}. The performance

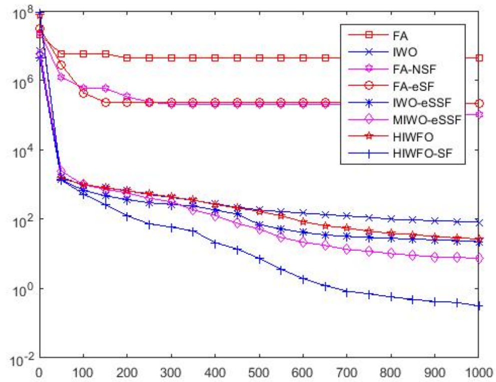
Table 4.15: Results for CEC 2014 test problems in dimensions 2 and 10

f_x	Dim	Stats	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_{11}	2	Mean	3.08E+06	2.89E+04	9.59E+04	5.33E+02	3.58E+02	7.75E+01	1.23E+03	**5.28E+00**
		Std Dev	5.46E+06	5.65E+04	2.19E+05	1.53E+03	6.56E+02	3.11E+02	2.05E+03	**1.96E+01**
	10	Mean	3.95E+08	3.54E+08	3.15E+08	3.32E+05	1.84E+05	1.40E+05	2.10E+05	**2.71E+05**
		Std Dev	3.35E+08	3.40E+08	2.66E+08	2.93E+05	1.65E+05	1.20E+05	2.20E+05	**2.25E+05**
f_{12}	2	Mean	4.35E+06	1.03E+05	2.22E+05	8.06E+01	2.23E+01	7.49E+00	2.67E+01	**3.16E-01**
		Std Dev	7.88E+06	2.21E+05	8.35E+05	1.91E+02	6.51E+01	3.66E+01	7.36E+01	**9.75E-01**
	10	Mean	9.66E+09	1.06E+10	1.19E+10	1.67E+04	1.18E+04	7.45E+04	1.15E+04	**3.24E+03**
		Std Dev	3.53E+09	3.49E+09	3.66E+09	7.11E+03	3.30E+03	1.83E+04	3.60E+03	**3.26E+03**
f_{13}	2	Mean	1.71E+06	6.23E+05	5.13E+04	2.61E+02	9.96E+01	1.97E+01	1.45E+02	**4.26E+00**
		Std Dev	3.71E+06	3.32E+06	1.38E+05	5.17E+02	2.25E+02	7.03E+01	3.42E+02	**1.53E+01**
	10	Mean	6.80E+05	6.50E+05	8.29E+05	1.19E+04	7.66E+03	5.26E+03	1.18E+04	**3.43E+03**
		Std Dev	2.35E+06	1.21E+06	1.43E+06	6.94E+03	5.67E+03	2.72E+03	8.99E+03	**3.60E+03**
f_{14}	2	Mean	2.32E+00	1.21E+00	4.79E-01	1.96E-08	7.56E-09	7.79E-08	3.84E-08	**9.92E-10**
		Std Dev	6.84E+00	3.95E+00	1.19E+00	1.83E-08	1.08E-08	7.51E-08	4.59E-08	**3.93E-09**
	10	Mean	2.80E+03	2.64E+03	3.17E+03	**2.30E+00**	2.27E+01	1.97E+01	3.05E+01	2.06E+01
		Std Dev	1.88E+03	1.40E+03	2.05E+03	**1.17E+00**	1.66E+01	1.69E+01	1.40E+01	1.70E+01
f_{15}	2	Mean	1.59E+01	1.37E+01	1.21E+01	9.33E+00	8.67E+00	**3.34E+00**	9.34E+00	4.00E+00
		Std Dev	**4.43E+00**	5.06E+00	5.11E+00	1.01E+01	1.01E+01	7.57E+00	1.01E+01	8.13E+00
	10	Mean	2.08E+01	2.07E+01	2.07E+01	2.01E+01	2.01E+01	**1.97E+01**	2.01E+01	2.00E+01
		Std Dev	1.46E-01	1.62E-01	1.49E-01	2.34E-02	1.52E-02	3.53E+00	2.58E-02	**2.08E-03**
f_{16}	2	Mean	1.27E+00	9.20E-01	9.18E-01	**4.77E-03**	4.63E-03	8.23E-03	3.68E-01	1.50E-01
		Std Dev	4.94E-01	4.34E-01	4.05E-01	1.92E-03	**1.78E-03**	3.42E-03	6.69E-01	4.58E-01
	10	Mean	1.29E+01	1.29E+01	1.26E+01	8.85E-01	1.10E+00	**4.59E-01**	6.00E+00	2.06E+00
		Std Dev	1.57E+00	1.26E+00	1.42E+00	1.17E+00	1.09E+00	**2.45E-01**	2.18E+00	1.76E+00
f_{17}	2	Mean	2.33E+00	2.66E+00	1.69E+00	**3.20E-04**	1.93E-03	8.70E-04	8.12E-04	8.30E-03
		Std Dev	1.93E+00	3.81E+00	1.53E+00	**1.34E-03**	3.52E-03	2.22E-03	2.26E-03	2.65E-02
	10	Mean	1.96E+02	2.40E+02	2.05E+02	2.17E+01	2.51E+00	**4.35E-01**	6.11E+01	8.36E+01
		Std Dev	7.53E+01	6.56E+01	7.37E+01	5.64E+01	1.30E+01	**7.66E-02**	7.25E+01	4.75E+01
f_{18}	2	Mean	5.24E+00	3.10E+00	3.27E+00	2.85E-06	1.71E-06	8.48E-06	2.42E+00	**3.90E-11**
		Std Dev	3.11E+00	1.85E+00	2.24E+00	4.66E-06	1.63E-06	8.13E-06	3.80E+00	**5.88E-11**
	10	Mean	1.01E+02	9.71E+01	1.05E+02	1.06E+02	9.99E+00	**8.66E+00**	4.87E+01	2.14E+01
		Std Dev	1.39E+01	1.59E+01	1.63E+01	3.18E+01	4.74E+00	**3.08E+00**	1.73E+01	9.55E+00
f_{19}	2	Mean	6.00E+00	2.99E+00	2.94E+00	3.32E-02	1.51E-06	1.15E-05	2.29E+00	**7.08E-11**
		Std Dev	3.88E+00	2.31E+00	2.41E+00	1.82E-01	1.48E-06	1.46E-05	2.01E+00	**1.40E-10**
	10	Mean	1.13E+02	1.11E+02	1.11E+02	1.65E+01	1.13E+01	**9.74E+00**	4.73E+01	1.77E+01
		Std Dev	1.59E+01	1.08E+01	9.99E+00	7.67E+00	4.93E+00	**3.50E+00**	1.38E+01	8.75E+00
f_{20}	2	Mean	1.67E+02	6.70E+01	7.74E+01	3.44E+01	4.23E+01	**1.15E-01**	1.08E+02	8.42E+01
		Std Dev	1.03E+02	5.32E+01	5.53E+01	5.24E+01	**5.12E+01**	1.53E-01	1.04E+02	6.73E+01
	10	Mean	2.45E+03	2.14E+03	2.16E+03	4.49E+02	2.88E+02	**2.41E+02**	1.14E+03	4.85E+02
		Std Dev	2.92E+02	**1.95E+02**	2.31E+02	2.10E+02	1.42E+02	1.98E+02	3.41E+02	1.96E+02
f_{21}	2	Mean	1.51E+02	2.92E+01	4.95E+01	2.30E+03	3.18E+01	**1.19E+01**	9.80E+01	6.67E+01
		Std Dev	9.23E+01	**3.30E+01**	5.39E+01	6.01E+02	5.34E+01	3.61E+01	1.07E+02	7.39E+01
	10	Mean	2.53E+03	2.34E+03	2.41E+03	4.81E+03	3.28E+02	**3.13E+02**	1.14E+03	5.38E+02
		Std Dev	2.88E+02	3.44E+02	3.36E+02	7.96E+02	2.17E+02	**1.99E+02**	3.90E+02	2.67E+02
f_{22}	2	Mean	2.88E+00	9.89E-01	1.26E+00	2.19E-02	1.70E-02	4.65E-02	2.78E-02	**1.21E-04**
		Std Dev	1.92E+00	6.93E-01	6.64E-01	9.92E-03	8.64E-03	2.56E-02	1.49E-02	**1.20E-04**
	10	Mean	4.30E+00	2.74E+00	2.29E+00	7.16E-02	**6.34E-02**	1.50E-01	1.33E-01	1.25E-01
		Std Dev	1.08E+00	9.07E-01	6.84E-01	**3.19E-02**	3.27E-02	5.76E-02	9.71E-02	8.25E-02
f_{23}	2	Mean	8.90E-01	5.44E-01	4.91E-01	4.24E-02	3.92E-02	4.74E-02	5.58E-02	**1.83E-02**
		Std Dev	4.18E-01	1.23E-01	1.30E-01	1.41E-02	**1.15E-02**	1.43E-02	1.87E-02	1.42E-02
	10	Mean	4.41E+00	4.96E+00	4.72E+00	1.31E-01	**1.14E-01**	1.45E-01	3.49E+00	3.11E-01
		Std Dev	9.20E-01	1.21E+00	1.18E+00	5.33E-02	3.43E-02	**3.35E-02**	3.33E+00	7.83E-01
f_{24}	2	Mean	7.25E-01	4.97E-01	6.15E-01	3.70E-03	**2.99E-03**	4.94E-03	5.56E-03	3.24E-03
		Std Dev	3.97E-01	2.67E-01	6.97E-01	1.88E-03	**1.61E-03**	2.16E-03	8.21E-03	4.48E-03
	10	Mean	4.07E+01	4.86E+01	5.17E+01	**1.35E-01**	1.42E-01	1.80E-01	3.50E-01	1.59E-01
		Std Dev	1.14E+01	1.36E+01	1.35E+01	**4.58E-02**	4.87E-02	7.04E-02	1.16E-01	5.62E-02
f_{25}	2	Mean	1.75E+00	5.82E-01	3.28E-01	1.00E-11	**1.63E-12**	2.89E-10	4.61E-03	6.58E-04
		Std Dev	2.75E+00	4.70E-01	2.73E-01	1.34E-11	**3.16E-12**	5.54E-10	8.49E-03	3.60E-03
	10	Mean	9.00E+04	5.57E+04	5.70E+04	9.16E-01	**8.77E-01**	1.15E+00	1.38E+00	1.02E+00
		Std Dev	1.23E+05	3.49E+04	3.68E+04	**2.71E-01**	2.74E-01	3.08E-01	6.68E-01	4.23E-01
f_{26}	2	Mean	5.04E-01	2.87E-01	2.48E-01	**3.25E-03**	4.54E-03	6.51E-03	3.25E-03	7.77E-03
		Std Dev	2.35E-01	2.24E-01	2.05E-01	**7.36E-03**	8.36E-03	9.29E-03	7.36E-03	9.68E-03
	10	Mean	4.43E+00	4.25E+00	4.21E+00	3.19E+00	2.90E+00	**2.36E+00**	3.64E+00	3.14E+00
		Std Dev	**1.63E-01**	2.20E-01	1.93E-01	4.55E-01	5.73E-01	3.76E-01	3.42E-01	4.00E-01

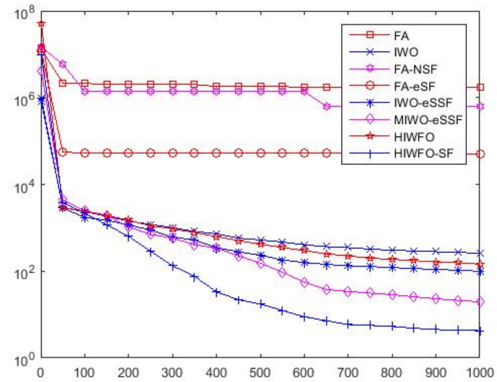
Table 4.16: Results for CEC 2014 test problems in dimensions 30 and 50

f_x	Dim	Stats	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f_{11}	30	Mean	3.59E+09	2.53E+09	2.62E+09	8.05E+06	**6.68E+06**	6.86E+06	7.44E+06	1.01E+07
		Std Dev	1.20E+09	4.35E+08	2.22E+08	4.62E+06	**3.72E+06**	4.43E+06	3.09E+06	5.93E+06
	50	Mean	1.12E+10	9.01E+09	9.45E+09	1.93E+07	**1.06E+07**	1.25E+07	1.23E+07	1.76E+07
		Std Dev	2.98E+09	2.64E+09	2.10E+09	6.30E+06	**4.01E+06**	4.99E+06	5.40E+06	7.49E+06
f_{12}	30	Mean	1.13E+11	9.57E+10	9.68E+10	3.89E+05	2.98E+05	2.18E+06	2.83E+05	**9.20E+04**
		Std Dev	2.19E+10	5.72E+09	6.09E+09	5.11E+04	4.05E+04	2.64E+05	5.83E+04	**2.34E+04**
	50	Mean	2.68E+11	1.91E+11	1.92E+11	1.34E+06	9.69E+05	7.27E+06	8.97E+05	**6.16E+05**
		Std Dev	4.38E+10	9.32E+09	5.75E+09	1.72E+05	1.21E+05	8.46E+05	1.29E+05	**1.36E+05**
f_{13}	30	Mean	2.30E+07	6.21E+06	6.08E+06	9.11E+04	6.07E+04	5.14E+04	8.93E+04	**4.78E+04**
		Std Dev	4.48E+07	1.06E+07	1.13E+07	1.96E+04	1.89E+04	1.34E+04	3.58E+04	**1.15E+04**
	50	Mean	1.52E+07	4.08E+05	5.85E+05	1.55E+05	1.10E+05	**9.14E+04**	1.51E+05	9.73E+04
		Std Dev	2.63E+07	2.43E+05	1.07E+06	3.48E+04	2.18E+04	**1.82E+04**	4.35E+04	2.26E+04
f_{14}	30	Mean	3.45E+04	2.27E+04	2.30E+04	**3.84E+01**	1.07E+02	1.11E+02	1.05E+02	1.19E+02
		Std Dev	1.08E+04	2.21E+03	2.45E+03	**2.47E+01**	3.53E+01	3.80E+01	4.23E+01	2.94E+01
	50	Mean	1.13E+05	6.69E+04	6.67E+04	**6.27E+01**	1.22E+02	1.19E+02	1.56E+02	1.37E+02
		Std Dev	3.10E+04	4.55E+03	6.01E+03	3.14E+01	3.12E+01	**2.63E+01**	4.87E+01	5.42E+01
f_{15}	30	Mean	2.12E+01	2.12E+01	2.12E+01	2.05E+01	2.04E+01	2.10E+01	2.04E+01	**2.01E+01**
		Std Dev	8.11E-02	7.63E-02	8.79E-02	6.04E-02	4.27E-02	5.22E-02	5.53E-02	**3.65E-02**
	50	Mean	2.14E+01	2.13E+01	2.13E+01	2.08E+01	2.07E+01	2.12E+01	2.07E+01	**2.05E+01**
		Std Dev	5.10E-02	5.83E-02	6.13E-02	4.46E-02	4.49E-02	**3.59E-02**	5.77E-02	7.79E-02
f_{16}	30	Mean	4.89E+01	4.60E+01	4.67E+01	7.22E+00	8.61E+00	**7.05E+00**	2.64E+01	1.22E+01
		Std Dev	2.26E+00	2.75E+00	2.32E+00	2.28E+00	2.72E+00	**2.06E+00**	3.95E+00	3.39E+00
	50	Mean	8.48E+01	8.22E+01	8.14E+01	**2.10E+01**	2.48E+01	2.44E+01	5.06E+01	3.28E+01
		Std Dev	2.81E+00	**2.67E+00**	2.72E+00	3.36E+00	4.07E+00	3.81E+00	5.11E+00	3.92E+00
f_{17}	30	Mean	1.05E+03	9.87E+02	9.78E+02	2.10E+02	7.19E+01	9.78E-01	6.89E+02	**2.43E-01**
		Std Dev	1.79E+02	9.61E+01	7.93E+01	3.54E+02	2.18E+02	2.15E-02	3.69E+01	**5.27E-02**
	50	Mean	2.51E+03	1.79E+03	1.81E+03	5.80E+01	7.48E-01	1.07E+00	1.20E+03	**6.98E-01**
		Std Dev	4.49E+02	7.36E+01	4.35E+01	3.13E+02	5.18E-02	**7.50E-03**	6.77E+02	7.41E-02
f_{18}	30	Mean	4.63E+02	4.30E+02	4.33E+02	1.03E+02	7.47E+01	**5.73E+01**	1.87E+02	9.27E+01
		Std Dev	4.29E+01	4.32E+01	4.22E+01	2.82E+01	2.34E+01	**1.39E+01**	3.83E+01	2.87E+01
	50	Mean	9.24E+02	7.96E+02	7.99E+02	2.56E+02	1.69E+02	**1.40E+02**	3.45E+02	2.12E+02
		Std Dev	6.46E+01	5.42E+01	5.87E+01	4.83E+01	3.70E+01	**2.39E+01**	4.67E+01	4.57E+01
f_{19}	30	Mean	5.32E+02	4.53E+02	4.54E+02	1.02E+02	7.83E+01	**6.23E+01**	2.52E+02	8.98E+01
		Std Dev	5.53E+01	1.93E+01	2.33E+01	3.02E+01	2.48E+01	**1.48E+01**	5.40E+01	2.82E+01
	50	Mean	1.16E+03	9.16E+02	9.17E+02	2.49E+02	1.63E+02	**1.53E+02**	4.93E+02	2.01E+02
		Std Dev	1.09E+02	5.94E+01	5.26E+01	5.39E+01	2.90E+01	**2.73E+01**	8.04E+01	4.38E+01
f_{20}	30	Mean	9.46E+03	8.63E+03	8.64E+03	2.40E+03	**2.17E+03**	2.28E+03	3.85E+03	2.85E+03
		Std Dev	4.47E+02	5.31E+02	5.36E+02	6.01E+02	**4.30E+02**	4.80E+02	6.11E+02	7.18E+02
	50	Mean	1.66E+04	1.56E+04	1.57E+04	4.91E+03	4.80E+03	**4.66E+03**	6.93E+03	5.37E+03
		Std Dev	7.67E+02	6.98E+02	7.08E+02	7.96E+02	**6.97E+02**	8.74E+02	1.08E+03	7.25E+02
f_{21}	30	Mean	9.63E+03	8.74E+03	8.78E+03	2.54E+03	**2.31E+03**	2.58E+03	3.83E+03	3.06E+03
		Std Dev	4.49E+02	5.01E+02	4.96E+02	**4.17E+02**	5.85E+02	5.96E+02	6.94E+02	6.20E+02
	50	Mean	1.69E+04	1.59E+04	1.57E+04	5.21E+03	5.24E+03	**5.02E+03**	7.17E+03	5.71E+03
		Std Dev	6.86E+02	5.37E+02	7.86E+02	7.50E+02	7.58E+02	**6.82E+02**	6.69E+02	7.99E+02
f_{22}	30	Mean	5.80E+00	4.93E+00	4.79E+00	**1.49E-01**	1.72E-01	3.69E-01	3.52E-01	3.22E-01
		Std Dev	1.17E+00	9.49E-01	1.11E+00	**5.78E-02**	5.74E-02	1.02E-01	1.61E-01	1.71E-01
	50	Mean	7.12E+00	5.36E+00	5.34E+00	**3.25E-01**	3.45E-01	6.31E-01	5.83E-01	5.51E-01
		Std Dev	9.02E-01	8.35E-01	7.72E-01	**9.17E-02**	1.20E-01	1.25E-01	2.45E-01	1.70E-01
f_{23}	30	Mean	9.44E+00	1.03E+01	9.88E+00	4.43E-01	4.01E-01	4.58E-01	3.82E+00	**3.86E-01**
		Std Dev	1.22E+00	4.63E-01	8.29E-01	1.03E-01	**9.34E-02**	1.22E-01	3.71E+00	1.07E-01
	50	Mean	1.18E+01	9.46E+00	9.48E+00	6.32E-01	**6.14E-01**	6.62E-01	5.11E+00	8.14E-01
		Std Dev	1.41E+00	1.74E-01	1.93E-01	1.03E-01	**8.65E-02**	9.18E-02	2.85E+00	1.43E+00
f_{24}	30	Mean	3.87E+02	3.75E+02	3.76E+02	4.43E-01	**2.72E-01**	3.51E-01	1.86E+00	3.05E-01
		Std Dev	6.76E+01	3.64E+01	2.75E+01	2.57E-01	**5.38E-02**	1.88E-01	7.42E+00	1.29E-01
	50	Mean	6.90E+02	4.56E+02	4.57E+02	6.03E-01	5.66E-01	**4.59E-01**	1.56E+01	5.07E-01
		Std Dev	6.99E+01	1.74E+01	2.01E+01	3.74E-01	3.51E-01	**2.86E-01**	4.51E+01	3.37E-01
f_{25}	30	Mean	1.38E+07	8.77E+05	8.97E+05	6.51E+00	**6.22E+00**	1.12E+01	3.96E+01	7.33E+00
		Std Dev	1.10E+07	1.48E+05	1.29E+05	**1.14E+00**	1.55E+00	2.15E+00	8.00E+01	1.89E+00
	50	Mean	1.03E+08	1.93E+07	2.01E+07	1.87E+01	**1.86E+01**	3.02E+01	8.37E+04	2.25E+01
		Std Dev	5.03E+07	5.87E+06	5.69E+06	3.32E+00	3.46E+00	**3.15E+00**	3.27E+05	4.04E+00
f_{26}	30	Mean	1.42E+01	1.38E+01	1.39E+01	1.23E+01	1.19E+01	**1.17E+01**	1.31E+01	1.22E+01
		Std Dev	2.12E-01	2.72E-01	2.99E-01	6.24E-01	6.92E-01	**5.45E-01**	6.04E-01	6.60E-01
	50	Mean	2.41E+01	2.35E+01	2.36E+01	2.15E+01	2.10E+01	**2.08E+01**	2.27E+01	2.20E+01
		Std Dev	2.37E-01	2.77E-01	**2.30E-01**	5.81E-01	6.73E-01	7.41E-01	4.07E-01	7.33E-01

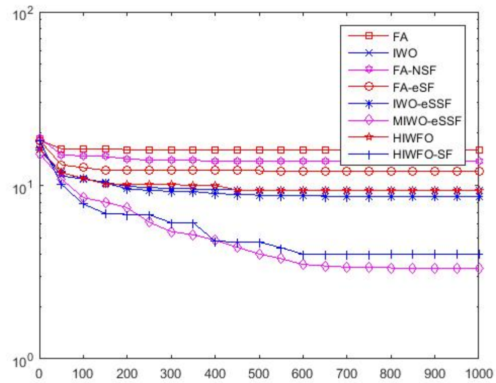
of HIWFO was competitive for functions, f_{12}, f_{13} and f_{24}. All the FA algorithms achieved decent results but seemed hard for it to converge. But, the proposed FA variants were able to get better values than their predecessor.



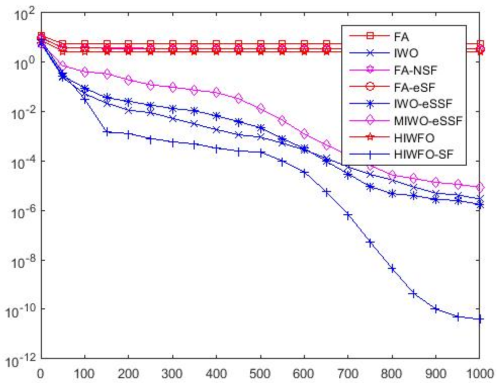
(a) f12



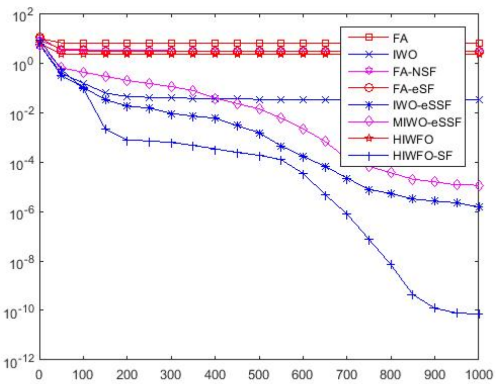
(b) f13



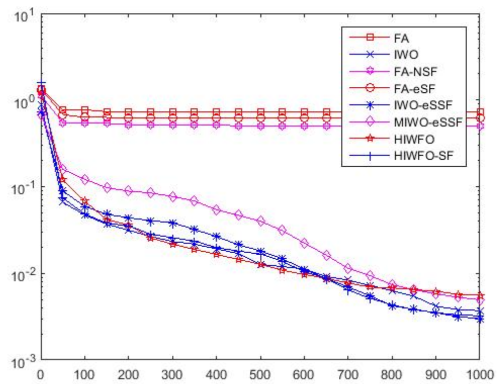
(c) f15



(d) f18



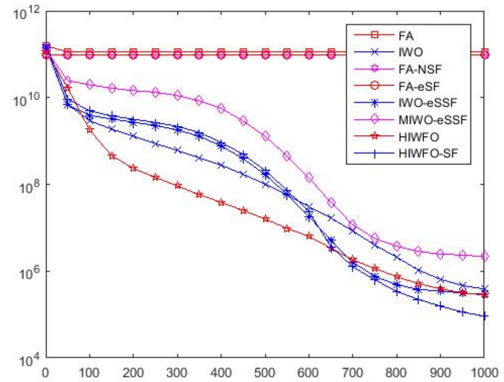
(e) f19



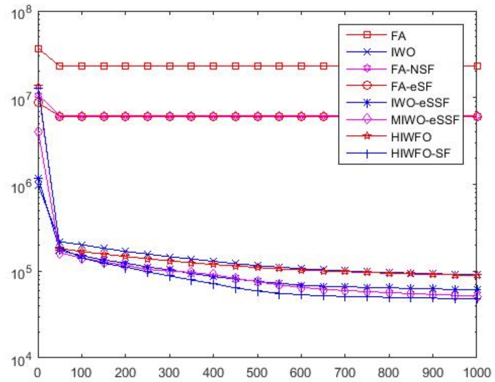
(f) f24

Figure 4.3: Convergence plots of 2-dimensional CEC2014 benchmark problems

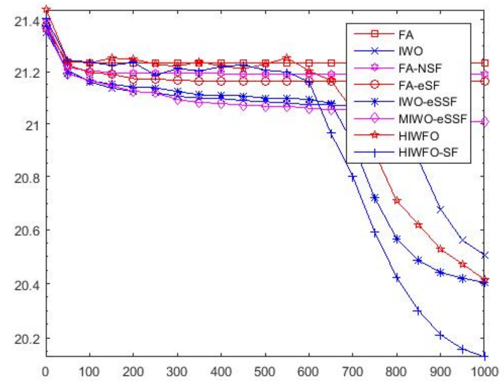
Based on observation in Figure 4.4, for less than 100 iterations, HIWFO, HIWFO-SF, IWO-eSSF and MIWO-eSSF showed faster convergence as the problem dimension increased to 30. After 100 iterations, the pattern continued as HIWFO-SF, IWO-eSSF and MIWO-eSSF converged even more to achieve better solution quality. After 200 iterations, for functions,



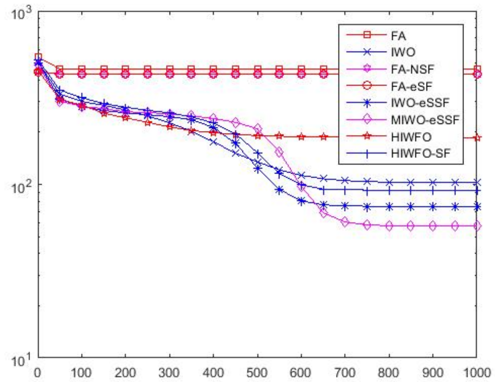
(a) f12



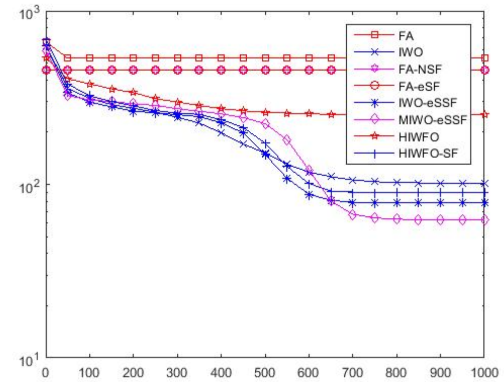
(b) f13



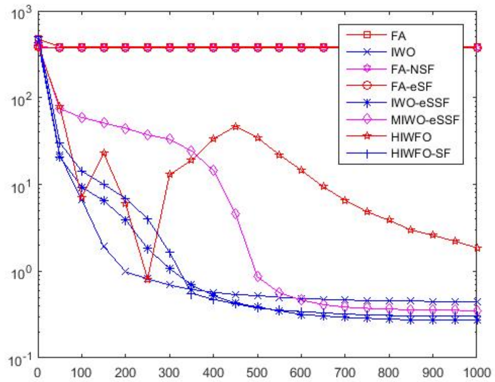
(c) f15



(d) f18



(e) f19



(f) f24

Figure 4.4: Convergence plots of 30-dimensional CEC2014 benchmark problems

f_{18} and f_{19}, HIWFO was unable to converge more and remained stuck at a local optimum. Furthermore, in function, f_{24}, HIWFO jumped toward higher local optimum. FA-NSF and FA-eSF showed better convergence quality as compared to FA, but, they also remained at local optimum and seemed hard for them to convergence.

It can be concluded that HIWFO-SF, IWO-eSSF and MIWO-eSSF outperformed other algorithms in solving CEC 2014 test functions. Not only achieving better accuracy, these algorithm have faster convergence in low and high dimensional optimisation problems.

Statistical Significant Test Result

In this section, Kruskal-Wallis non-parametric test is used for comparison study of CEC 2014 test results. The Kruskal-Wallis test is conducted based on 95% confidence interval and 30 simulation runs for each algorithm. The output results show the mean rank, rank number (bracket) and two-tailed p-value. The significant difference is considered if the probability value is less than $0.05 (p - value < 0.05)$. Tables 4.17 and 4.18 show the results of Kruskal-Wallis non-parametric test of the algorithms for the 16 CEC 2014 test functions.

As noted, for all the function with 4 different dimensions, the two-tailed $p - value$ was less than 0.05, which shows there were significant differences among the output results of all the algorithms. The test ranked the results in ascending order from minimum to maximum value. Hence, the lowest mean rank shows the significantly better result than others. Based on the observations from Tables 4.17 and 4.18, HIWFO-SF has shown the smallest mean rank and rank for functions f_{11}, f_{12}, f_{15}, f_{17} and f_{23} and MIWFO-SF dominated the performance for the function f_{11}, f_{18}, f_{19}–f_{21}. IWO-SF achieved better mean rank and rank for functions f_{22} and f_{26}. The overall performance can be evaluated by taking the average mean rank for all the test problems. As shown in the last column in Table **??**, HWIFO-SF scored at average 61.35, which is slightly better than IWO-eSSF (60.74) and MIWO-eSSF (78.62). Overall, it can be concluded that HIWFO-SF, IWO-eSSF and MIWO-eSSF performed better than other algorithms. Also, the proposed FA variants outperformed the original FA. In short, HIWFO-SF was the best performing algorithm compared with other algorithms for the unconstrained optimisation problems.

Table 4.17: The ranking of algorithms based on statistical significant test results for CEC 2014 test problems

	Dim	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF	p-value*
f_{11}	2	218.63 (8)	172.83 (6)	181.60 (7)	87.33 (4)	82.50 (3)	69.53 (2)	113.97 (5)	37.60 (1)	1.95E-34
	10	199.27 (8)	191.70 (6)	195.53 (7)	94.23 (5)	69.93 (3)	58.53 (1)	69.87 (2)	84.93 (4)	5.44E-34
	30	211.87 (8)	187.97 (7)	186.67 (6)	77.70 (5)	64.83 (2)	62.27 (1)	77.27 (4)	95.43 (3)	2.35E-34
	50	207.90 (8)	186.10 (6)	192.50 (7)	109.00 (5)	48.03 (1)	64.50 (2)	59.77 (3)	96.20 (4)	9.30E-37
f_{12}	2	215.63 (8)	185.27 (7)	182.80 (6)	91.53 (5)	79.47 (3)	87.30 (4)	76.97 (2)	45.03 (1)	1.12E-34
	10	186.60 (8)	195.00 (6)	204.90 (7)	90.03 (4)	67.30 (3)	135.50 (5)	66.47 (2)	18.20 (1)	1.14E-42
	30	211.33 (8)	185.80 (6)	189.37 (7)	101.30 (4)	65.83 (3)	135.50 (5)	59.30 (2)	15.57 (1)	3.09E-44
	50	223.83 (8)	180.50 (6)	182.17 (7)	105.03 (4)	64.80 (3)	135.50 (5)	53.43 (2)	18.73 (1)	1.96E-45
f_{13}	2	210.40 (8)	187.80 (7)	184.70 (6)	91.73 (5)	83.20 (4)	78.93 (2)	81.93 (3)	45.30 (1)	5.40E-34
	10	195.73 (8)	195.33 (7)	191.60 (6)	104.70 (5)	79.03 (3)	62.03 (2)	97.93 (4)	37.63 (1)	1.20E-35
	30	207.87 (8)	190.63 (7)	183.00 (6)	118.33 (5)	68.73 (3)	49.63 (2)	104.50 (4)	41.30 (1)	9.06E-38
	50	221.63 (8)	180.87 (6)	183.13 (7)	115.80 (5)	65.00 (3)	40.10 (1)	109.03 (4)	48.43 (2)	2.14E-40
f_{14}	2	207.10 (8)	186.80 (6)	192.60 (7)	85.77 (3)	59.40 (2)	114.87 (5)	97.17 (4)	20.30 (1)	2.40E-40
	10	192.97 (6)	193.63 (7)	199.90 (8)	42.87 (1)	83.57 (3)	85.70 (5)	80.03 (2)	85.33 (4)	9.97E-35
	30	215.17 (8)	182.77 (6)	188.57 (7)	25.93 (1)	83.57 (3)	88.93 (4)	80.37 (2)	98.70 (5)	5.44E-38
	50	223.07 (8)	181.47 (6)	181.97 (7)	27.60 (1)	79.43 (3)	76.80 (2)	105.73 (5)	87.93 (4)	7.38E-39
f_{15}	2	161.87 (8)	145.73 (7)	133.83 (6)	126.60 (5)	120.87 (3)	98.73 (2)	125.73 (4)	50.63 (1)	1.85E-08
	10	204.17 (8)	192.20 (7)	188.50 (6)	89.97 (4)	64.50 (2)	133.13 (5)	75.03 (3)	16.50 (1)	5.77E-42
	30	206.60 (8)	192.33 (7)	184.03 (6)	98.90 (4)	61.37 (2)	139.03 (5)	66.23 (3)	15.50 (1)	2.68E-43
	50	209.13 (8)	190.77 (7)	182.77 (6)	102.87 (4)	61.90 (2)	139.33 (5)	60.60 (3)	16.63 (1)	1.21E-43
f_{16}	2	199.50 (8)	180.80 (7)	179.13 (6)	74.80 (3)	70.60 (2)	107.63 (4)	116.93 (5)	34.60 (1)	6.94E-31
	10	196.93 (7)	197.43 (8)	191.47 (6)	50.87 (1)	54.50 (2)	61.03 (3)	132.90 (5)	78.87 (4)	9.84E-39
	30	212.20 (8)	184.60 (6)	189.70 (7)	44.30 (2)	60.93 (3)	42.27 (1)	135.50 (5)	94.50 (4)	1.04E-41
	50	212.40 (8)	189.97 (7)	184.13 (6)	31.50 (1)	57.17 (2)	53.33 (3)	135.47 (5)	100.03 (4)	1.15E-42
f_{17}	2	201.13 (8)	195.07 (7)	190.23 (6)	70.87 (2)	72.60 (3)	100.60 (5)	73.93 (4)	59.57 (1)	3.56E-34
	10	182.07 (6)	203.03 (8)	185.67 (7)	54.57 (2)	23.27 (1)	70.37 (3)	116.37 (4)	128.67 (5)	1.69E-38
	30	201.97 (8)	193.00 (7)	189.00 (6)	88.97 (3)	54.03 (2)	94.50 (4)	126.60 (5)	15.93 (1)	1.17E-41
	50	221.20 (8)	178.57 (6)	185.07 (7)	83.93 (3)	38.57 (2)	111.50 (5)	119.80 (4)	25.37 (1)	2.87E-43
f_{18}	2	202.50 (8)	180.20 (6)	181.53 (7)	71.47 (3)	68.97 (2)	102.10 (4)	141.73 (5)	15.50 (1)	5.02E-38
	10	179.17 (6)	171.90 (5)	187.87 (8)	180.97 (7)	35.60 (2)	32.37 (1)	105.30 (4)	70.83 (3)	1.99E-39
	30	207.97 (8)	188.77 (6)	189.77 (7)	86.77 (4)	53.80 (2)	28.93 (1)	133.50 (5)	74.50 (3)	2.73E-41
	50	220.83 (8)	182.17 (6)	183.50 (7)	97.50 (4)	46.43 (2)	25.97 (1)	132.87 (5)	74.73 (3)	1.69E-43
f_{19}	2	208.47 (8)	176.60 (7)	175.20 (6)	72.70 (3)	58.93 (2)	97.13 (4)	159.47 (5)	15.50 (1)	7.27E-41
	10	196.90 (8)	194.30 (6)	195.30 (7)	75.00 (3)	48.47 (2)	44.33 (1)	134.63 (5)	75.07 (4)	9.30E-40
	30	219.27 (8)	182.07 (6)	185.17 (7)	82.93 (4)	55.20 (2)	34.10 (1)	135.50 (5)	69.77 (3)	1.09E-41
	50	224.90 (8)	180.67 (6)	180.93 (7)	94.03 (4)	42.93 (2)	34.77 (1)	135.47 (5)	70.30 (3)	2.00E-43

Table 4.18: continued from Table 4.17

	Dim	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF	p-value*
f_{20}	2	188.60 (8)	137.17 (5)	144.93 (6)	80.53 (2)	94.93 (3)	46.43 (1)	145.80 (7)	125.60 (4)	6.14E-16
	10	215.37 (8)	183.50 (6)	186.97 (7)	73.97 (3)	50.03 (2)	39.87 (1)	132.53 (5)	81.77 (4)	2.29E-40
	30	218.40 (8)	183.73 (6)	184.37 (7)	62.23 (3)	46.33 (1)	53.63 (2)	129.77 (5)	85.53 (4)	6.18E-40
	50	214.53 (8)	183.73 (6)	188.23 (7)	61.13 (3)	56.30 (2)	50.97 (1)	126.90 (5)	82.20 (4)	1.01E-38
f_{21}	2	167.77 (7)	103.30 (4)	121.43 (5)	225.50 (8)	67.70 (2)	54.93 (1)	128.13 (6)	95.23 (3)	1.62E-25
	10	172.17 (7)	159.77 (5)	164.40 (6)	225.43 (8)	39.50 (2)	38.70 (1)	101.07 (4)	62.97 (3)	3.02E-42
	30	220.13 (8)	182.20 (6)	184.17 (7)	58.40 (2)	44.53 (1)	59.00 (3)	125.10 (5)	90.47 (4)	1.20E-39
	50	218.43 (8)	186.47 (7)	181.60 (6)	56.83 (2)	61.17 (3)	48.70 (1)	131.30 (5)	79.50 (4)	9.30E-40
f_{22}	2	213.47 (8)	181.43 (6)	191.60 (7)	81.67 (3)	68.70 (2)	118.33 (5)	93.30 (4)	15.50 (1)	2.67E-41
	10	220.17 (8)	188.70 (7)	177.63 (6)	51.57 (2)	41.40 (1)	109.40 (5)	91.47 (4)	83.67 (3)	2.07E-38
	30	208.67 (8)	190.00 (7)	187.83 (6)	34.13 (2)	45.77 (1)	109.70 (5)	99.03 (4)	88.87 (3)	3.57E-39
	50	221.23 (8)	183.30 (7)	181.97 (6)	38.00 (2)	43.57 (1)	111.50 (5)	92.60 (4)	91.83 (3)	7.46E-40
f_{23}	2	211.77 (8)	192.17 (7)	182.57 (6)	79.90 (3)	71.77 (2)	92.07 (4)	106.73 (5)	27.03 (1)	2.84E-38
	10	175.57 (6)	187.87 (8)	183.37 (7)	63.47 (3)	51.43 (2)	77.23 (4)	173.77 (5)	51.30 (1)	1.34E-35
	30	178.80 (8)	205.20 (7)	192.03 (6)	75.03 (3)	60.07 (2)	76.80 (4)	125.57 (5)	50.50 (1)	1.50E-34
	50	224.77 (8)	178.30 (6)	181.07 (7)	68.17 (3)	61.37 (2)	79.43 (4)	123.93 (5)	46.97 (1)	1.82E-38
f_{24}	2	207.97 (8)	190.63 (7)	187.90 (6)	77.70 (3)	62.20 (2)	99.77 (5)	85.33 (4)	52.50 (1)	4.64E-35
	10	182.20 (6)	198.17 (7)	206.13 (8)	50.93 (1)	56.23 (2)	76.83 (4)	127.87 (5)	65.63 (3)	2.54E-38
	30	197.77 (8)	196.10 (7)	192.63 (6)	89.53 (5)	59.67 (1)	81.37 (4)	76.83 (3)	70.10 (2)	1.32E-33
	50	225.50 (8)	179.03 (6)	181.97 (7)	80.50 (5)	76.80 (4)	73.50 (2)	75.03 (3)	71.67 (1)	9.40E-35
f_{25}	2	208.47 (8)	195.17 (7)	182.33 (6)	84.12 (3)	61.88 (2)	109.02 (5)	93.53 (4)	29.48 (1)	1.32E-38
	10	193.57 (8)	196.10 (6)	196.83 (7)	61.97 (2)	55.27 (1)	89.57 (4)	99.03 (5)	71.67 (3)	1.05E-34
	30	224.73 (8)	180.20 (6)	181.57 (7)	44.40 (2)	40.73 (1)	113.37 (4)	119.03 (5)	59.97 (3)	3.63E-42
	50	225.50 (8)	179.67 (6)	181.33 (7)	37.67 (2)	37.20 (1)	107.53 (4)	131.37 (5)	63.73 (3)	3.72E-44
f_{26}	2	212.10 (8)	189.30 (7)	185.10 (6)	75.73 (4)	68.95 (2)	106.15 (5)	73.67 (3)	53.00 (1)	1.97E-35
	10	211.07 (8)	186.90 (7)	180.17 (6)	86.77 (4)	65.13 (2)	26.17 (1)	124.97 (5)	82.83 (3)	8.55E-38
	30	214.97 (8)	177.70 (6)	184.70 (7)	82.20 (4)	54.77 (2)	45.50 (1)	130.40 (5)	73.77 (3)	7.13E-37
	50	222.33 (8)	180.83 (6)	180.90 (7)	68.20 (3)	48.47 (2)	39.63 (1)	130.00 (5)	93.63 (4)	1.01E-40
Average		206.47	183.77	183.74	81.63	60.74	78.62	107.68	61.35	
Average (Rank)		7.8	6.44	6.58	3.41	2.16	3.03	4.22	2.39	

4.3 Constrained Optimisation Problems

Generally, a constrained optimisation problem is best described as follows:

$$Minimize f(\overrightarrow{x}), \overrightarrow{x} = [x_1, x_2, \cdots, x_n] \tag{4.4}$$

subject to

$$g_i(x) \leq 0, for\ i = 1, \cdots, q$$

$$h_j(x) = 0, for\ j = 1, \cdots, m$$

However, for equality constraints handling, the equations are transformed into inequalities of the form

$$|h_j(x)| - \varepsilon = 0, for\ j = 1, \cdots, m \tag{4.5}$$

where a solution of $\overrightarrow{x}$ is regarded as feasible solution if and only if $g_i(x) \leq 0$ and $|h_j(x)| - \varepsilon = 0$ with ε is a very small number. The presence of constraints in any optimisation problem may have significant effect on the performance of the optimisation algorithm. In this work, penalty function method is used to solve the constrained optimisation problem. This method is easy to implement and is often chosen due to its simplicity (He and Wang, 2007). In this method, the problem is solved much simpler by transforming the constrained optimization problem to unconstrained problem.

4.3.1 Constraint-handling Mechanism

Generally, constraint-handling techniques can be divided into five major groups, namely; penalty functions, special representations and operations, repair algorithms, separation of objectives and constraints and hybrid methods (Coello, 2002). Due to its simplicity, the penalty function method has been considered as the most popular technique to handle problem-specific constraints (Gandomi et al., 2011; Kaveh and Talatahari, 2010). In this work, penalty function method is used as a constraint handling mechanism to solve the constrained optimisation problem. The penalty function method is a popular method used as compared to most traditional algorithms that are usually based on the concept of gradient. This method is easy to implement and is often chosen due to its simplicity (He and Wang, 2007). In the transformation to unconstrained problem, a certain value is added to the objective function based on the constraint violations. Actually, the constraint boundaries act as barriers during the process of optimisation search (Rao and Rao, 2009).

The most adopted approach for handling constraints is the penalty function approach. According to Arora (2004), this method is simple, has the ability to handle nonlinear constraints and also can be used with unconstrained methods. However, the method requires several preliminary trials and is very sensitive to the choice of the associated penalty parameter (Arora, 2004).

In general, the method can be characterized by internal and external types of penalty function. For the external penalty method, the search process starts with an infeasible individual. According to Jorhedi (2015), due to the penalty effects, these individuals are attracted to feasible regions in the search space. Jorhedi (2015) states that for internal type, a penalty function is defined whose values at points away from constraint boundaries are small and tend to infinity when the constraint boundaries are approached. Thus, during the process, if it starts with a feasible solution, the generated individuals are later all within the feasible solution region. In this work, the exterior penalty approach is used (Fogel, 1995). The general form of a penalized objective function is given as:

$$\varphi(x) = f(x) + \left[\Sigma_{i=1}^{i=m} K_i H_i + \Sigma_{i=1}^{i=m} C_i G_i\right] \tag{4.6}$$

where $\varphi(x)$ is an expanded objective function. $H_i = |h_i(x)|^\gamma$ and $G_j = max\{0, g_j(x)\}^\beta$; γ and β are commonly set as 1 or 2. K_i and C_j are called penalty factors. The right-hand bracket is referred to as penalty function.

The value of penalty factor in the setting is a concern. If it has a low value, the search effort will be heavier on the infeasible region and the feasible region is not explored accordingly. However, if the penalty factor has a high value, the infeasible region is not explored efficiently. Thus, a lot of valuable information may not get extracted.

4.3.2 Experiments on Constrained Optimisation Problems

In order to assess the performance of the proposed algorithms on constrained optimisation problems, two categories of problems are considered in this study; a set of well-known benchmark functions of constrained problems and practical engineering problems. The tests will evaluate the efficiency, robustness and superiority on searching the global best value of the constrained optimisation problems.

Constrained Benchmark Functions

In this section, a brief summary of the selected benchmark problems is given. The first category includes ten well-known benchmark functions of constrained optimisation problems; Problem 1, f_{g01} is a well-known constrained benchmark problem introduced by Bracken and McCormick (1968). The objective function optimum solution of this constrained minimization problem is noted at $f(x^*) = 1.393454$ located at $x^* = (0.82288, 0.91144)$. It has previously been tested with well-known metaheuristic algorithms such as GA (Homaifar et al.,1994), evolutionary programming (EP) (D. B. Fogel, 1995), HS (Lee and Geem, 2005) and the mine blast algorithm (MBA) (Sadollah et al., 2013).

This research uses a selection of constrained benchmark functions of CEC 2006 test suite (Liang et al., 2006). In this study, solutions to constrained problems are investigates with their respective objective functions and various types and nature with various number of design variables. These main characteristics include linear inequality (LI), non-linear inequality (NL), linear equality (LE) and nonlinear equality (NE). It also includes the number of design variables (d, dimensions) and the type of the problem. The objectives and constraints of the selected functions have different characteristics such as linear, nonlinear, quadratic, cubic and polynomial as shown in Table 4.19.

Table 4.19: The characteristics of the constrained benchmark problems

$f^*(x)$	Dim	Type	*LI*	*NI*	*LE*	*NE*
f_{g01}	2	Quadratic	0	1	0	1
f_{g02}	5	Quadratic	0	6	0	0
f_{g03}	2	Cubic	0	2	0	0
f_{g04}	10	Quadratic	3	5	0	0
f_{g05}	2	Nonlinear	0	2	0	0
f_{g06}	7	Polynomial	0	4	0	0
f_{g07}	2	Quadratic	0	0	0	1
f_{g08}	3	Quadratic	0	0	1	1
f_{g09}	5	Nonlinear	4	34	0	0
f_{g10}	9	Quadratic	0	13	0	0

Note that d (dimensions) – the number of design variables (d, dimensions), LI – linear inequality, NL – non-linear inequality, LE – linear equality (LE) and NE – nonlinear equality.

Further numerical simulations were carried out based on carefully selected benchmark functions that are widely used in the literature (Liang et al., 2006). Nine benchmark functions were chosen from the CEC 2006 test suite (Liang et al., 2006) and used for this study. The mathematical formula and the respective constraints of each benchmark functions are provided in Appendix B.

Practical Engineering Constrained Problems

The performance of the proposed algorithms were also accessed with complex real world engineering problems, including five well-studied engineering design optimisation problems adopted from the literature. The problems considered are a welded beam design problem (Rao and Rao, 2009), a tension / compression string design problem (Belegundu, 1983), a pressure vessel design problem (Kannan and Kramer, 1994), a speed reducer design problem (Sandgren, 1990) and gear train design problem. The main characteristics of the problems are shown in Table 4.20.

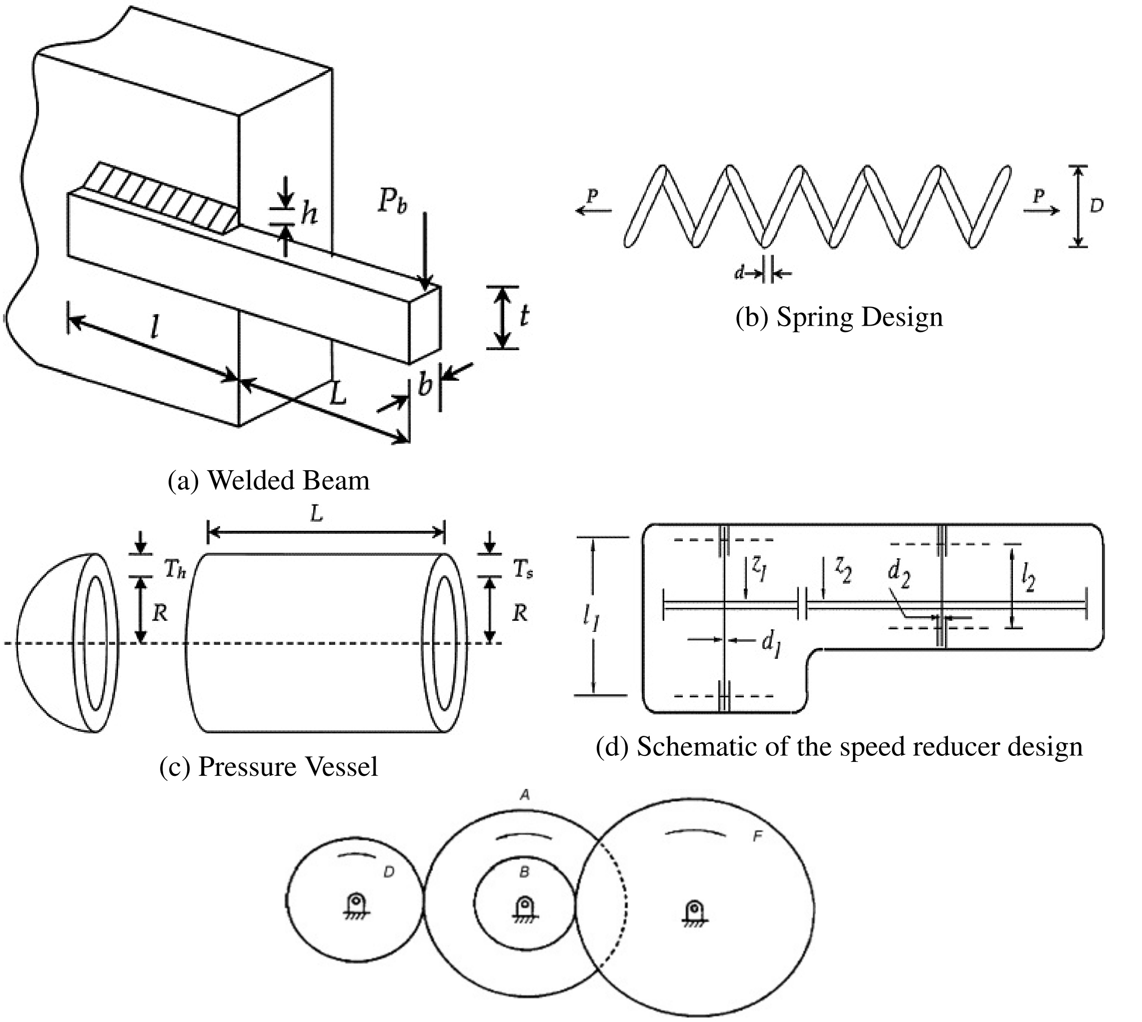
Table 4.20: The characteristics of the practical constrained problems

$f^*(x)$		Dim	Type	*LI*	*NI*	*LE*	*NE*
f_{e01}	Welded beam design	4	Nonlinear	2	5	0	0
f_{e02}	Pressure vessel design	4	Nonlinear	3	1	0	0
f_{e03}	Tension / Compression spring	3	Nonlinear	1	3	0	0
f_{e04}	Speed reducer design	7	Nonlinear	4	7	0	0
f_{e05}	Gear train design	4	Nonlinear	0	0	0	0

Welded beam design problem Figure 4.5a shows the welded beam structure that is often used as benchmark problem for testing optimisation methods with constrained problems where it was first described by Coello (2000). The problem is designed to find the minimum cost f(x) of fabrication of the welded beam subject to constraints on bending stress in the beam (θ), end deflection of the beam (δ), shear stress (τ), buckling load on the bar (P_b) and side constraint. As stated by Lui et al. (2010) and Zhou et al. (2013), four design variables, namely thickness of the weld ($h(x_1)$), the length of the welded joint ($l(x_2)$), the width of the beam ($t(x_3)$) and the thickness of the beam ($b(x_4)$) are to be considered.

Tension / compression spring design problem The tension / compression spring design as shown in Figure 4.5b is also one of the practical engineering benchmark problems. The problem is well described by Belegundu and Arora (1985) and Arora (2004), where the design is to minimize the weight of a tension / compression spring subject to constraints on minimum deflection, surge frequency and shear stress. The design variables of this problem are the mean coil diameter, $D(x_1)$, the wire diameter, $d(x_2)$ and the number of active coils, $N(x_3)$.

Pressure vessel design problem The pressure vessel problem (Figure 4.5c) is a practical problem that is often used as benchmark problem for testing optimisation methods. The objective of pressure vessel design problem is to find the minimum total cost of fabrication, including costs from a combination of welding, material and forming. Thickness of the cylindrical skin, $T_s(x_1)$, thickness of the spherical head, $T_h(x_2)$, the inner radius, $R(x_3)$, and the length of the cylindrical segment of the vessel, $L(x_4)$ were included as the optimisation design variables of the problem.



(a) Welded Beam

(b) Spring Design

(c) Pressure Vessel

(d) Schematic of the speed reducer design

(e) Schematic of the gear train design

Figure 4.5: Practical engineering design problems

Speed reducer design problem The speed reducer problem (Figure 4.5d) is part of the gear box of mechanical system, and is used as one of the practical benchmark problems because it involves seven design variables (Lin et al., 2013). In this problem, the objective is to minimize the weight of speed reducer subject to its constraints. According to Sadollah et al. (2013), the constraints of the design problems are bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. Detailed information have been further discussed in Mezura-Montes and Coello (2005) and (Sadollah et al., 2013).

Gear train design problem The gear train design problem (Figure 4.5e) is targeted to get the minimum cost of the gear ratio of the gear train. For this problem, the constraints are limits on design variables (side constraints). Design variables to be optimised are in discrete form since each gear has to have an integer number of teeth. Constrained problems with discrete variables may increase the complexity of the problem (Sadollah et al., 2013). The value of x_1, x_2, x_3, and x_4, represent the design variables of n_A, n_B, n_D and n_F, respectively. The integer design variables are bounded between 12 and 60.

The mathematical formulation of the cost function for all the practical engineering optimisation problems are shown in Appendix B with their respective constraint functions and variable regions.

4.3.3 Parameter Set Up and Performance Measurement

The proposed algorithms are tested with the problems and the results are compared with IWO and FA algorithms. For fair comparison of performance of all the algorithms used in this section, the parameter setting for each algorithm is described as in Table 4.4.

In general, in order to evaluate the performance of the algorithms, the algorithm which requires less NFE to get the same best solution can be considered as better as compared to the other algorithms. The statistical simulation and comparison results with the mentioned algorithms are shown in Tables 4.21 – 4.28. The results listed include the best fitness value, the mean and worst value found, and the SD. Note that the statistical results are based on feasible solutions only.

4.3.4 Experimental Results and Performance Analyses

Constrained Benchmark Functions

Optimisation results of constrained benchmark problems are presented in this section. The algorithms are implemented to achieve the global optimum results, $f(x^*)$ and to satisfy all the constraint conditions. The algorithms use the same parameters such as population size of 30 and within 30,000 function evaluations (i.e 1000 iterations). It should be note that in these experiments, all the inequalities in the problem became equalities as mentioned in Liang et al. (2006). The best, standard deviation and the mean solution values are presented in Tables 4.21 and 4.22. The solutions achieved with the algorithms are compared with each other and with FA and IWO algorithms.

Comparative results listed in Table 4.21 are those achieved with FA, IWO and the proposed variants of FA and IWO. On the other hand, Table 4.22 shows comparison of results achieved by the proposed hybrid algorithms with FA and IWO. As noted in Table 4.21, IWO-eSSF and MIWO-eSSF outperformed the original IWO, FA and proposed FA variants for

Table 4.21: Results of FA and IWO variants on the constrained benchmark functions

$f(x)$		$f(x^*)$	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF
$g01$	Best	1.393	1.396	**1.393**	**1.393**	1.394	1.393	**1.393**
	Mean		1.424	**1.393**	**1.393**	1.397	1.394	**1.394**
	Std Dev		2.37E-02	2.12E-09	**2.65E-10**	2.43E-03	4.03E-04	3.75E-04
$g02$	Best	-30665.539	-30688.402	-30656.438	-30671.591	-30705.829	**-30665.529**	**-30665.5**
	Mean		-30850.685	-30603.828	-30639.651	-30705.828	**-30665.435**	**-30665.398**
	Std Dev		3.46E+02	1.05E+02	7.81E+01	3.40E-04	6.20E-02	6.60E-02
$g03$	Best	-6961.814	-7042.731	-6961.807	**-6961.814**	-6975.316	-6961.783	-6961.872
	Mean		-3877.144	-6007.39	**-6938.197**	-6975.177	-6960.89	-6960.821
	Std Dev		2.61E+03	1.71E+03	1.20E+02	1.32E-01	6.11E-01	7.04E-01
$g04$	Best	24.306	329.354	25.043	24.164	24.438	**24.305**	**24.304**
	Mean		583.104	28.662	23.441	24.657	**24.337**	24.376
	Std Dev		1.48E+02	7.09E+00	1.02E+00	1.59E-01	5.58E-02	9.17E-02
$g05$	Best	-0.096	-0.096	-0.096	**-0.096**	-0.096	-0.096	-0.096
	Mean		-0.096	-0.096	**-0.096**	-0.096	-0.096	-0.096
	Std Dev		5.00E-06	1.42E-14	**2.20E-15**	5.64E-07	6.66E-08	7.53E-08
$g06$	Best	680.63	877.712	680.628	680.634	680.637	680.645	**680.632**
	Mean		1908.994	680.808	680.836	680.611	680.582	**680.591**
	Std Dev		6.34E+02	2.28E-01	2.13E-01	5.55E-02	3.10E-02	4.79E-02
$g07$	Best	0.75	0.75	0.75	0.75	0.75	0.75	0.75
	Mean		0.75	0.75	0.75	0.75	0.75	0.75
	Std Dev		7.54E-05	3.30E-09	**5.34E-10**	1.59E-04	4.29E-06	1.95E-06
$g08$	Best	961.715	961.718	961.711	961.716	961.716	**961.715**	961.716
	Mean		961.857	966.418	961.825	961.915	**961.811**	961.821
	Std Dev		3.15E-01	2.43E+01	4.97E-01	4.45E-01	3.96E-01	3.40E-01
$g09$	Best	-1.905	-1.94	-1.907	-1.9	-1.911	**-1.905**	-1.905
	Mean		-1.661	-1.802	-1.904	-1.846	**-1.879**	-1.862
	Std Dev		3.44E-01	2.50E-01	3.35E-01	9.49E-02	5.41E-02	4.55E-02
$g10$	Best	-0.866	-2.672	-0.867	-0.868	-0.869	-0.866	**-0.866**
	Mean		-17.694	-0.824	-0.842	-0.871	-0.866	**-0.866**
	Std Dev		9.63E+00	9.34E-02	7.16E-02	6.46E-04	4.91E-04	6.38E-04

most of the problems. However, FA-eSF achieved better performance for problem $g06$ as the results were near to the optimal solution. Although all the algorithms achieved competitive results for problems $g08$ and $g11$, FA-eSF showed better robustness as the standard deviation was higher than those of other algorithms.

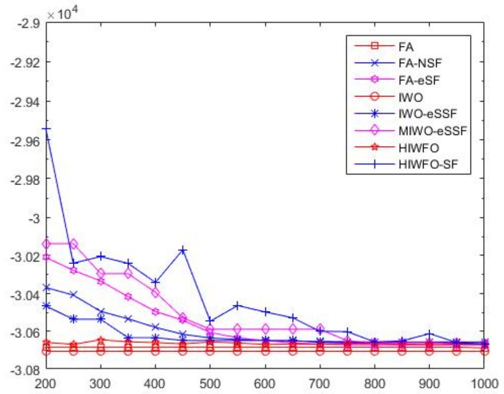
As shown in Table 4.22, the proposed hybrid algorithms (HIWFO and HIWFO-SF) performed better than FA and IWO algorithms. The performance of HIWFO algorithm was better than HIWFO-SF for $g03$, $g04$, $g06$ and $g10$. And, HIWFO-SF performed slightly better than the rest of the algorithms for $g01$, $g02$ and $g05$.

Comparison graphs of convergence rates are is shown in Figure 4.6. The graphs show the comparison between the algorithms for problems $g02$, $g03$, $g08$ and $g10$. Based on Figure 4.6, MIWO-eSSF algorithm took time to converge to the optimal solution. The fluctuation outcome of HIWFO-SF in problem $g02$ justified the SF mechanism that helped the algorithm

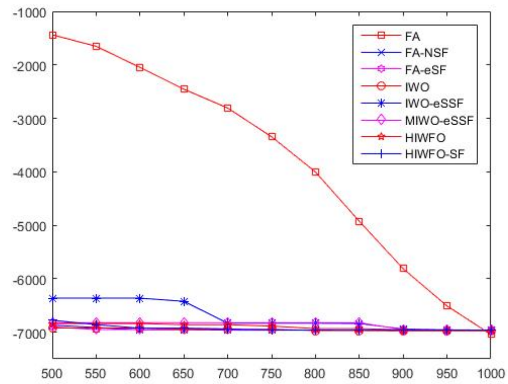
Table 4.22: Results of proposed hybrid algorithms on the constrained benchmark functions

$f(x)$		$f(x^*)$	FA	IWO	HIWFO	HIWFO SF
$g01$	Best	1.393	1.396	1.394	1.394	**1.394**
	Mean		1.424	1.397	1.397	**1.395**
	Std Dev		2.37E-02	2.43E-03	2.11E-03	2.52E-03
$g02$	Best	-30665.539	-30688.402	-30705.829	-30667.41	**-30665.641**
	Mean		-30850.685	-30705.828	**-30668.998**	-30647.069
	Std Dev		3.46E+02	3.40E-04	7.52E-01	3.99E+01
$g03$	Best	-6961.814	-7042.731	-6975.316	**-6961.8**	-6961.897
	Mean		-3877.144	-6975.177	**-6961.265**	-6959.903
	Std Dev		2.61E+03	1.32E-01	1.27E+00	1.58E+00
$g04$	Best	24.306	329.354	24.438	**24.322**	24.329
	Mean		583.104	24.657	**24.343**	24.697
	Std Dev		1.48E+02	1.59E-01	2.05E-01	5.74E-01
$g05$	Best	-0.096	-0.096	-0.096	-0.096	-0.096
	Mean		-0.096	-0.096	-0.096	-0.096
	Std Dev		5.00E-06	5.64E-07	8.04E-07	**5.95E-15**
$g06$	Best	680.63	877.712	680.637	**680.633**	680.643
	Mean		1908.994	**680.611**	680.712	680.915
	Std Dev		6.34E+02	5.55E-02	9.17E-02	3.47E-01
$g07$	Best	0.75	0.75	0.75	0.75	**0.75**
	Mean		0.75	0.75	0.75	**0.75**
	Std Dev		7.54E-05	1.59E-04	3.06E-05	**7.53E-06**
$g08$	Best	961.715	961.718	961.716	**961.716**	961.71
	Mean		961.857	961.915	**961.724**	962.506
	Std Dev		3.15E-01	4.45E-01	5.27E-02	1.31E+00
$g09$	Best	-1.905	-1.94	-1.911	-1.911	-1.884
	Mean		-1.661	-1.846	-1.84	-1.67
	Std Dev		3.44E-01	9.49E-02	9.13E-02	1.28E-01
$g10$	Best	-0.866	-2.672	-0.869	**-0.866**	**-0.866**
	Mean		-17.694	-0.871	**-0.816**	-0.798
	Std Dev		9.63E+00	6.46E-04	1.06E-01	1.20E-01

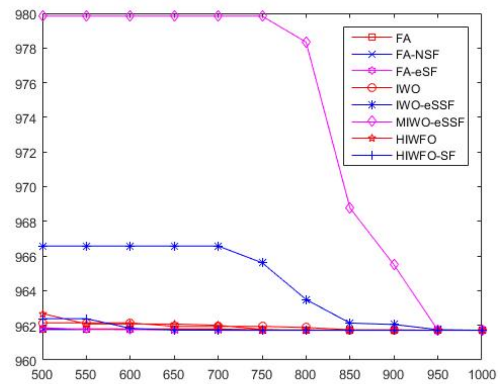
to re-adjust the algorithm to fine tune the solution. The IWO-eSSF and MIWO-eSSF also showed small fluctuation during the convergence.



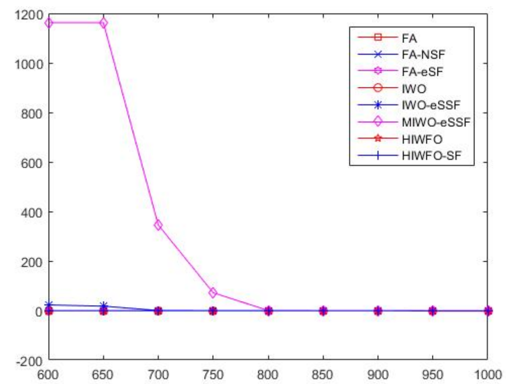
(a) f_{g02}



(b) f_{g03}



(c) f_{g08}



(d) f_{g10}

Figure 4.6: Convergence plots of constrained benchmark problems

Comparative results with other metaheuristics algorithms

The proposed algorithms have already been assessed in comparison to well-known meta-heuristics algorithm such as GA (Homaifar et al., 1994), HS (Lee and Geem, 2005) and the mine blast algorithm (MBA) introduced by A. Sadollah and his colleagues (Sadollah et al., 2013). The results shown in Table 4.23 were also compared with other algorithms; meta-heuristic inspired with particle swarm optimization, COPSO (Aguirre et al., 2007) and CVI-PSO (Mazhoud et al., 2013) and differential evolution, DECV (Mezura et al., 2010). The statistical simulation values obtained show the best objective value found. $f(x*)$ shows the optimal value of the problems and the results obtained near the optimal value are highlighted in bold in the table.

Table 4.23: Comparative results with constrained benchmark functions

	g01	g02	g03	g04	g05	g06	g07	g08	g09	g10
f(x*)	1.393	-30665.5	-6961.8	24.306	-0.096	680.63	0.75	961.715	-1.905	-0.866
FA	1.396	-30688.4	-7042.7	329.354	-0.096	877.712	0.75	961.718	-1.94	-2.672
IWO	1.394	-30705.8	-6975.3	24.438	-0.096	680.637	0.75	961.716	-1.911	-0.869
GA	1.434	NA	NA	NA	NA	NA	NA	NA	NA	NA
HS	1.377	NA	NA	NA	NA	NA	NA	NA	NA	NA
MBA	**1.393**	NA	NA	NA	NA	NA	NA	NA	NA	NA
COPSO	NA	**-30665.5**	**-6961.8**	24.3062	-0.0958	680.63	0.75	961.715	-1.905	-0.866
CVI-PSO	NA	-30665.8	**-6961.8**	24.4738	-0.1055	680.64	0.75	961.716	-1.905	-0.865
DECV	NA	-30665.5	**-6961.8**	24.306	-0.0958	**680.63**	0.75	961.715	-1.905	-0.866
FA-NSF	**1.393**	-30656.4	**-6961.8**	25.043	**-0.096**	**680.63**	0.75	961.711	-1.907	-0.867
FA-eSF	**1.393**	-30671.6	**-6961.8**	24.164	**-0.096**	**680.63**	0.75	961.716	-1.9	-0.868
IWO-eSSF	**1.393**	**-30665.5**	**-6961.8**	24.305	**-0.096**	680.65	0.75	**961.715**	**-1.905**	**-0.866**
MIWO-eSSF	**1.393**	**-30665.5**	-6961.9	24.304	**-0.096**	**680.63**	0.75	961.716	**-1.905**	**-0.866**
HIWFO	1.394	-30667.4	**-6961.8**	24.322	**-0.096**	**680.63**	0.75	961.716	-1.911	**-0.866**
HIWFO SF	1.394	-30665.6	-6961.9	24.329	**-0.096**	680.64	0.75	961.71	-1.884	**-0.866**

As noted in Table 4.23, IWO-eSSF achieved the best performance among the metaheuristics algorithms and very near to the optimal solution. It also outperformed other established state-of-the-art algorithms. MIWO-eSSF and HIWFO algorithms also showed better performance as compared with the mentioned state-of-the-art algorithms. It is noted that the results were close to the optimal values.

Practical Engineering Design Problems

The proposed algorithms were also used to solve well known practical engineering design problems such as pressure vessel design, spring design, welded beam, speed reducer and gear design problems. In these experiments, the penalty function approach is used to handle the constraints and to solve practical constrained engineering problems. All the problems are minimization problems. A total of 30 runs per algorithms were performed and the mean, standard deviation, minimum (best value) and maximum (worst value) of design problems are shown in Figure 4.5. The respective design variables of each design problem are also compared. As it can be seen in Tables 4.24 – 4.28, the approach was to find feasible solutions for all the problems. The overall results suggest that the proposed algorithms were able to provide competitive performance. As the experiments used the same set of parameters, the proposed algorithms seemed to be more stable as they gave lower reading of standard deviation. The lower reading of standard deviations also shows that the solution quality is consistent throughout the experiments.

Table 4.24: Results on the pressure vessel design problem

	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO SF
Best	17837.39	5896.4	5928.02	5963.69	**5890.39**	5891.63	5929.15	5948.1
x_1	1.89	0.78	0.79	0.79	**0.78**	0.78	0.79	0.8
x_2	1.11	0.39	0.39	0.39	**0.38**	0.39	0.39	0.4
x_3	62.74	40.64	41.08	40.87	**40.33**	40.33	40.62	41.59
x_4	65.79	195.65	190.41	194.27	**199.97**	199.97	196.12	183.96
$t, (sec)$	2.70	6.07	6.09	**1.46**	2.04	1.98	1.62	1.58
Mean	624843.6	6185.75	6262.21	6357.58	6137.64	**6063.1**	6598.59	6410.68
Std Dev	5.22E+05	2.54E+02	3.05E+02	3.65E+02	2.94E+02	**1.62E+02**	8.29E+02	3.54E+02
Min	17837.39	5896.4	5928.02	5963.69	**5890.39**	5891.63	**5929.15**	5948.1
Max	1862851.58	7006.15	7184.1	7198.79	6925.38	6458.05	**9310.08**	7130.31

Table 4.25: Result on the spring design problem

	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO SF
Best	1.27E-02	1.27E-02	1.27E-02	1.27E-02	1.27E-02	1.27E-02	1.25E-02	**1.17E-02**
x_1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	**0.05**
x_2	0.32	0.36	0.35	0.32	0.32	0.32	0.36	**0.34**
x_3	13.91	11.04	11.42	14.01	13.87	13.89	10.77	**10.71**
$t, (sec)$	2.70	5.74	5.57	1.36	1.27	1.18	1.41	**1.50**
Mean	1.31E-02	1.29E-02	1.32E-02	1.30E-02	1.27E-02	1.27E-02	1.25E-02	**1.24E-02**
Std Dev	3.86E-04	2.37E-04	9.83E-04	4.89E-04	1.42E-06	**3.18E-07**	8.81E-05	3.40E-04
Min	1.27E-02	1.27E-02	1.27E-02	1.27E-02	1.27E-02	1.27E-02	1.25E-02	**1.17E-02**
Max	1.39E-02	1.36E-02	1.70E-02	1.44E-02	**1.27E-02**	**1.27E-02**	1.30E-02	1.30E-02

Based on Table 4.24, IWO-eSSF algorithm provided the best performance in pressure vessel design problem. MIWO-eSSF algorithm showed comparable results and achieved more consistent results as compared with IWO-eSSF and other algorithms by showing the

Table 4.26: Results on the welded beam design problem

	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO SF
Best	1.74	1.67	**1.67**	1.72	1.72	1.7	1.68	1.68
x_1	0.24	0.30	**0.31**	0.21	0.21	0.26	0.28	0.33
x_2	2.90	2.18	**2.13**	3.29	3.4	2.63	2.39	1.93
x_3	9.08	9.03	**9.04**	9.05	9.04	9.08	9.04	9.08
x_4	0.21	0.21	**0.21**	0.21	0.21	0.21	0.21	0.21
$t, (sec)$	2.69	5.94	5.92	1.36	**1.29**	1.32	1.56	1.51
Mean	1.87	1.78	**1.72**	1.91	1.86	1.81	1.76	1.94
Std Dev	1.24E-01	1.30E-01	**6.84E-02**	2.06E-01	1.89E-01	1.70E-01	1.30E-01	2.65E-01
Min	1.74	1.67	**1.67**	1.72	1.72	1.7	1.68	1.68
Max	2.31	2.13	**1.95**	2.23	2.18	2.11	2.16	2.79

Table 4.27: Result on the speed reducer design problem

	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO SF
Best	2982.40	**2859.65**	2859.66	2978.25	2978.11	2978.16	2861.41	2859.75
x_1	3.46	**3.19**	3.19	3.45	3.45	3.45	3.19	3.18
x_2	0.7	**0.7**	0.7	0.7	0.7	0.7	0.7	0.7
x_3	17	**17**	17	17	17	17	17	17
x_4	7.44	**7.3**	7.3	7.3	7.3	7.3	7.37	7.3
x_5	7.68	**7.42**	7.43	7.67	7.68	7.67	7.42	7.42
x_6	3.34	**3.32**	3.32	3.35	3.35	3.35	3.32	3.32
x_7	5.27	**5.13**	5.13	5.27	5.27	5.27	5.14	5.13
$t, (sec)$	2.86	5.92	6.30	**1.48**	1.99	2.00	1.58	1.58
Mean	2989.65	**2862.03**	2862.70	2982.79	2979.16	2979.43	2866.67	2864.06
Std Dev	5.07	2.66	3.12	3.89	**1.64**	2.13	**4.21**	4.89
Min	2982.4	**2859.65**	2859.66	2978.25	2983.38	2985.21	2861.41	2859.75
Max	3000.64	**2868.36**	2868.71	2993.47	2978.11	2978.16	2879.29	2880.08

Table 4.28: Results on the gear design problem

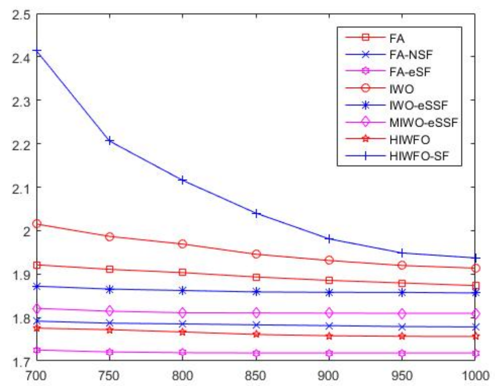
	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO SF
Best	1.02E-11	4.41E-22	**4.00E-27**	2.19E-19	3.77E-21	2.18E-20	2.21E-16	1.97E-24
x_1	24.91	49.91	**45.22**	46.77	59.21	55.73	57.38	56.65
x_2	13.93	18.82	**21.16**	24.06	42.56	40.14	20.97	15.58
x_3	14.53	15.59	**14.84**	15.67	12	12	19.97	29.07
x_4	56.31	40.74	**48.14**	55.88	59.78	59.91	50.59	55.42
$t, (sec)$	2.53	5.58	5.44	**1.20**	1.72	1.54	1.37	1.29
Mean	4.89E-03	2.84E-19	**6.34E-23**	3.79E-15	2.83E-16	3.06E-16	1.92E-14	1.08E-21
Std Dev	8.19E-03	3.60E-19	**1.98E-22**	5.45E-15	8.48E-16	5.57E-16	2.63E-14	1.71E-21
Min	1.02E-11	4.41E-22	**4.00E-27**	2.19E-19	3.77E-21	2.18E-20	2.21E-16	1.97E-24
Max	2.85E-02	1.23E-18	1.09E-21	2.55E-14	4.65E-15	1.97E-15	1.07E-13	**7.30E-21**

lowest value in mean and standard deviation. In the pressure vessel design problem, the proposed algorithms achieved better results compared to their predecessors.

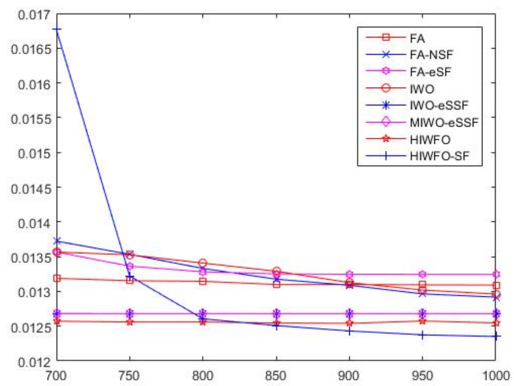
From Table 4.25, both hybrid algorithms (HIWFO and HIWFO-SF) achieved better performance in the spring design problem. The performance of the hybrid algorithms were also more robust as their mean, standard deviation were better compared to those of other algorithms. The results of all the algorithms for the welded beam problem are presented in Table 4.26. As noted, the variants of FA (FA-NSF and FA-eSF) were competitively better than HIWFO and HIWFO-SF algorithms. The proposed FA variants also showed better quality on the minimum and maximum value. However, both algorithms were the slowest in time taken to converge.

Table 4.27 also shows the same pattern as in Table 4.25. The proposed hybrid algorithms achieved better results compared to other algorithms in solving speed reducer design problem. However, it costed computational time for HIWFO and HIWFO-SF to solve the problem. Other proposed algorithms also achieved better results than FA and IWO algorithms.

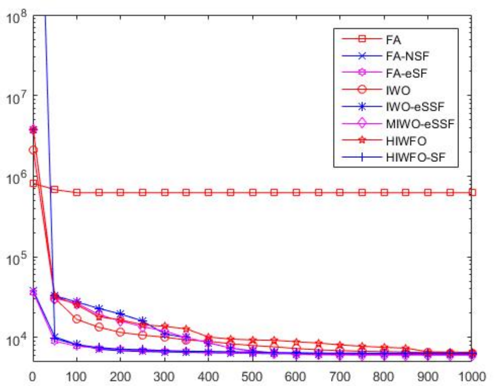
Table 4.28 shows the experimental results for gear train design problem. The statistical results show that HIWFO-SF outperformed other algorithms. The computational cost of the enhanced hybrid algorithm was also competitive as compared with other algorithms. The proposed FA and IWO variants showed good performance as compared to their respective predecessor algorithms, IWO and FA. It can be concluded that the proposed algorithms were able to explore the boundaries of feasible regions in each constrained problem to reach quality solution and better results.



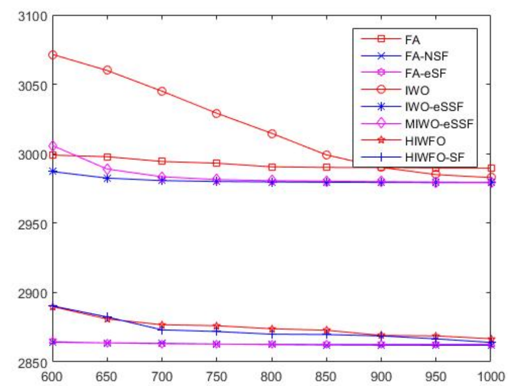
(a) Welded beam



(b) Spring design



(c) Pressure vessel



(d) Speed reducer

Figure 4.7: Convergence plots of practical engineering design problems

Figure 4.7 shows the convergence plot of practical engineering constrained problems, comparing the rate of convergence for the pressure vessel problem, welded beam, spring and speed reducer design problems for eight algorithms used in this experiment. As noted in Figure 4.7a, IWO-eSSF and MIWO-eSSF managed to converge faster than other algorithms although after reaching maximum iterations, HIWFO gave better solution quality. In 4.7b, all algorithms show fast convergence except HIWFO-SF algorithms. FA-eSF and HIWFO show better convergence value as compared with other algorithms. Figure 4.7c shows that HIWFO-SF had the slowest convergence, however, it managed to get better mean average value at the stopping point. In 4.7d, FA-eSF, FA-NSF, HIWFO and HIWFO-SF show faster convergence and better solution quality.

Comparative results with other metaheuristics algorithms

The performance of the proposed algorithms were also compared with four known state-of-the-art algorithms such as improved teaching-learning based optimization, ITLBO (Yu et al., 2014), improved ant colony optimization, IACO (Kaveh and Talahari, 2010), artificial bee colony, ABC (Akay and Karaboga, 2011), co-evolutionary particle swarm optimization, CPSO (He and Wang, 2007) and cuckoo search algorithm, CS (Gandomi et al, 2013). The results are shown in Table 4.29. These include the best fitness value and the standard deviation. Note that the statistical results are based on feasible solutions only.

Table 4.29: Comparative results of algorithms with practical engineering design problems

	Pressure vessel		Spring		Welded Beam		Speed	
	Best	Std dev	Best	Std dev	Best	Std dev	Best	Std dev
TLBO	6059.7	1.85E-12	0.01267	2.12E-06	1.725	6.77E-16	2994.471	4.62E-13
IACO	6059.7	6.72E+01	0.01264	3.49E-05	1.725	9.20E-03	NA	NA
ABC	6059.7	2.05E+02	0.01267	1.28E-02	1.725	3.12E-02	2997.058	0
CPSO	6061.1	8.65E+01	0.01267	5.20E-05	1.728	1.29E-02	NA	NA
CS	6059.7	5.03E+02	NA	NA	NA	NA	3000.981	4.96
FA-NSF	5896.4	2.54E+02	0.01270	2.37E-04	1.670	1.30E-01	**2859.650**	2.66
FA-eSF	5928.0	3.05E+02	0.01270	9.83E-04	**1.670**	6.84E-02	2859.660	3.12
IWO-eSSF	5890.4	2.94E+02	0.01270	1.42E-06	1.720	1.89E-01	2978.110	1.64
MIWO-eSSF	**5891.6**	1.62E+02	0.01270	3.18E-07	1.700	1.70E-01	2978.160	2.13
HIWFO	5929.2	8.29E+02	0.01250	8.81E-05	1.680	1.30E-01	2861.410	4.21
HIWFO SF	5948.1	3.54E+02	**0.01170**	3.40E-04	1.680	2.65E-01	2859.750	4.89

As noted in Table 4.29 the best feasible solutions found in the pressure vessel, spring and speed design problems by the proposed algorithms were better than those of other approaches with relatively small standard deviation, although TLBO showed better consistency with lower standard deviation.

It can also be seen that the best solution of the proposed hybrid algorithms were better than those of other mentioned methods in the spring design problems. Both algorithms achieved better best mean value compared to the other approaches.

Thus, it is clearly seen that the proposed IWO-NSSF and IWO-eSSF algorithms have good potential to solve various constraint problems.

4.4 Summary

In this chapter, overall performance of the proposed algorithms on single-objective optimisation problems are presented and compared. The proposed algorithms (FA-eSF, FA-NSF, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF) have also been compared with their predecessor algorithms, IWO and FA. Ten standard benchmark functions and 16 CEC 2014 test functions have been used to evaluate the algorithms on the unconstrained optimisation problems. The experimental results have been studied and the performance of the algorithms evaluated by using numerical results, convergence plot, success rate of the algorithm and statistical analysis by using Kruskal-Wallis tests.

The proposed algorithms have also been tested with constrained optimisation problems. The basic benchmark, nine CEC 2006 test functions and five practical engineering design problems were chosen to test and analyse the algorithms in the constrained problem domains. The practical engineering design problems include pressure vessel design, spring design, welded beam design, speed reducer and gain design problems which also constitute constrained optimisation problems.

From the analysis of unconstrained and constrained optimisation problems, the proposed algorithms especially IWO-eSSF, MIWO-eSSF and HIWFO-SF have achieved better performance than other algorithms investigated in the experiment. Although some algorithms performed better in different test functions, HIWFO-SF algorithm was more efficient than other proposed algorithms as well as FA and IWO. This is due to its capability to achieve better solution accuracy and faster convergence rate when solving single-objective optimisation problem. Based on the results obtained,

- For unconstrained benchmark problems, FA-NSF and FA-eSF have achieved far better solution quality in the various dimension sets used in these investigations. Moreover, IWO-eSSF and HIWFO-SF showed slight improvement as they outperformed FA and IWO algorithms. FA-eSF converged faster than all the algorithms used in the research. In terms of success rate and ranking based on the statistical test, FA-NSF, FA-eSF and HIWFO-SF performed better among the algorithms.

- For CEC 2014 benchmark problems, IWO-eSSF, MIWO-eSSF and HIWFO-SF achieved slightly better solution quality and faster convergence than other algorithms. Moreover, these algorithms showed the smallest mean rank in the statistical test. However, HIWFO appeared to suffer due to high randomisation value of FA elements in the algorithm during the search process.

- For constrained benchmark problems, IWO-eSSF, MIWO-eSSF and HIWFO performed better than other algorithms. They also outperformed the state-of-the-art algorithms as stated in Table 4.23.

- For practical engineering design problems, IWO-eSSF outperformed other algorithms in the pressure vessel design problem. FA-eSF, on the other hand, achieved better solutions in the welded beam and gear box design problems. FA-NSF achieved the best solution quality in the speed reducer problem and HIWFO-SF performed better than other algorithms in the spring design and showed competitive results in the welded beam and the speed reducer problems.

- FA-eSF showed competitive results in unconstrained problems and several practical problems. However, with more complex problems such as CEC 2014 and constrained benchmark problems, IWO-eSSF and MIWO-eSSF showed good optimisation potential.

.

The proposed algorithms are further tested and evaluation with multi-objective optimisation problems in subsequent chapters of the thesis.

Chapter 5

Multi-objective adaptive firefly and invasive weed optimisation algorithms

5.1 Introduction

This chapter presents evaluation of the proposed algorithms in solving multi-objective (MO) optimisation problems. Some modifications and parameters adjustment of the algorithm are made to accommodate multi-objective problems. The proposed algorithms are evaluated and comparative assessment is made with other algorithms. Initially, the parametric study is carried out to determine the best condition to solve multi-objective problems. Two multi-objective problems are used in the study focusing on the implication of using different number of Pareto (NPareto) and different number of iterations. Eight multi-objective benchmark problems are then used to analyse the proposed algorithms and compare their performances with those of original FA and IWO algorithms. The performances of the algorithms are measured by the Pareto graph and three selected performance measurements. In these evaluations, the same experimental platform is used as mentioned in the previous chapter. In solving multi-objective optimisation tests, each problem is tested in 30 independent runs.

5.2 Multi-objective Optimisation Problem

In real engineering and science problems, optimisation applications usually comprise more than one objective. In such cases, the objectives usually conflicting with each another. There is no unique solution for these kind of problems. Therefore, the best trade-offs between objectives could be the best solution (Mirjalili and Lewis, 2015). Such problems are also called multi-objective problems. It means that according to the requirement of these tasks, there are a number of optimal non-dominated solutions to the problem. The set of solution is called Pareto optimal solution set, which represents the best trade-offs between the objectives (Mirjalili and Lewis, 2015). Hence, for any multi-objective optimisation problem, a possible of generic mathematical optimisation form (Yang, 2010a) can be represented as;

$$\underset{x\in\Re}{Minimize}\ f_i(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_p(\mathbf{x})] \tag{5.1}$$

subject to

$$\phi_i(x) \leq 0, for\ i = 1, \cdots, q$$

$$\psi_j(x) = 0, for\ j = 1, \cdots, m$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$ represent the decision variables. The above equation refers to minimization problem. Hence, the goal in equation (5.1) is to minimize all p functions $f_i(x)$, simultaneously. However, the definition of optimality of multi-objective must be defined. Definition of Pareto optimality in the multi-objective is as follows:

- Non-domination and domination – A point x^* is said to be dominated if the condition where a vector function $f_i(x^*) \leq f_i(x)$ for all $i \in [1, p]$ or at least one within $i \in [1, p]$ where x^* value is better than x at least one objective function or all objective function values. The value x^* is non-dominated if there is no x that dominates it.

- Pareto optimal point – A Pareto optimal solution x^*(also called Pareto point) is one value that is not dominated by any other x in the search space.

- Pareto optimal set – Also called Pareto set, P_s or set of Pareto optimal points is the set of all non-dominated values, x^*.

- Pareto front – Also called non-dominated set, P_f is the set of all function vectors $f(x^*)$ corresponding to the non-dominated solution or Pareto set, where

$$P_f = \{f(x^*) : x^* \in P_s \tag{5.2}$$

In single optimisation problem a single optimal solution is determined, whereas multi-objective optimisation leads to a number of solutions, called Pareto optimal set / solutions, P_s and the corresponding decision vectors are called non-dominated point or Pareto optimal point, x^*.

There are numerous approaches to solve multi-objective problems in the literature. There are several ways in which these methods can be characterised whether the methods are non-Pareto based or Pareto based evolutionary algorithms. Non pareto-based evolutionary algorithms, do not explicitly use the concept of Pareto dominance (Simon, 2013). However, the approaches still enable to find diverse Pareto-optimal set, for example, aggregation methods, goal attainment, vector evaluated genetic algorithm (VEGA), lexicographic ordering and $\epsilon-$constraint method. Another characterization is by Pareto-based evolutionary algorithms, which directly use Pareto dominance such as simple evolutionary multi-objective optimiser (SEMO), nondominated sorting genetic algorithm (NSGA) as well as $NGSA-II$, niched Pareto genetic algorithm (NPGA) and others. These approaches are well discussed by Marler and Arora (2004), Chinchuluun and Pardalos (2007) and Simon (2013). In this research, non-Pareto evolutionary algorithm of aggregation approach is used for all the algorithms.

5.3 Aggregation Approach for Solving Multi-objective Problems

In this research, aggregation approach is used to solve the multi-objective optimisation problems. Yang (2013) used aggregation approach by combining all objectives using the weighted sum method. That is by converting those objective functions into a scalar objective function. Chandrasekaran and Simon (2012) also used the same method to fine-tune optimal deviation in solving the unit commitment problems in a power system. It is one of the non-Pareto evolutionary algorithm which is also called weighted sum method (Zadeh, 1963). The basic function of the method is

$$\cup = \min_{x} \ \Sigma_{i=1}^{k} \omega_i f_i(x) \tag{5.3}$$

where $\Sigma_{i=1}^{k}\omega_i$ and usually all the weights, ω_i are positive values and the minimum of above is the Pareto optimal solution (Zadeh, 1963). Thus, $f_i(x)$ is combination of all the objectives of the problem or 'aggregation' into single objective. This method is computationally efficient, simple and easy to use (Coello, 2001; Marler and Arora, 2004). It has been widely applied in the literature and discussed by Chinchuluun and Pardalos (2007) and Marler and Arora (2010). In this method, the choice of weight representation is essential in order to obtain possible Pareto optimal solution. A lot of work of weighted sum approach has been discussed and reviewed by Marler and Arora (2010). Parsopoulos and Vrahatis (2002) has elaborated other weighted aggregation approaches such as Bang-bang weighted aggregation (BWA), dynamic weighted aggregation (DWA) and conventional weighted aggregation (CWA). Furthermore, Yang (2011; 2013) proposed random weight approach from a uniform distribution to solve multi-objective problems. By using this approach, the Pareto optimal front can be produced directly (Yang and He, 2013).

5.4 Multi-objective Benchmark Problems

In this analysis, eight well-known multi-objective benchmark problems are used to evaluate the proposed algorithms and the results are also compared with those of IWO and FA algorithms. The algorithms are tested on several multi-optimisation conditions such as in unconstrained and constrained optimisation problems as well as practical engineering design problems. Table 5.1 shows the multi-objective optimisation problems used in this experiment. The mathematical equations and related constraints (if any) are listed in Appendix B.

Table 5.1: Brief summary of the multi-objective benchmark problems

No	Functions	Optimisation problem	Variable bounds	Reference
MO1	Schaffer function 1 (SCH 1)	Unconstrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Knowles and Corne (1999)
MO2	Zitzler-Deb-Thiele's function (ZDT 1)	Unconstrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Zitzler et al. (2000)
MO3	Kursawe	Unconstrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Kursawe (1990)
MO4	CTP 1	Constrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Deb (2001)
MO5	Constr-Ex	Constrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Deb (2001)
MO6	Bihn and Korn	Constrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Binh and Korn (1997)
MO7	Chankong and Haimes	Constrained problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Babu and Gujarathi (2007)
MO8	Four bar plane truss	Engineering design problem	$x \in [-10, 10]^i$ $1 \leq i \leq 20$	Coello (2001)

5.4.1 Formulation of Multi-objective Optimisation Problems

In this section, the aggregation approach is adopted in the algorithms used in this research to solve multi-objective optimisation problems. The aggregation or weighted sum approach is set the same for the algorithms to evaluate their performances. The algorithms produced in Chapter 3 are extended to develop new multi-objective optimisation algorithms, as summarized in Algorithm 5.

As noted in Algorithm 5, the algorithm starts with the definition of the objective functions. If the problem is with constraints, constraint handling method is performed at this stage. After that, the population is randomly distributed in the search space within the boundary range. A number of Pareto fronts and iterations are pre-defined. The initial parameters of the algorithms used are also pre-determined.

The initial value of gbest* or initial Pareto solution is defined by the location of initial population. In the algorithm, a set of non-dominated solutions is kept aside for each set of weighted value. Then, the value of weights for each objective is applied. This is an important stage that has to update at each cycle after one complete iteration. The weights used in this experiment are systematically divided accordingly and their values depend on the number of non-dominated Pareto (NPareto).

Algorithm 5 Pseudo code of proposed multi-objective algorithm

Input: Objective function of $f(x_d)$, Pre-determined parameter; β_0, γ, α, variable boundary and population size N.
Output: Global minimum, elapsed time.
Generate initial population $x_i, (i = 1, \cdots, n)$ randomly, determine the light intensity, $I(x_d)$ based on individual fitness, $f(x_i)$.
while t, current iteration $t \leq$ maximum iteration **do**
  for all i to n **do**
    for all j to n **do**
      Evaluate the distance, r between two units (x_i, x_j) and their attractiveness via $e^{-\gamma r^2}$
      if $I_j > I_i$), move i towards j **then**
        Evaluate new solution x_{i+1} via equation **??**
      end if
    end for
  end for
  if x_{i+1} exceeds boundary **then**
    Set to its boundary
  end if
  Update light intensity, $I(x_d)$ based on the update location;
  Rank the fireflies and find the current best;
  Export global minimum and elapsed time;
end while

After that, iterations start with updating the value of gbest*. If the algorithm used is with SF mechanism, the adaptive parameters are updated. The algorithm is processes until the maximum number of iterations is reached. For each of the weight set per iteration, the gbest* value is noted and defined as direct Pareto front value. The overall results are processed and illustrated.

For this multi-objective algorithm, if the number of non-dominated Pareto is set high, it may cause increased computational cost. This is because, number of iterations and size of population of the algorithm also play an important role on the computational cost.

For fair comparison, the FA and IWO algorithms are also modified into multi-objective algorithms as described above. The set of weight are also set the same throughout the experiments to evaluate the performances of the proposed algorithms.

5.4.2 Constraint Handling in Multi-objective Problems

Many works relating to multi-objective problems with constraints have been reported in the literature. The research works dealing with equality and inequality constraints have been discussed in Efren (2009), Hughes (2001) and Kundu et al. (2011). In this experiment, a simple penalty technique is chosen for handling the constraints. In dealing with the solution of multi-objective problems with constraints, the absolute value of constraint violation is simply added to the individual objectives to increase the values of objectives (Kundu et al., 2011).

5.4.3 Performance Measurement

In this experiment, it is assumed that the Pareto front location is unknown. Based on this assumption, the performances of the algorithms are evaluated by the performance metrics and by graphical illustration of Pareto front. There is no best specific performance indicator in the multi-objective field (Mirjalili and Lewis, 2015). A lot of works in this field try to evaluate and compare qualitatively the obtained shapes of Pareto optimal fronts with performance metrics. As these metrics measure and evaluate properties of non-dominated solutions such as convergence and uniform distribution.

In this research, the selected performance metrics can be classified into two main criteria; convergence and coverage. The convergence criterion measures the closeness of solutions obtained to the true Pareto front (Rudolph, 1998; Mirjalili and Lewis, 2015). The coverage criterion defines how well the Pareto solution obtained covered the range of each of the objectives (Farhang-Mehr and Azarm, 2002; Mirjalili and Lewis, 2015). The hypervolume (HV) is selected here to evaluate the convergence criterion, and spacing (SP) and maximum spread (MS) are chosen to evaluate the coverage criterion. These metrics also involve uniform distribution and extensiveness (Jariyatantiwait and Yen, 2014) of the population during the search process in obtaining the non-dominated solutions of each multi-objective problems. The selected coverage and convergence performance metrics are explained below;

- Spacing, SP is a metric to measure whether the non-dominated solutions obtained are evenly distributed. SP is aimed to converge to zero implying that all the solutions are equally spaced. It was proposed by Schott (1995). The mathematical expression of SP (Schott, 1995) is as follows

$$SP = \sqrt{\frac{1}{n-1}\Sigma_{i=1}^{n}(\overline{d} - d_i)^2} \tag{5.4}$$

  where $\overline{d}$ is the average of d_i for all $1, 2, \ldots, n$ and n is the number of Pareto optimal solutions obtained.

- Maximum spread, MS measures the diversity of the Pareto optimal solution. This method was proposed by Zitzler (1999) and the mathematical equation (Zitzler, 1999) is as follows;

$$MS = \sqrt{\Sigma_{i=1}^{n} max(d(a_i, b_i))} \tag{5.5}$$

  where i is the number of objectives, and $d(a_i, b_i)$ calculates the Euclidean distance, a_i is the maximum value in the i-th objective, and b_i is the minimum value in the i-th objective. The value of MS will converge from zero to one as the obtained solution completely covered the true Pareto front.

- Hyper-volume metric, HV measures both convergence and distribution of Pareto set and has been discussed by Auger et al. (2009) and Jariyatantiwait and Yen (2014). This metric is for quantifying the convergence behaviour of MOEAs (Zitzler, 1999). The idea is to calculate the area/volume of the objective space that is dominated by the non-dominated Pareto optimal solutions obtained. Note that this performance indicator is also called size of space covered (SCC) in some references (Tan et al., 2002).

In all the multi-objective benchmark problem tests, the algorithms used the same population size, n and maximum number of iterations is set for a fair comparative evaluation. These were as follows:

- Maximum number of population, $n_{max} = 30$.
- Maximum number of iterations, $it_{max} = 1,000$ (NFE = 30, 000).

NFE is used in the experiments as a measure of computational time instead of the number of generations. The algorithms are terminated when NFE = 30,000 is reached. For a fair comparison of the algorithms, most of the parameters are set identical. The parameter set in Table 4.4 is used with the algorithms during initialization in all the tested problems.

5.5 Parameters and Their Impact on Accuracy and Convergence

In this section, the effect of number of Pareto and selection number of iteration with predetermined parameters of the algorithms used are studied. The impact on the accuracy of the Pareto optimal solution, convergence and coverage characteristics are observed and investigated. The simulation in this section provides a comparative assessment of performance of the proposed algorithms with those of their predecessors.

The multi-objective optimisation problems considered used in this study comprise an unconstrained optimisation (Kursawe problem) and a constrained optimisation (Constr-Ex problem). The properties of these problems are stated in Table 5.1 and their mathematical formulae in Appendix B.

Both problems are of minimization type. The performances of the algorithms are assessed by observation of the of Pareto front based on the Pareto set, P_s. The parameters chosen for the study are shown in Table 5.2

The performances of the algorithms with parameter changes are monitored and evaluated. The simulations with multi-objective optimisation using Kursawe and Constr-Ex problems were carried out and the impact of NPareto and it_{max} is discussed based on the results of Pareto front. The stopping criterion for the simulations was based on it_{max} value.

Table 5.2: Parameters to be studied

No.	Parameter	Symbol	Kursawe problem	Constr-Ex problem
1	Number of Pareto front	NPareto	20, 100 and 200	20, 100, 200
2	Number of iteration	it_{max}	20, 100 and 200	10 and 50

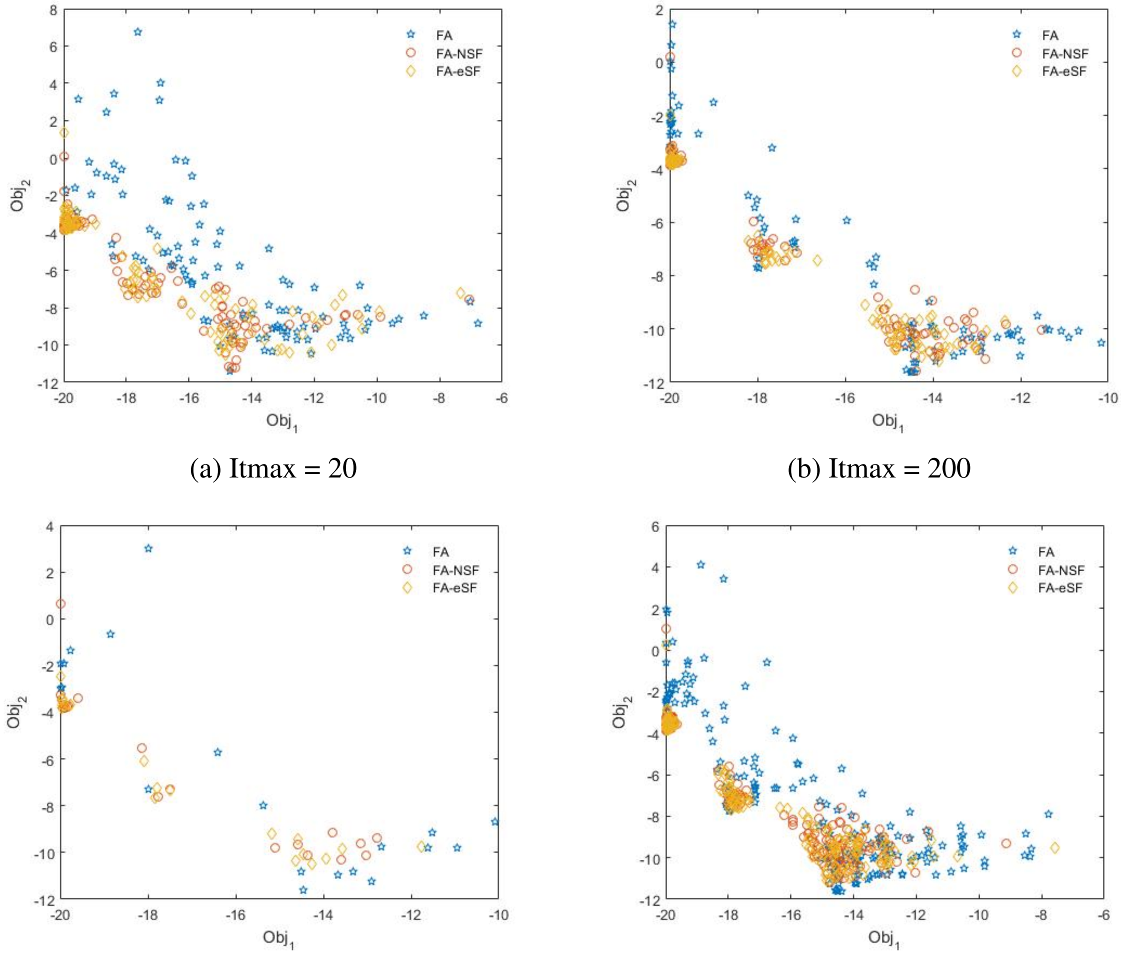
5.5.1 Kursawe Problem

In this experiment, Kursawe function is used to study the effect of NPareto and different ranges of iteration on the Pareto front of the algorithms used. The Kursawe function is a discrete convex unconstrained multi-objective problem. Figures 5.1, 5.2 and 5.3 show the results for FA, IWO variants and hybrid algorithms, respectively. For all the figure in this section, in sub-figure (a) and (b), the NPareto is fixed at 100. The iteration is set to 10 and 50 for sub-figures (a) and (b), respectively. On the other hand, for sub-figures (c) and (d), iteration is set to 50. The NPareto is set to 20 and 200 for sub-figures (c) and (d) respectively.

Figures 5.1a and 5.1b show that the distribution of trade-off points for FA is fairly uniform and scattered across the search space. The points are far from Pareto front. However, both proposed FA variants show good search space coverage and distribution of the Pareto front. This is because the Pareto optimal points of FA-NSF and FA-eSF are nearer to the Pareto front. By using different NPareto, Figures 5.1c and 5.1d show that the Pareto optimal sets of the proposed FA variants managed to get closer to the Pareto front and uniformly distributed. Again, in this case, FA shows the largest distribution range and scattered far from the Pareto front.

In Figure 5.2, the Pareto front of IWO variants are shown. Figure 5.2a shows that the MIWO-eSSF has achieved good distribution and very close to the Pareto front as compared to IWO-eSSF and IWO algorithms. When the iteration increased as in Figure 5.2b, all of the IWO variants achieved low distribution. However, there were a few of IWO and IWO-eSSF solution points scattered further away. Figure 5.2c shows the same pattern as in Figure 5.2b where they all converged to some point with low distribution on the Pareto front. When NPareto increased to 200 as in Figure 5.2d, the distribution of MIWO-eSSF improved showing better result.

Figure 5.3 shows performance results of the proposed hybrid algorithms. As noted in Figure 5.3a HIWFO and HIWFO-SF showed good distribution and very close to the Pareto front. However, as the iteration increased to 200 as in Figure 5.3b, their Pareto set concentrated on the objective points on Pareto front and had low distribution in the Pareto front. In lower NPareto and 100 iteration, as shown in Figure 5.3c, HIWFO achieved better distribution at the Pareto front and as NPareto increased to 200 as in Figure 5.3d, HIWFO-SF managed to improve the Pareto optimal solutions and performed better than HIWFO in term of Pareto front coverage.



(a) Itmax = 20 (b) Itmax = 200

(c) NP = 20 (d) NP = 200

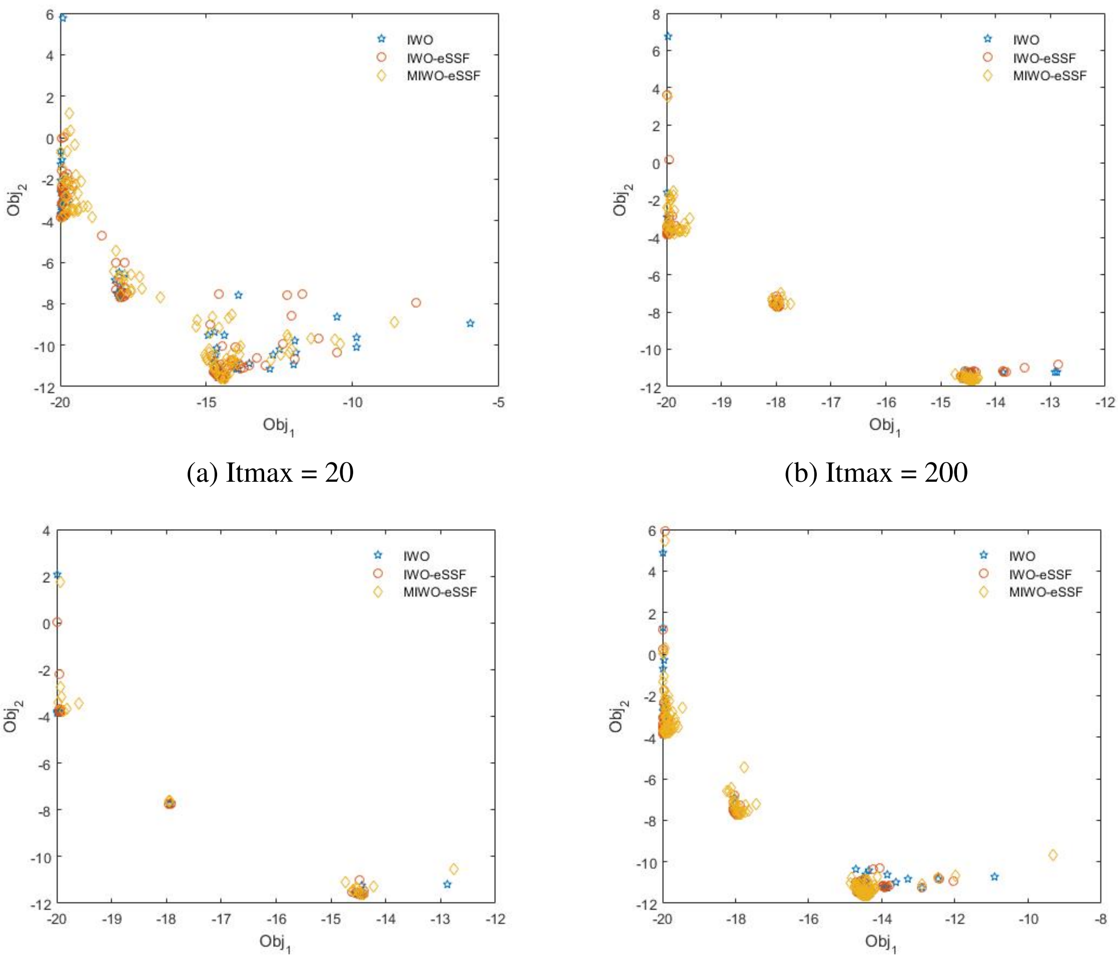
Figure 5.1: Analysis of MOFA for Kursave Problem

5.5.2 Constr-Ex Problem

In the objective space, the Constr-Ex problem is convex and uniformly distributed Pareto front is expected. This is an example of constrained multi-objective problem. In this experiment the use of different NPareto and iterations and their effect on performances of the algorithms. The results show the Pareto front of modified FA and IWO variants and modified proposed hybrid algorithm as in Figures 5.4, 5.5 and 5.6 respectively.

In sub-figures (a) and (b) of Figures 5.4, 5.5 and 5.6, the NPareto is fixed at 100, and the iteration is set to 10 and 50 for sub-figures (a) and (b), respectively. On the other hand, for sub-figures (c) and (d), iteration is set to 50, and the NPareto is set to 20 in sub-figure (c) and increased to 200 in sub-figure (d).

As noted in Figure 5.4a, the FA-NSF managed to achieve better trade-off distribution and nearer to the Pareto front than other FA variants. FA variants showed better results when the iteration increased to 100 as in Figure 5.4b, although a few Pareto optimal points of FA scattered far from Pareto front and not uniformly distributed. By fixing the iteration to 50 with lower NPareto and higher NPareto used as in Figures 5.4c and 5.4d, respectively, the



(a) Itmax = 20

(b) Itmax = 200

(c) NP = 20

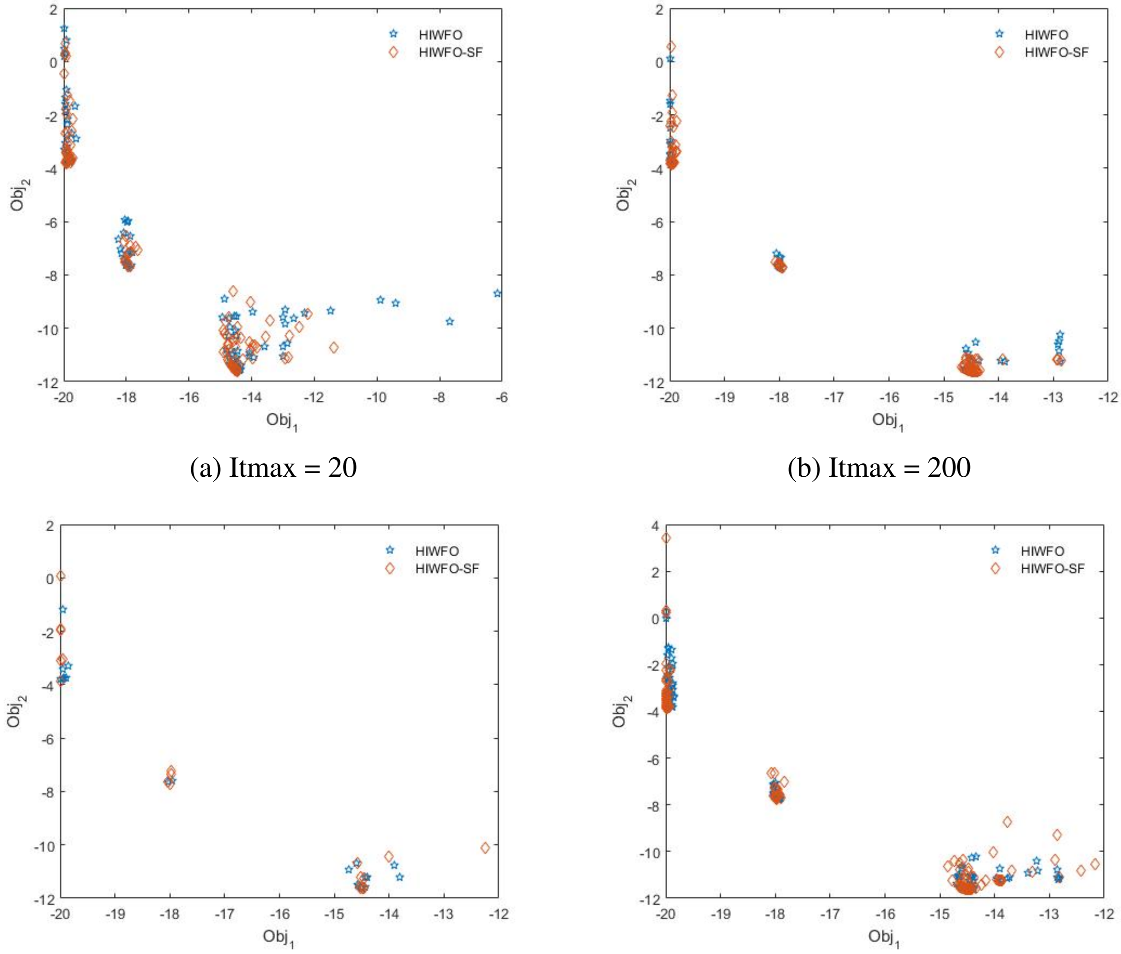
(d) NP = 200

Figure 5.2: Analysis of MOIWO for Kursave Problem

same pattern of Figure 5.4b was shown by the FA variants.

Figure 5.5 shows the performance results of IWO variants in solving Constr-Ex problem. As noted, both multi-objective IWO variants showed good distribution and their Pareto optimal sets were nearer to the Pareto front. Meanwhile, MIWFO-eSSF showed better distribution of trade-offs especially in low iterations as shown in Figure 5.5a. However, in low iteration (Figure 5.5a) and low NPareto shown in Figure 5.5c, IWO failed to achieve good distribution and performance as most of the points were scattered across the search area and far from the true Pareto front. As noted in Figure 5.5d, the IWO algorithm managed to improve the solution as the iteration and NPareto increased.

In Figure 5.6, performance results of both hybrid algorithms are shown. As noted, both HIWFO and HIWFO-SF algorithms achieved good and uniform distribution of solutions along the Pareto front. However, for low iterations, as noted in Figure 5.6a, few solutions of HIWFO-SF scattered far from the Pareto front. Other than that, they produced smooth Pareto sets which were near to Pareto front for all the cases as shown in Figures 5.6b, 5.6c and 5.6d.



(a) Itmax = 20

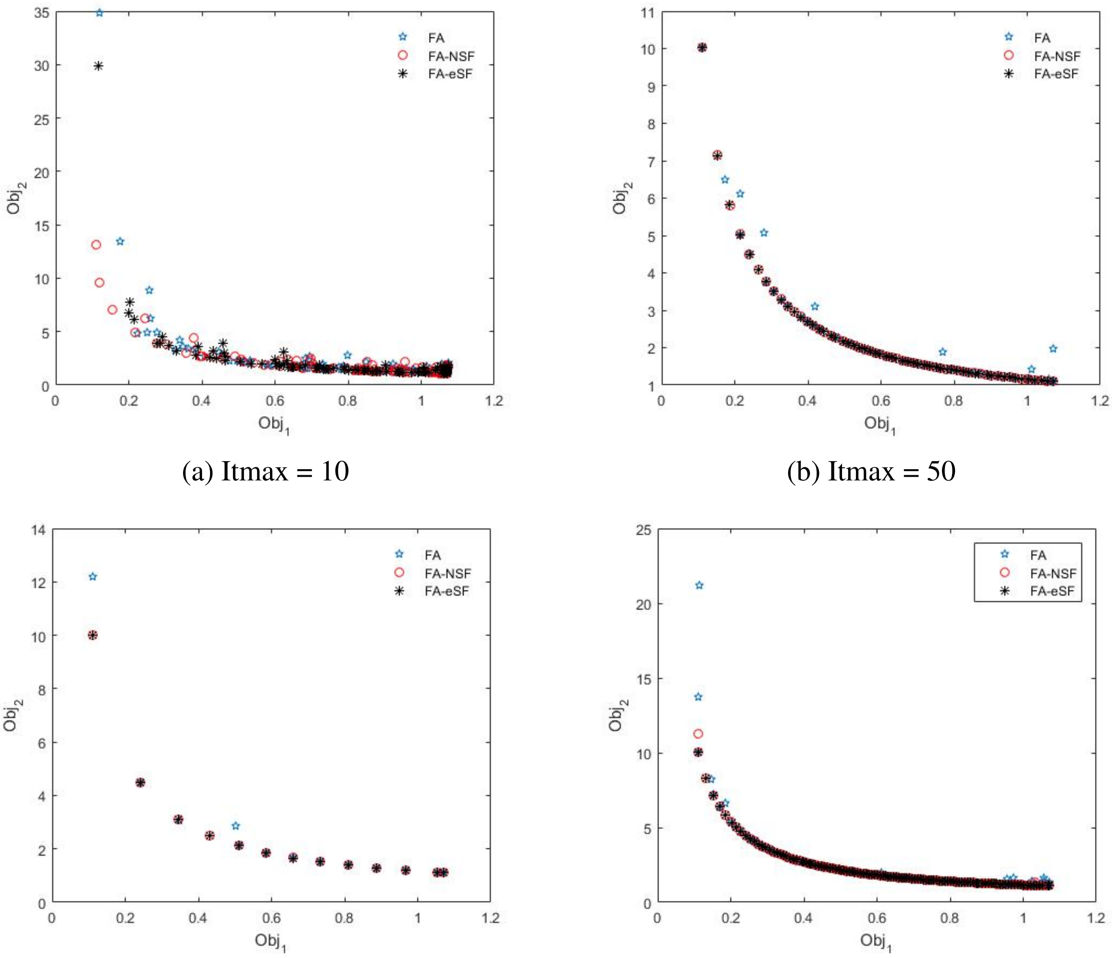
(b) Itmax = 200

(c) NP = 20

(d) NP = 200

Figure 5.3: Analysis of MO-HIWFO algorithms for Kursave Problem

Based on the study, it can be concluded that the modified algorithms of both proposed FA variants, HIWFO-SF, HIWFO and MIWO-eSSF for multi-objective problems show better performance than other algorithms. However, this study also indicated that large NPareto and suitable iteration value are needed for evaluation of performances of the algorithms. Thus, in evaluations presented in later sections, the NPareto and iteration will be set to more than 50 for a fair solution standard and quality.



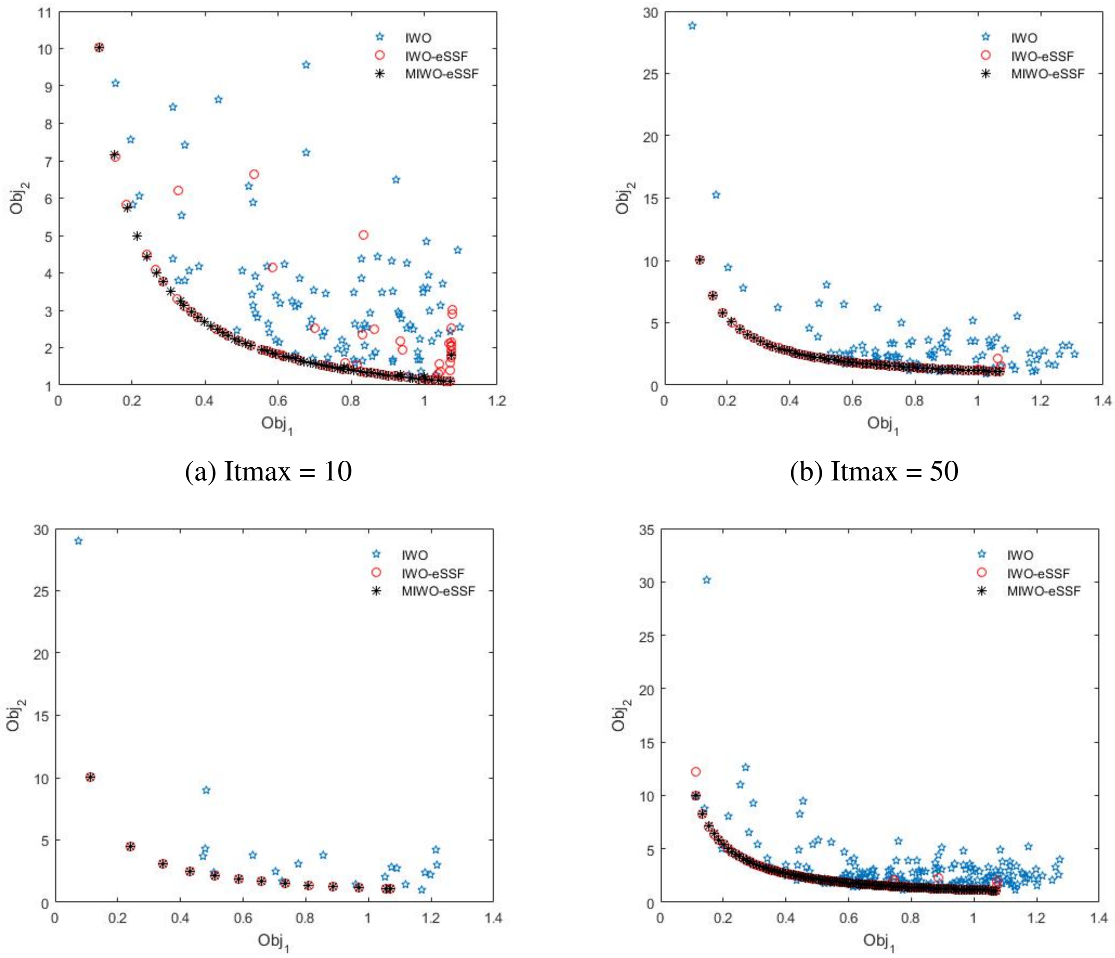
(a) Itmax = 10

(b) Itmax = 50

(c) NP = 20

(d) NP = 200

Figure 5.4: Analysis of MOFA for Constr-Ex Problem



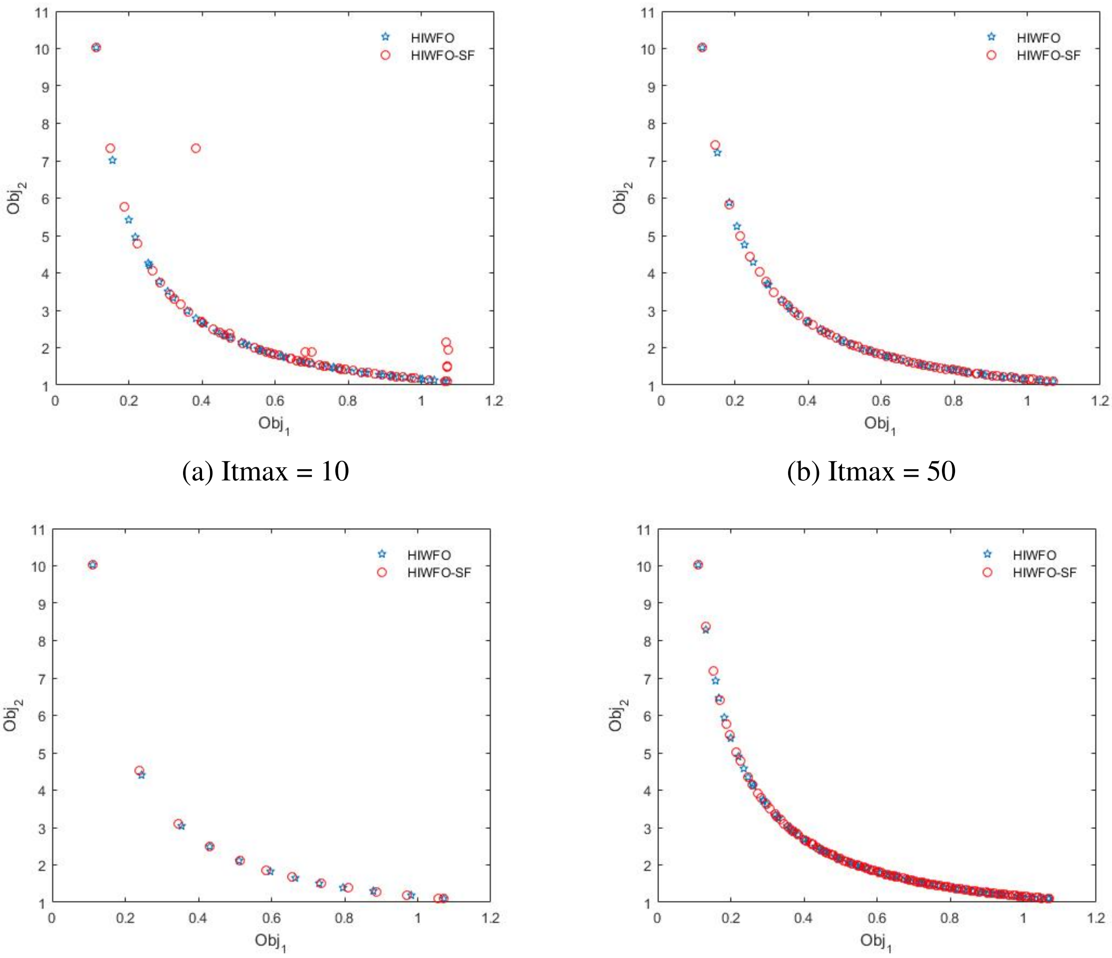
(a) Itmax = 10

(b) Itmax = 50

(c) NP = 20

(d) NP = 200

Figure 5.5: Analysis of MOIWO for Constr-Ex Problem



(a) Itmax = 10

(b) Itmax = 50

(c) NP = 20

(d) NP = 200

Figure 5.6: Analysis of MO-HIWFO algorithms for Constr-Ex Problem

5.6 Experiments for Multi-objective Optimisation Problems

This section presented performance results of the algorithms in solving multi-objective optimisation problems. The parameter setting and algorithms used in the experiments are described in the previous sections. The computational time and NFEs achieved by each algorithm are given in Table 5.3. The results assessed by performance metrics are shown in Tables 5.4, 5.5 and 5.6. The Pareto fronts of each algorithms for each problem are illustrated in Figures 5.7 − 5.14. The performances of the algorithms are evaluated based on the convergence and diversity of the Pareto optimality by assuming that the Pareto front of each problem is unknown. The convergence means that the result obtained is near to the true Pareto front. The diversity refers to the distribution of Pareto optimal solutions found within the range of solution search. The highlighted column in Table 5.3 designates competitive results in relation to other results.

Table 5.3: Computational time for multi-objective benchmark problems

Algorithm	Problem	MO1	MO2	MO3	MO4	MO5	MO6	MO7	MO8
FA	NFE	6000	1500	3000	1500	3000	1500	3000	3000
	$t, (sec)$	26.606	4.878	24.841	5.801	9.854	5.416	10.367	22.121
FA-NSF	NFE	1500	1500	1500	1500	1500	1500	1500	1500
	$t, (sec)$	6.082	5.953	12.995	5.517	5.322	5.77	5.713	11.198
FA-eSF	NFE	3000	1500	1500	1500	1500	1500	3000	1500
	$t, (sec)$	12.384	5.83	13.366	5.445	5.416	5.862	5.609	11.047
IWO	NFE	6000	3000	3000	3000	3000	1500	3000	**3000**
	$t, (sec)$	5.424	4.175	13.952	4.336	4.335	2.533	4.943	**8.195**
IWO-eSSF	NFE	1500	**900**	3000	1500	3000	1500	3000	3000
	$t, (sec)$	6.428	**2.333**	14.224	3.649	6.631	3.545	6.898	9.87
MIWO-eSSF	NFE	**1500**	1500	3000	1500	**1500**	3000	**3000**	3000
	$t, (sec)$	**3.42**	3.203	7.258	3.033	**2.931**	5.677	**2.855**	10.612
HIWFO	NFE	4500	1500	3000	1500	3000	1500	1500	3000
	$t, (sec)$	10.632	3.383	15.918	3.494	6.689	3.447	3.459	12.149
HIWFO-SF	NFE	3000	1500	**3000**	**900**	1500	**1500**	3000	3000
	$t, (sec)$	6.979	3.859	**7.193**	**2.027**	3.095	**3.162**	3.079	12.842

Based on Table 5.5, the reading is captured after the Pareto optimal solutions have converged to the Pareto front. MIWO-eSSF and HIWFO-SF achieved lower NFE and time value as compared with other algorithms. As noted in Table 5.6 and Figures 5.9 − 5.13, HIWFO-SF achieved smooth Pareto front with fewer NFE and time in problems MO3 – MO7. In addition, MIWO-eSSF achieved competitively lower NFE and time with smooth Pareto front in problem MO1 (Figure 5.7), MO3 (Figure 5.7), MO5 (Figure 5.11), and MO7 (Figure 5.13).

With respect to SP metrics shown in Tables 5.4, 5.5 and 5.6, HIWFO-SF achieved competitive results compared with other algorithms. HIWFO-SF achieved competitive results in 4 of the problems (MO1, MO2, MO5 and MO6). FA-eSF, IWO-eSF, MIWFO-SF and HIWFO also achieved 4 competitive SP results as highlighted in the respective tables. The results indicate that both algorithms achieved competitive diversification with accuracy in obtaining solutions near Pareto front.

Based on the MS metric results of the algorithms, FA-eSF achieved significant scores on 5 problems as highlighted in Table 5.4. HIWFO, HIWFO-SF and both proposed IWO variants also scored competitively on 4 of the problems as highlighted in their respective tables. The metric of HV designates that the solution obtained measure the spread-out along the Pareto front and nearer to the Pareto front. Based on the simulation results of HV metric, FA-eSF, IWO-eSSF and MIWO-eSSF achieved competitive results in 3 problems. FA-eSF and MIWO-eSSF on the other hand, achieved better performances on convergence and coverage of the Pareto sets than the other algorithms.

Table 5.4: Statistical results of the MO-FA variants

			FA			FA-NSF			FA-eSF	
		SP	MS	HV	SP	MS	HV	SP	MS	HV
MO1	Ave	0.01	0.13	48.54	0.01	0.13	48.58	**4E-02**	**0.13**	**48.71**
	Std Dev	2.24E-03	1.11E-03	2.86E-01	2.07E-03	4.66E-04	2.12E-01	1.16E-03	0.00E+00	5.01E-01
MO2	Ave	0.03	**1.00**	52.10	0.03	**1.00**	52.06	0.03	**1.00**	52.04
	Std Dev	1.74E-03	0.00E+00	2.50E-01	2.45E-04	2.24E-05	1.32E-02	1.65E-04	0.00E+00	2.58E-02
MO3	Ave	**0.10**	0.78	**30.24**	0.12	0.81	21.48	0.21	0.83	24.23
	Std Dev	7.12E-02	1.14E-01	7.55E+00	1.22E-01	1.07E-01	4.63E+00	2.37E-01	1.26E-01	8.30E+00
MO4	Ave	0.04	0.82	48.18	0.04	**0.84**	**49.96**	0.04	**0.84**	**49.96**
	Std Dev	3.01E-03	1.93E-02	3.01E+00	1.04E-04	4.56E-16	2.19E-14	1.84E-04	3.08E-05	2.19E-14
MO5	Ave	0.06	0.59	79.79	0.05	0.59	80.58	**0.05**	0.59	80.88
	Std Dev	1.63E-02	1.02E-03	3.37E+00	8.08E-03	3.17E-04	1.05E+00	1.57E-04	2.28E-16	7.90E-04
MO6	Ave	0.02	0.05	26.02	**0.02**	0.05	25.46	**0.02**	0.05	25.46
	Std Dev	1.33E-03	1.98E-03	5.17E-01	4.11E-03	2.86E-03	1.54E-01	3.09E-03	1.90E-03	1.10E-01
MO7	Ave	0.12	0.03	125.77	0.15	0.02	**129.90**	0.21	**0.03**	**129.06**
	Std Dev	4.17E-02	3.09E-03	6.44E+00	3.44E-02	1.84E-03	4.52E+00	3.48E-02	1.75E-03	4.28E+00
MO8	Ave	7E-03	0.68	18.27	7E-03	0.68	18.05	**6E-03**	**0.68**	18.01
	Std Dev	1.26E-03	1.02E-02	2.65E-01	1.22E-03	1.05E-02	2.72E-01	1.56E-03	1.15E-02	1.77E-01

Table 5.5: Statistical results of the MO-IWO variants

			IWO			IWO-eSSF			MIWO-eSSF	
		SP	MS	HV	SP	MS	HV	SP	MS	HV
MO1	Ave	0.01	0.13	48.58	0.01	0.13	**48.64**	**3E-03**	**0.13**	**48.67**
	Std Dev	2.87E-03	1.43E-03	1.28E+00	3.16E-03	2.08E-03	5.79E-01	4.81E-04	9.73E-05	2.74E-02
MO2	Ave	0.06	0.93	**59.88**	0.03	**1.00**	52.08	0.03	**1.00**	52.08
	Std Dev	1.14E-02	3.21E-02	4.12E+00	1.25E-04	0.00E+00	3.78E-03	1.20E-04	0.00E+00	1.26E-02
MO3	Ave	0.85	0.86	27.72	0.32	**0.88**	25.92	0.37	0.86	**28.37**
	Std Dev	1.98E+00	1.42E-01	1.06E+01	2.74E-01	1.31E-01	6.53E+00	3.61E-01	1.41E-01	8.92E+00
MO4	Ave	**0.03**	0.70	38.20	0.04	**0.84**	**49.96**	0.04	**0.84**	**49.96**
	Std Dev	3.48E-03	4.22E-02	3.85E+00	0.00E+00	4.56E-16	2.19E-14	4.89E-05	4.56E-16	2.19E-14
MO5	Ave	0.06	0.36	**84.69**	**0.05**	0.59	80.88	**0.05**	0.59	80.89
	Std Dev	1.51E-02	2.33E-02	2.90E+00	3.08E-05	6.71E-04	2.24E-05	1.31E-04	2.28E-16	2.45E-02
MO6	Ave	0.03	**0.07**	**37.20**	**0.02**	0.05	25.45	0.03	0.01	26.34
	Std Dev	1.21E-02	1.67E-02	2.28E+00	2.40E-04	2.05E-04	6.93E-03	7.61E-05	2.24E-05	6.17E-02
MO7	Ave	0.13	0.02	120.74	0.23	**0.03**	**139.36**	0.17	**0.03**	123.52
	Std Dev	3.60E-02	2.62E-03	8.69E+00	2.04E-02	5.03E-04	3.08E+00	3.36E-02	1.59E-03	7.91E+00
MO8	Ave	0.02	0.65	**24.37**	0.01	0.66	18.43	0.01	0.66	18.39
	Std Dev	2.31E-03	2.19E-02	1.27E+00	3.10E-03	1.86E-02	1.16E-01	3.12E-03	1.89E-02	1.42E-01

Figures 5.7 – 5.14 illustrates of the Pareto fronts based on the best Pareto optimal solutions achieved by the algorithms. As noted in Figures 5.7, 5.8, 5.10, 5.11 and 5.12, most of the algorithms achieved good distribution of solutions and successfully converged to the Pareto front. However, MIWO-eSSF produced extended distribution as noted in Figure 5.11.

Table 5.6: Statistical results of the MO-HIWFO and MO-HIWFO-SF

		HIWFO			HIWFO-SF		
		SP	MS	HV	SP	MS	HV
MO1	Ave	0.01	**0.13**	48.48	**3E-03**	**0.13**	**48.67**
	Std Dev	3.50E-03	1.67E-03	8.66E-01	2.24E-05	0.00E+00	2.53E-03
MO2	Ave	0.03	**1.00**	52.10	**0.03**	**1.00**	52.24
	Std Dev	1.49E-03	0.00E+00	2.05E-01	1.45E-03	0.00E+00	6.75E-01
MO3	Ave	0.77	**0.91**	25.77	0.26	**0.88**	24.21
	Std Dev	1.86E+00	1.29E-01	9.76E+00	4.16E-01	1.28E-01	5.99E+00
MO4	Ave	0.04	**0.84**	**49.95**	0.04	**0.84**	**49.96**
	Std Dev	1.57E-03	4.59E-04	5.01E-02	1.82E-04	4.10E-05	2.19E-14
MO5	Ave	**0.05**	0.59	77.28	**0.05**	0.59	80.88
	Std Dev	3.69E-04	2.28E-16	1.61E+01	1.40E-04	2.28E-16	4.21E-04
MO6	Ave	**0.02**	0.05	25.50	**0.02**	0.05	25.45
	Std Dev	1.57E-03	1.88E-03	2.44E-01	1.08E-03	5.46E-04	1.23E-02
MO7	Ave	**0.11**	0.03	108.11	0.14	0.03	112.24
	Std Dev	2.97E-02	2.33E-03	6.12E+00	3.90E-02	2.27E-03	6.35E+00
MO8	Ave	0.01	0.66	18.57	0.01	0.66	18.61
	Std Dev	2.70E-03	1.94E-02	1.84E-01	4.78E-03	1.78E-02	3.19E-01

MIWO-SF also had good coverage and convergence, as noted in Figures 5.9, 5.12 and 5.13 by maximising the range covered by its Pareto optimum solutions. In all the distributions, IWO clearly exhibited significant scatter points, as seen in Figures 5.10, 5.11, 5.12 and 5.14. However, as noted in Figures 5.7 and 5.13, the IWO algorithm achieved good distribution and solutions near to Pareto front. HIWFO and HIWFO-SF also produced good distribution, although a few solutions were scattered far from Pareto front, as seen in Figure 5.9 and 5.12. The solutions with FA-eSF and FA-NSF, on the other hand, were hard to trace into a proper Pareto front.

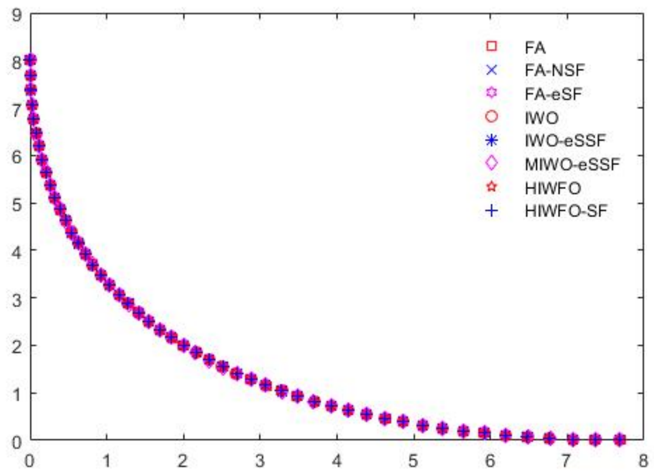


Figure 5.7: Solution set of algorithms for SCH 1

From the outcome results of the Pareto front in Figure 5.14 as well as the SP, MS and HV metrics after 30,000 NFE as shown in Tables 5.4, 5.5 and 5.6, it can be summarized that the modified multi-objective algorithm with FA-eSF showed superior performance compared to the other algorithms used. HIWFO-SF and MIWO-eSF also showed competitive results

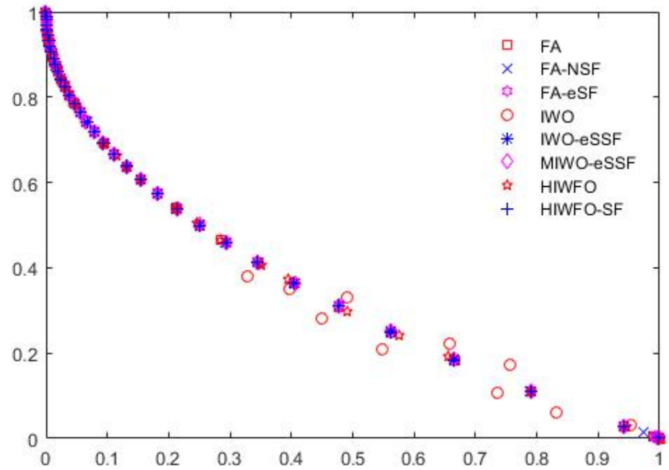


Figure 5.8: Solution set of algorithms for ZDT 1

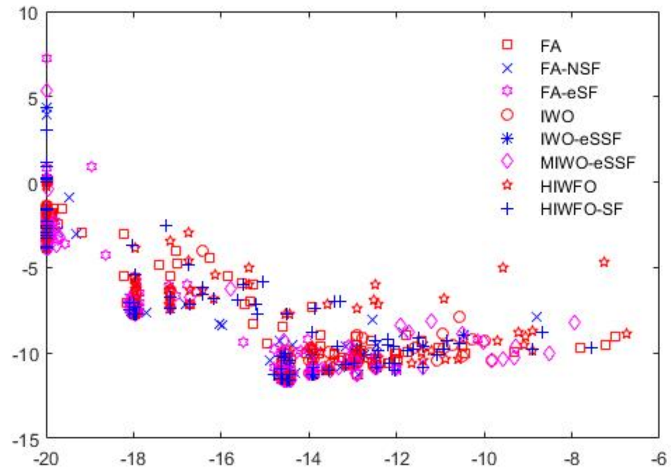


Figure 5.9: Solution set of algorithms for Kursawe

in converging to solutions to the Pareto front. The adaptive SF mechanism helped these algorithms to improve the solution to converge to the non-dominated solution set during the optimisation process. It also helped the algorithms to diversify the solution by maximizing the range covered by the solution and improve the distribution along the Pareto front.

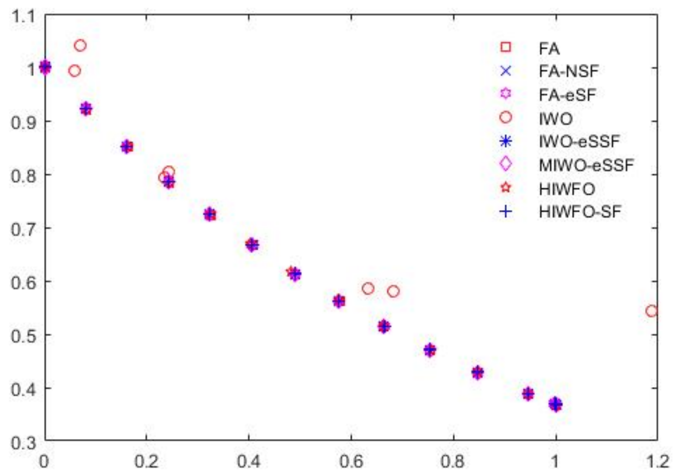


Figure 5.10: Solution set of algorithms for CTP

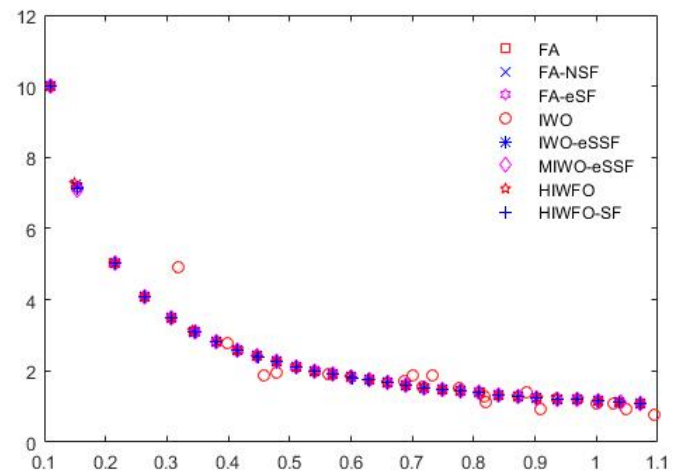


Figure 5.11: Solution set of algorithms for Constr

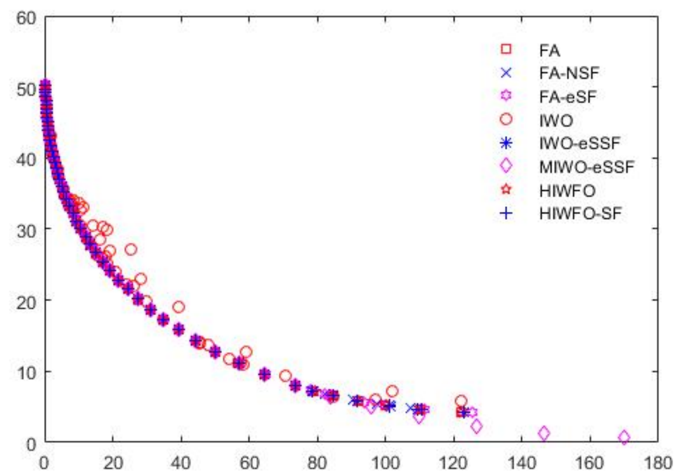


Figure 5.12: Solution set of algorithms for Bihn and Korn

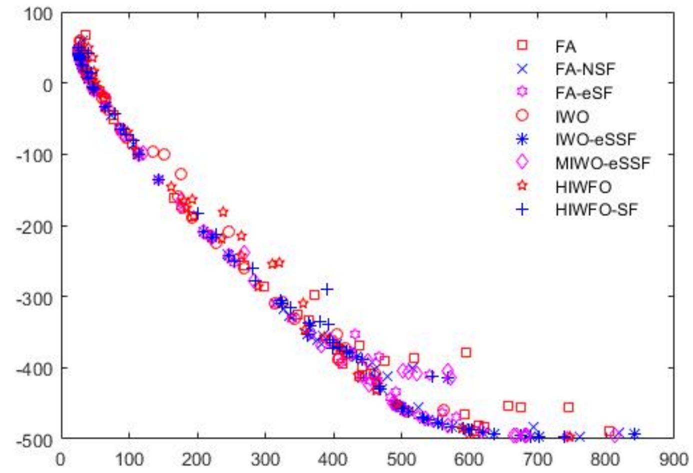


Figure 5.13: Solution set of algorithms for Chankong and Haimes

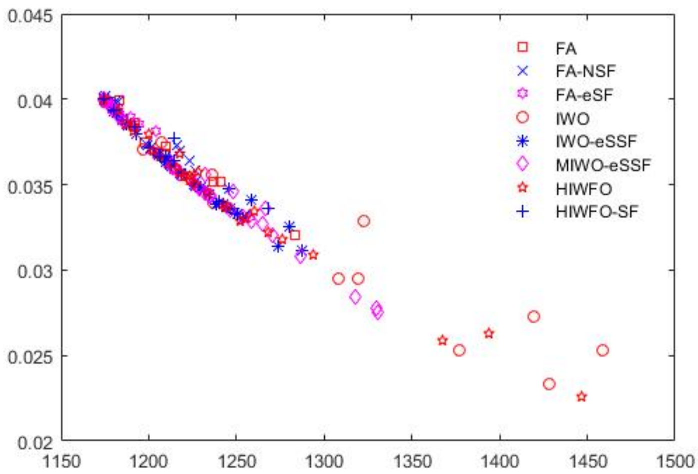


Figure 5.14: Solution set of algorithms for four bar place truss

5.7 Summary

The algorithms were successfully extended to solve multi-objective problems. The focus of this research was to study and compare the proposed algorithms with their predecessors in terms of effectiveness in handling multi-objective problems. The algorithms were further modified with constraint-handling techniques to deal with constrained problems MO4 – MO8. Based on the results obtained,

- MIWO-eSSF and HIWFO-SF showed lower value of NFE and time in solving multi-objective benchmark problems. They also produced good Pareto distribution. However, based on the performance metrics used, FA-eSF outperformed other algorithms in converging to the optimal solution on the Pareto front.
- Thus, FA-eSF is potentially a good optimiser for solving multi-objective problems.

The proposed FA and IWO variants enhanced with adaptive SF mechanism have shown potential in solving multi-objective problems. The above results suggest that the proposed multi-objective algorithms MIWO-eSSF and HIWFO-SF showed superior performance over other algorithms. Based on the evaluations carried out, these algorithms could find better Pareto-optimal solutions as well as managed to improve the coverage and convergence of the Pareto sets to the Pareto front. This observation was based on comparison with other algorithms used in the experiments.

These proposed algorithms have shown potential in solving single and multi-objective optimisation problems. The handling of constraints and various conditions have been tested and evaluated. The next chapter will discuss the application of the proposed algorithms to solving four engineering problems. The chapter focuses on system modelling of twin rotor system (TRS) and controller design of flexible manipulator system (FMS), human arm model and also lower extremities mechanism.

Chapter 6

Application to engineering problems

6.1 Introduction

In this chapter, the proposed bio-inspired algorithms are applied to four engineering applications, namely system modelling of a twin rotor system (TRS), tracking control of a flexible manipulator system (FMS) and controller design for two exoskeleton applications. The characteristics of the systems used are briefly described. The modelling of TRS and control design for position tracking control of FMS as well as for exoskeleton have received considerable attention from many researcher. However, the use of FA and IWO-based algorithms have not been reported in modelling of TRS and optimisation of control mechanism for application to FMS and exoskeleton.

For system modelling, linear parametric modelling of the TRS is implemented using the proposed bio-inspired algorithms. This chapter also presents the application of the algorithms for optimisation of parameters of proportional-derivation fuzzy logic controller (PD-FLC) for hub-angle position tracking controller of the FMS and proportional-derivation-integral (PID) controller for set-point tracking control of human movement for both upper and lower extremity exoskeleton applications. Comparative assessments of the results among the algorithms are presented.

The experimental testing hardware platform comprises a personal computer (PC) with processor CPU Intel (R) Core (TM) i5-2400 with operating systems Window 7 Professional, frequency of 3.10 GHz and memory installed of 4.00 GB RAM. The program is coded in MATLAB R2013a. Each problem is tested with the same basic initial parameters for a fair comparative evaluation as shown in Table 4.4.

6.2 Application to Modelling of Twin Rotor System

In this section, the proposed algorithms are used in modelling a TRS. Developed by Feedback Instrument Ltd (Feedback Ltd, 1996), the TRS is a laboratory-scaled platform of a flexible manoeuvring structure that resembles essential characteristics of an air vehicle. It is a simplified version of practical helicopter and has attracted many researchers as 'test rig' for aerodynamic experiments (Alam and Tokhi, 2007; Toha and Tokhi, 2010).

In order for TRS to be controlled and manoeuvred, development of system modelling of TRS is still a challenge especially in obtaining the highly nonlinear dynamic model of the system. Aldebrez et al. (2004) have investigated and analysed the potential of modelling approach of the TRS using non-parametric neural networks (NN) and parametric linear modelling using conventional recursive least squares (RLS) technique. Aldebrez et al. (2004) have also investigated parametric dynamic modelling of TRS using genetic algorithm (GA) in comparison with conventional RLS. Alam and Tokhi (2007) have investigated TRS using particle swarm optimisation (PSO) for both 1 and 2 degrees of freedom (DOF) of the system. PSO has also been used in adaptive neuro-fuzzy interface system (ANFIS) modelling of TRS (Toha and Tokhi, 2009). Moreover, the PSO, RLS and GA have been assessed on a comparative basis in parametric modelling of the TRS (Toha and Tokhi, 2010). Toha et al., (2012) proposed the use of ACO technique for modelling the TRS. Nasir and Tokhi (2014, 2015) combined bacteria foraging algorithm (BFA) and spiral dynamic algorithm and applied the algorithm to dynamic modelling of TRS. Furthermore, adaptive spiral algorithm has been employed by Nasir et al. (2016) to modelling of TRS.

6.2.1 System Modelling

The schematic diagram of the TRS is as shown in Figure 6.1 (Toha et al., 2012). The experimental rig of the system have been described in the literature (Ahmad et al., 2001; Nasir and Tokhi, 2014; Toha and Tokhi, 2010). The investigation here focuses on dynamic modelling of the system in vertical movement.

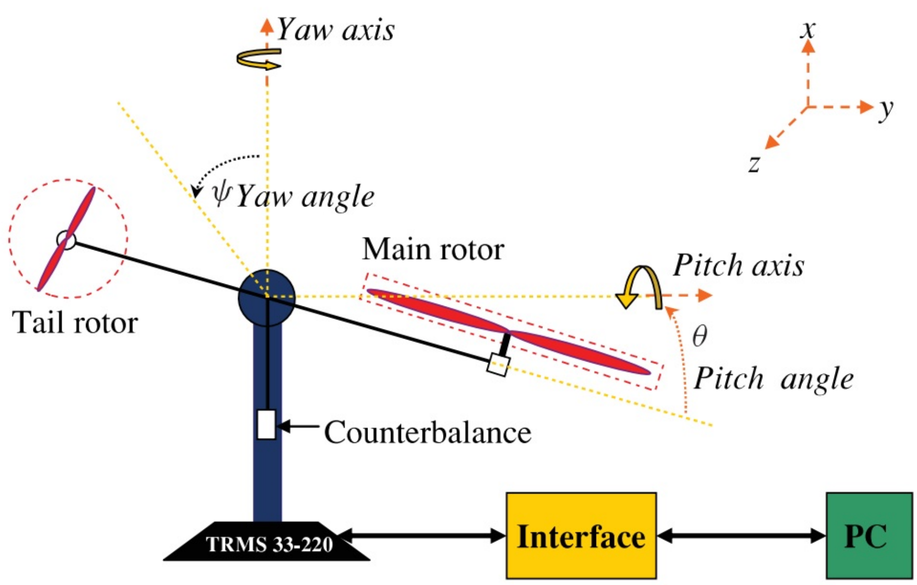


Figure 6.1: Schematic diagram of TRS

The TRS consists of two motors; the main rotor and tail rotor motors placed at both ends of a beam pivoted on a base of the system allowing it to rotate freely in vertical and horizontal planes. The rotational speed of each rotor can be controlled by changing the input voltage to the motor. The articulated joint will allow the beam to move and rotate accordingly. The system is equipped with a countermeasure device as seen in Figure 6.1, which is used for balancing the angular momentum. It based on a pendulum counterweight hanging from the beam. In TRS, the vertical movement of the system is driven by the main rotor, whereas the tail rotor drives the horizontal movement.

In this experiment, the identification of a dynamic model in linear parametric form is implemented by using the proposed bio-inspired algorithms. Linear system identification is implemented to acquire model of the TRS based on input-output data collected from the actual system. Linear parametric modelling is one of the techniques in system identification to estimate a linear model of a system (Ljung, 1987). In this experiment, a set of unknown parameters in a predefined structure are identified using the proposed optimisation algorithms.

In the preliminary task, a random signal with a sampling time, Ts of 0.1 s or sampling frequency, fs of 10 Hz was used as input in the vertical channel of the system. A total of 3000 actual input–output data was recorded to estimate the vertical channel model of the TRS. The first 2000 or two-thirds of the data were used in the modelling phase and the remaining 1000 data was used in the validation phase of the estimated model.

The ARX model structure is used as it is a simple structure that offers good performance with relatively low computational cost in flexible systems (Nasir and Tokhi, 2014; Toha et al., 2012). The general mathematical expression of the selected model structure can be written as (Ljung, 1987)

$$\widehat{y}(t) = -\Sigma_{i=1}^{N} a_i y(t-i) + \Sigma_{i=1}^{M} b_j u(t-j) + \eta(t) \tag{6.1}$$

where $\widehat{y}(t)$ is the predicted output, $y(t)$ represents the measured system output, $u(t)$ is the measured system input, $\eta(t)$ is the system noise and a_i and b_j are the output and input coefficients, N and M are the number of coefficients for the output and input samples. Assuming that the actual model of the system is very good, then the measured output is highly dependent on the excited input and previous measured output and thus the noise term in the ARX expression can be neglected (Toha et al., 2012). From equation (6.1) the simplified equation predicting output can be written in discrete form as

$$\widehat{y}(k) = -a_1 y(k-1) - \cdots - a_i y(k-N) + b_1 u(k-1) + \cdots + b_j u(k-M) \tag{6.2}$$

Hence, based on equation (6.2), the transfer function used in the experiments is viewed using the backshift operator, z^{-1} as

$$H(z) = \frac{Y(z)}{U(z)} = \frac{b_0 z^M + b_1 z^{M-1} + \cdots + b_{M-1} z^1 + b_M}{z^N + a_1 z^{N-1} + \cdots + a_{N-1} z^1 + a_N} \tag{6.3}$$

From the discrete transfer function in equation (6.3), the poles can be determined from the denominator. The stability and accuracy of the predicted model of the system can be evaluated by the unknown coefficient of the denominator and numerator. A fourth order model is employed in this experiment as it also gives better representation of the system dynamics than a second or sixth order model (Ahmad et al., 2001; Toha et al., 2012). The error, e can be determine between the actual output, $y(t)$ and the predicted output, $\widehat{y}(t)$ as follows:

$$e(t) = y(t) - \widehat{y}(t) \tag{6.4}$$

The error and accuracy have an inverse relationship where if the error value is reduced then the accuracy of the predicted model is improved. The optimisation algorithms are used to determine the lowest error reading and the resultant parameters will be the optimised model parameters. In this process, mean-squared error (MSE) is used as the objective function.

$$MSE = \sqrt{\frac{1}{N}\Sigma_{i=1}^{N} e^2} \tag{6.5}$$

where e^2 represents the square of error captured between the actual output and predicted output of the TRS. Figure 6.2 shows the block diagram of the parameters estimation exercise to determine the parametric model of the system. Optimisation algorithms are used in this

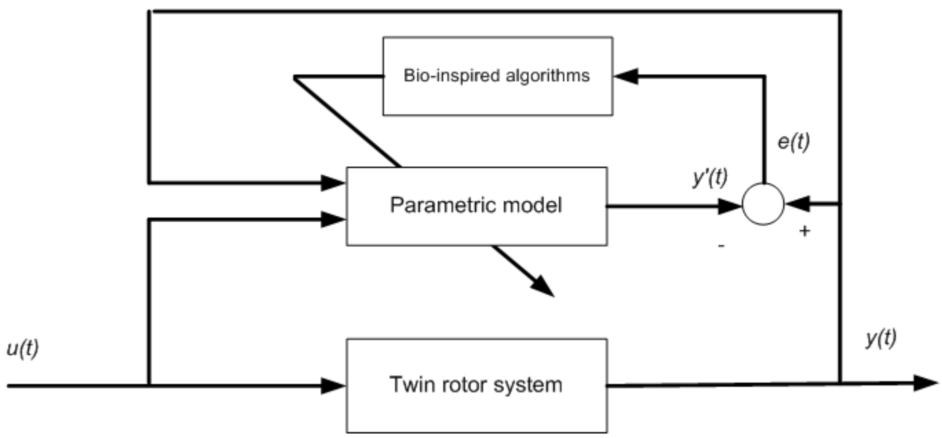


Figure 6.2: Block diagram for parametric modelling with bio-inspired algorithms

research to search for parameter values of the system model such that the objective function represented by MSE value converges to zero. Hence, the predicted model is formed based on a fourth-order discrete transfer function, $H(z)$ of the form in equation (6.3). After that, the obtained predicted model is tested so that it can adequately describe the data set in any model or algorithm identification (Billings and Zhu, 1994). For this reason, the validation process is carried out as shown in Figure 6.3. In the validation of the model, correlation tests are carried out. The auto-correlation test of the residuals and cross-correlation test between the residuals and the input are performed in the validation test.

The validation process is done to ensure the identified model gives information on the

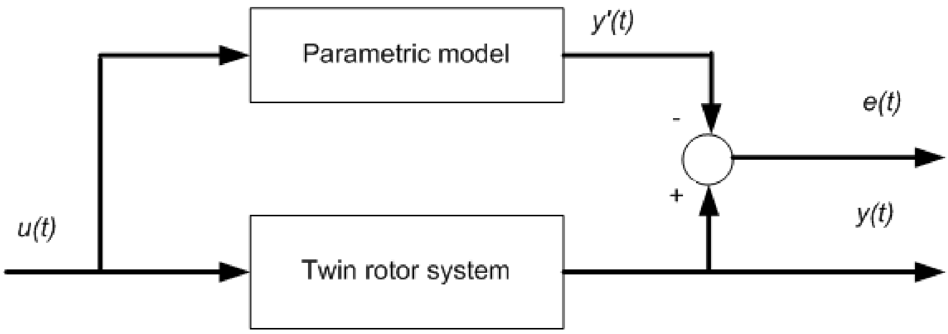


Figure 6.3: Block diagram of validation process

quality of the model structure being investigated and also could indicate bias to noise. If the model is found to be adequate, the residuals or prediction errors $e(t)$ should be unpredictable from all linear and nonlinear combinations of past inputs and outputs. Based on convergence and validation tests, the performance of the algorithms used in this research are compared in modelling of the TRS.

6.2.2 Experiments

This section describes dynamic modelling of vertical channel of the TRS. The basic criteria used are as follows:

- Maximum number of population, $n_{max} = 30$.
- Stopping criterion based on when the cost function, $f(x)_{min} < 10^{-4}$.

The cost function, $f(x)$ values which represent the MSE values, the number of iterations and computational time are as shown in Table 6.1. The min and max values represent the minimum and maximum error values from the actual signal and predicted output signal produced in the validation phase.

Table 6.1: The numerical results for the modelling of TRS

Algorithm	MSE, f(x)	Iteration	NFE	$t(sec)$	Min	Max
FA	8.85E-05	20000	600000	3196.426	-0.0323	0.0166
FA-NSF	9.42E-05	5000	150000	359.8766	-0.0319	0.0143
FA-eSF	8.99E-05	5000	150000	371.4814	-0.0312	0.0145
IWO	4.00E-05	5000	150000	355.6572	-0.0225	0.011
IWO-eSSF	2.37E-05	1000	**30000**	**39.30204**	-0.0157	0.0165
MIWO-eSSF	3.73E-05	2000	60000	129.2702	-0.0167	0.0221
HIWFO	**1.61E-05**	5000	150000	422.7999	**-0.0147**	**0.0099**
HIWFO-SF	2.10E-05	5000	150000	583.0652	-0.0161	0.0100

Based on Table 6.1, IWO-eSSF produced lower MSE and NFE values compared with other algorithms. Furthermore, Figure 6.4 shows the convergence plots of the algorithms for up to 5000 iterations. The graphs show that FA, IWO, HIWFO and HIWFO-SF converged fast in the early stage. However, the FA appeared stuck at a local optimum. However, IWO-eSSF and MIWO-eSSF showed steady convergence and were able to reach the stopping criterion faster than other algorithms. They also needed lower number of iterations and NFE as well as time to converge. The HIWFO and HIWFO-SF algorithms also managed to converge and produce better fitness values than MIWO-eSSF. However, their computational time was higher than the other proposed algorithms. FA-NSF and FA-eSF were able to outperform their predecessor algorithm, however, they took more iterations and NFE to reach the stopping criterion.

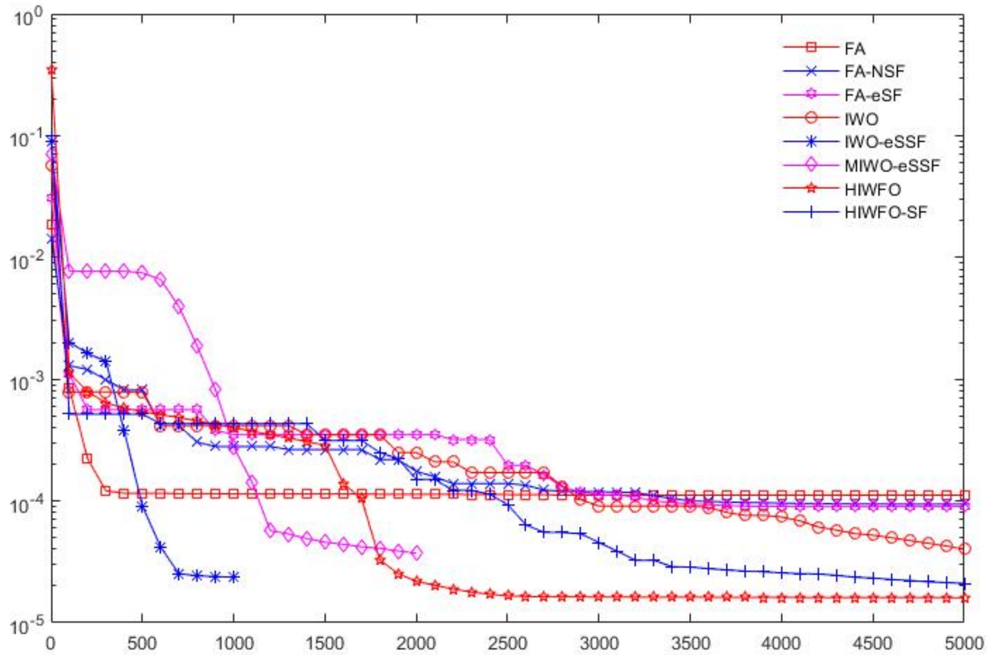


Figure 6.4: Convergence graph for twin rotor system

6.2.3 Validation Tests

Based on the parameters obtained from the optimisation process, the forth-order discrete transfer function for the predicted vertical channel model can be derived. The derived ARX models thus obtained with the algorithms used can be presented as

$$H(z)_{FA} = \frac{-0.01645z^4 + 0.00602z^3 - 0.00371z^2 + 0.00449}{z^4 - 1.00000z^3 - 0.59903z^2 + 0.44862z^1 + 0.24220} \tag{6.6}$$

$$H(z)_{FA-NSF} = \frac{-0.01251z^4 - 0.00156z^3 - 0.00413z^2 + 0.004690}{z^4 - 1.00000z^3 - 0.501049z^2 + 0.26926z^1 + 0.32466} \tag{6.7}$$

$$H(z)_{FA-eSF} = \frac{-0.01325z^4 + 0.00176z^3 - 0.00455z^2 + 0.04784}{z^4 - 0.99999z^3 - 0.55613z^2 + 0.37766z^1 + 0.26924} \tag{6.8}$$

$$H(z)_{IWO} = \frac{-0.00874z^4 + 0.00078z^3 + 0.00095z^2 + 0.02699}{z^4 - 1.70101z^3 + 0.65977z^2 - 0.03937z^1 + 0.13707} \tag{6.9}$$

$$H(z)_{IWO-eSSF} = \frac{0.00475z^4 - 0.00072z^3 + 0.00099z^2 - 0.00545}{z^4 - 2.58219z^3 + 1.68939z^2 + 0.42686z^1 - 0.53998} \tag{6.10}$$

$$H(z)_{MIWO-eSSF} = \frac{0.00327z^4 + 0.00817z^3 - 0.002381z^2 - 0.01696}{z^4 - 2.69469z^3 + 1.53923z^2 + 1.02261z^1 - 0.89435} \tag{6.11}$$

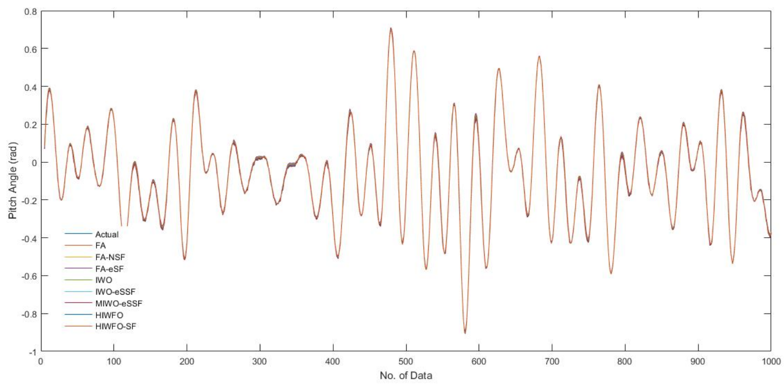
$$H(z)_{HIWFO} = \frac{0.00157z^4 - 0.83555z^3 + 0.00241z^2 + 0.00196}{z^4 - 2.15974z^3 + 0.83555z^2 + 0.90498z^1 - 0.57123} \tag{6.12}$$

$$H(z)_{HIWFO-SF} = \frac{-0.00189z^4 - 0.00212z^3 + 0.00287z^2 + 0.01374}{z^4 - 1.88913z^3 + 0.65861z^2 + 0.49052z^1 - 0.22617} \tag{6.13}$$

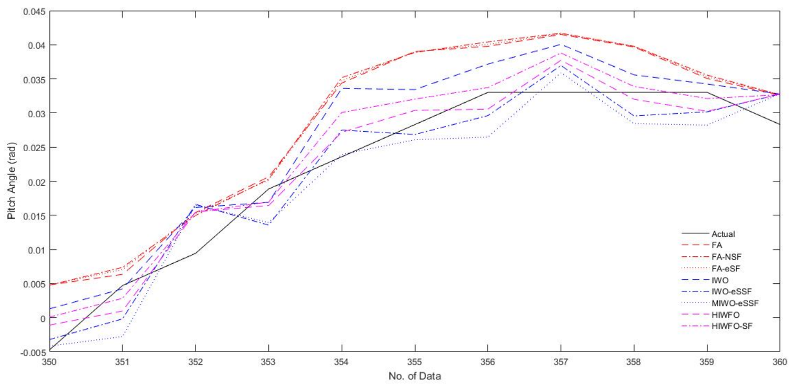
The graphical results for the validation tests are shown in Figures 6.5, 6.6 and 6.7. Figure 6.5 shows the plots of actual and predicted outputs based on the models obtained. The corresponding output-error plots are also shown in Figure 6.6. Figure 6.5b show the zoom-in plot of Figure 6.5a.The ranges of errors for the algorithms are listed in Table 6.1. Based on the observation in Figure 6.5, it can be stated that all the derived models based on all algorithms managed to replicate and predict the pitch movement very well. In Figure 6.5b, it is clearly shown that IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF were able to predict the movement more accurately than the other algorithms. HIWFO and HIWFO-SF also produced low error range as seen in Table 6.1. The average error accuracy between the actual and predicted output was around 99.9% for all the algorithms.

The power spectral density plots are shown in Figure 6.7. In the actual system output, the main resonance mode is found at 0.34 Hz which can be attributed to the main body dynamics. It is noted in Figure 6.7b, that all the algorithms managed to capture the dynamic characteristics of the vertical channel. It is observed that the derived model successfully replicated the actual system dynamics in the low frequency region and had some reading differences at higher frequency, which are not significant in the operation of the TRS.

The stability of the derived system model produced is analysed by pole-zero diagram as illustrated in Figure 6.8. It can clearly be observed that all the poles locations for all the models are within the unit circle in the z-plane. It is further observed that there are some



(a) The output response



(b) The response (zoomed-in)

Figure 6.5: The actual and predicted outputs

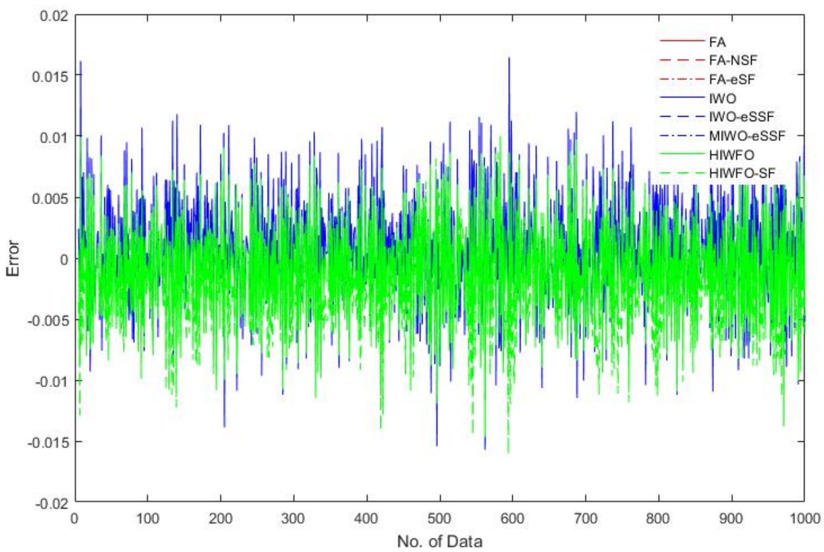
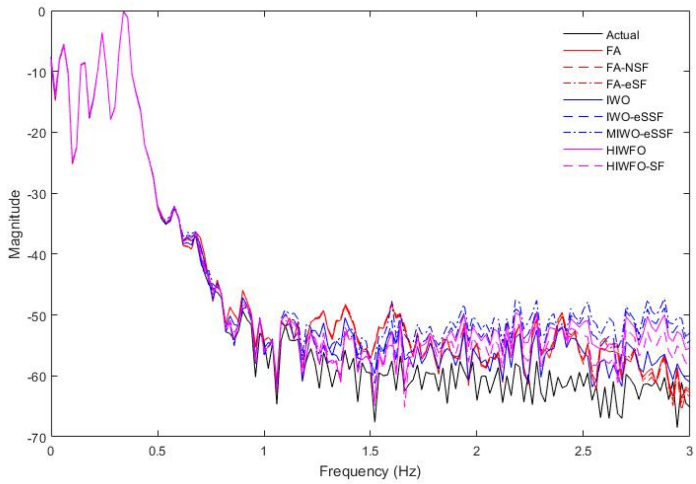


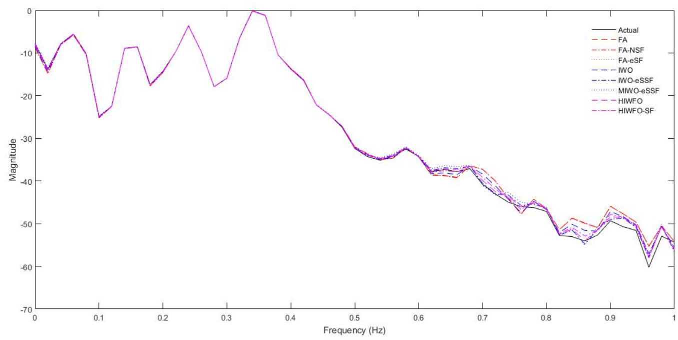
Figure 6.6: The error measured between actual and predicted outputs

zeros located outside the unit circle. Therefore, it could indicate that the models obtained were stable with non-minimum phase behaviour.

The results of the correlation validation tests for the models are shown in Figures 6.9, 6.10 and 6.11 where the red tick lines represent the 95% confidence boundary intervals with 1000 data pairs used in the validation phase. It is noted that for models obtained with IWO, IWO-eSSF, MIWO-eSSF and both hybrid algorithms, HIWFO and HIWFO-SF show that their correlation functions were within the 95% interval. This implies that the model outputs were unbiased and the predicted model outputs were acceptable. On the other hand, FA variants algorithms showed some difficulty in achieving correlation functions within 95% confidence interval.

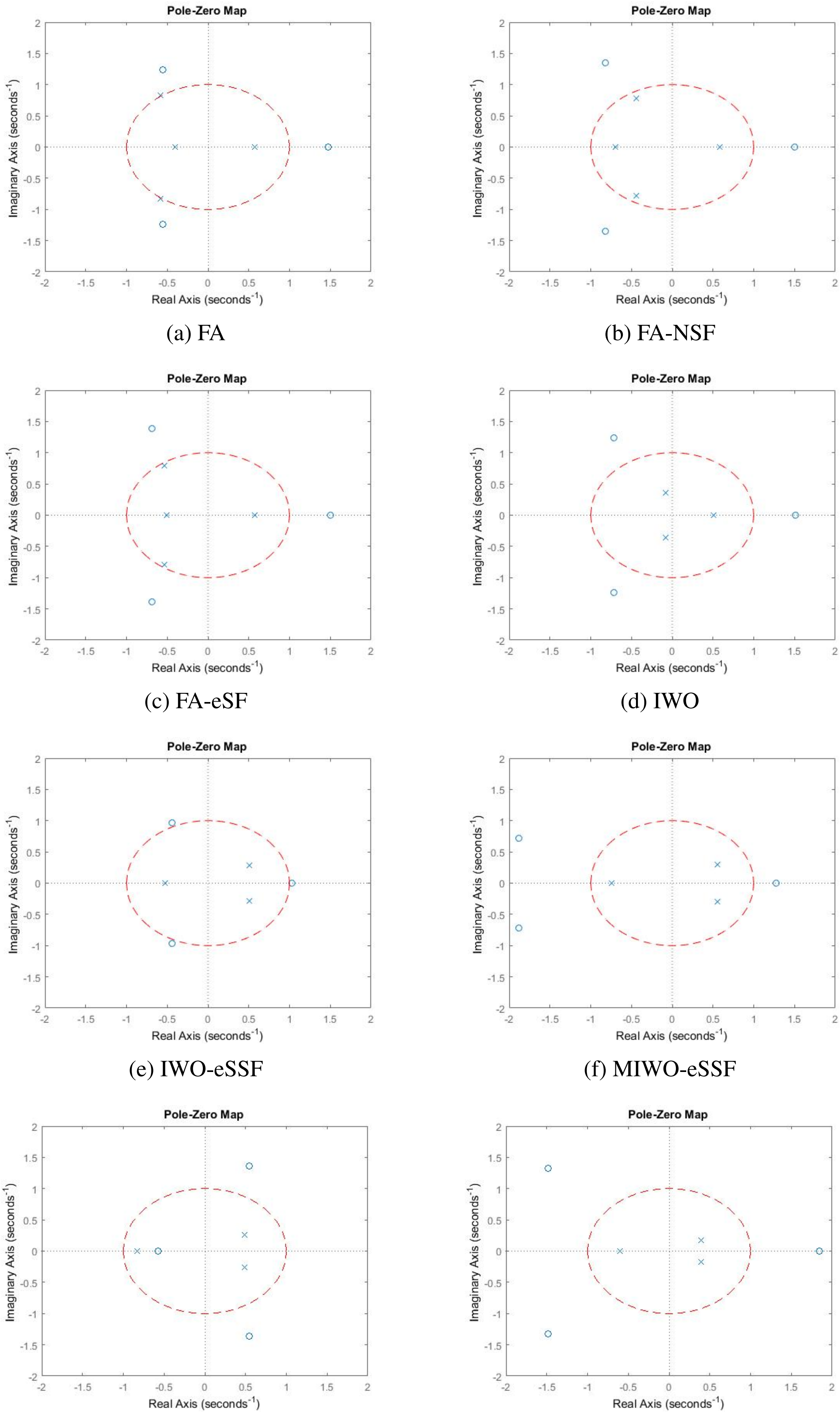


(a) The output response



(b) The response (zoomed-in)

Figure 6.7: Power spectrum densities of actual and predicted outputs



(a) FA

(b) FA-NSF

(c) FA-eSF

(d) IWO

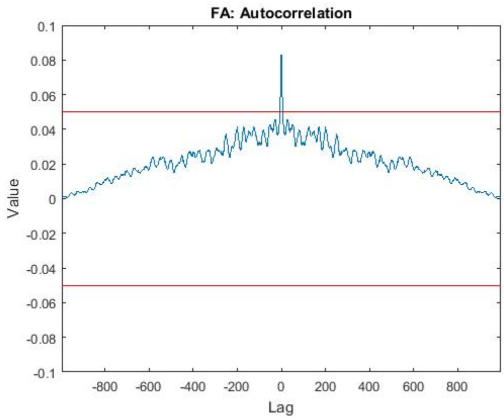
(e) IWO-eSSF

(f) MIWO-eSSF

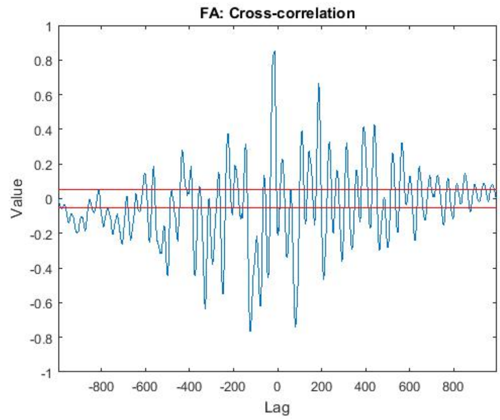
(g) HIWFO

(h) HIWFO-SF

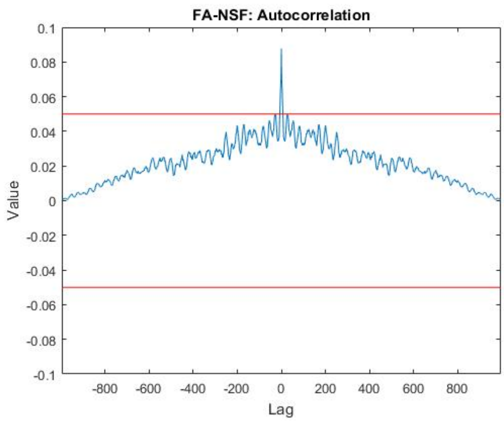
Figure 6.8: Pole-zero diagrams of the obtained models



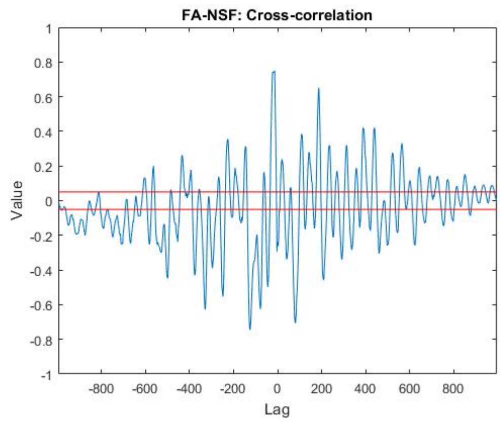
(a) FA: auto-correlation



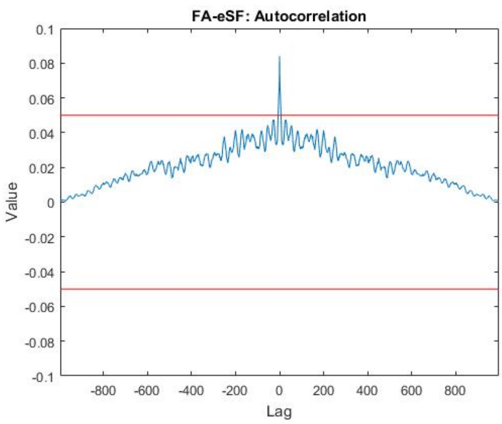
(b) FA: cross-correlation



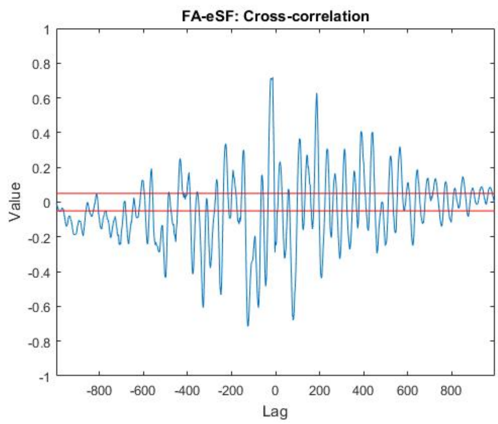
(c) FA-NSF: auto-correlation



(d) FA-NSF: cross-correlation

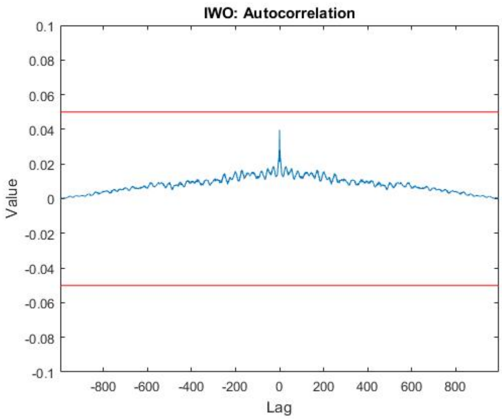


(e) FA-eSF: auto-correlation

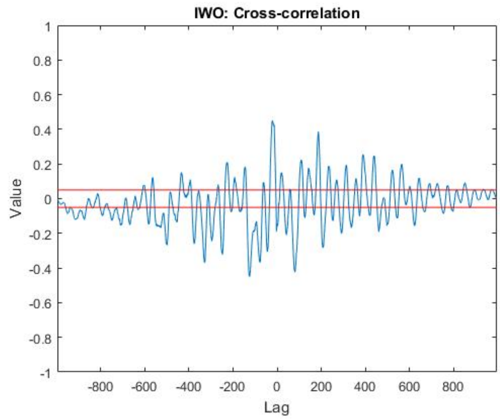


(f) FA-eSF: cross-correlation

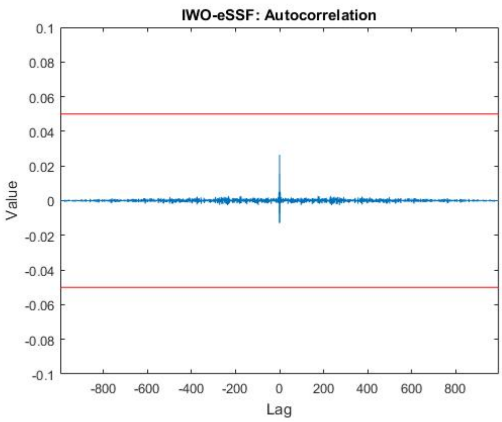
Figure 6.9: Correlation tests of residuals for FA variants



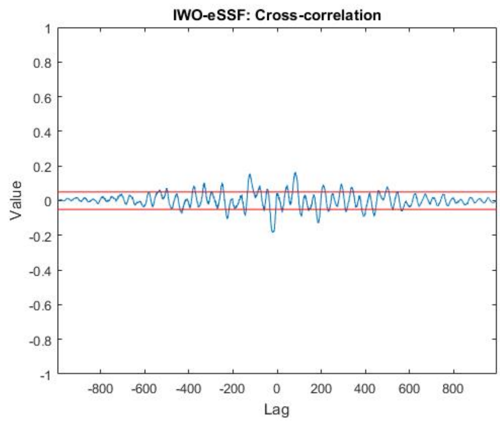
(a) IWO: auto-correlation



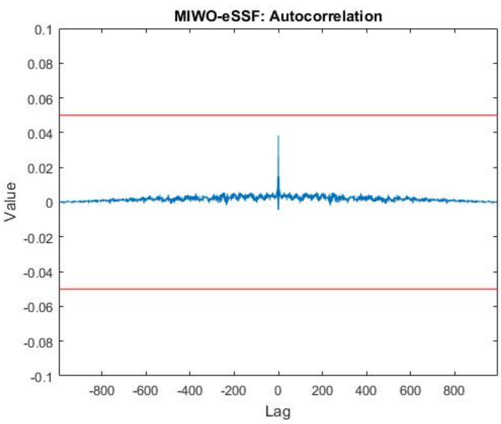
(b) IWO: cross-correlation



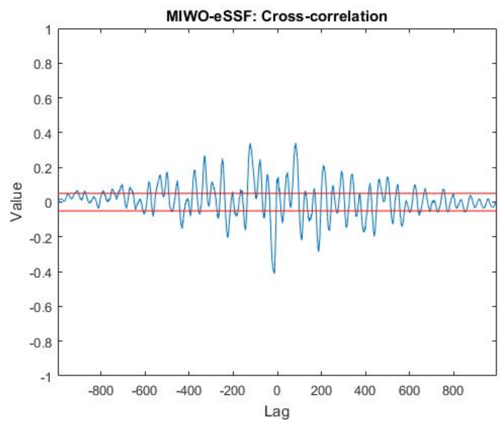
(c) IWO-eSSF: auto-correlation



(d) IWO-eSSF: cross-correlation

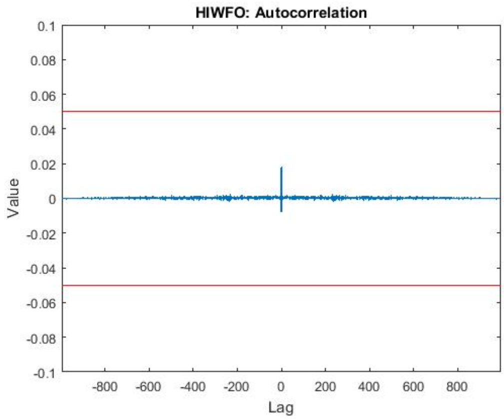


(e) MIWO-eSSF: auto-correlation

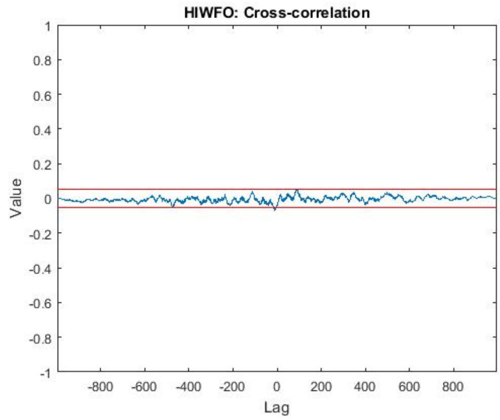


(f) MIWO-eSSF: cross-correlation

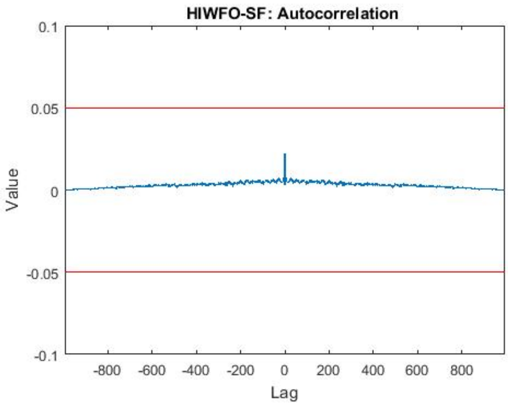
Figure 6.10: Correlation tests of residuals for IWO variants



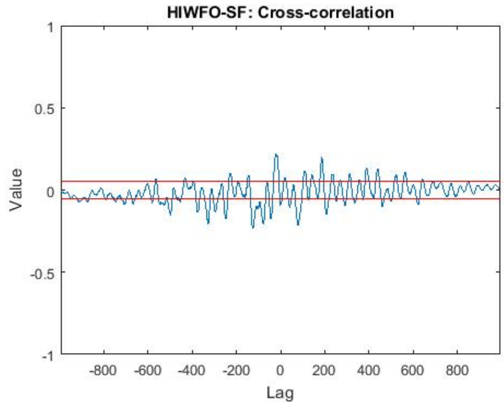
(a) HIWFO: auto-correlation



(b) HIWFO: cross-correlation



(c) HIWFO-SF: auto-correlation



(d) HIWFO-SF: cross-correlation

Figure 6.11: Correlation tests of residuals for HIWFO and HIWFO-SF

6.3 Application to Control Mechanism of a Flexible Manipulator

In this section, the proposed algorithms are used to determine the desired optimal parameter values of controller for single-link FMS. The FMS has been widely researched especially in the area of industrial automation. The manipulators complexity and flexible character have motivated researchers to look into the potential of FMS in various applications within the automation industry. Due to the flexible nature of FMS, accurate system model is needed in order to achieve precision control. Among classical control techniques based on dynamical model, PID control is found most effective with direct calculation of the torque. However, in order to derive the inverse model of FMS, the approach requires a time-consuming procedure. Thus, to avoid such issues artificial intelligence techniques such as neural networks, fuzzy logic and their combination are employed, and these type of controllers for FMS have drawn interest of many researchers.

Optimisation techniques can be used in the control systems that have big impact on achieving the desired characteristics of a control process. Recently, considerable attention has been paid for bio-inspired optimisation techniques. Various control mechanisms with bio-inspired algorithms have been proposed for flexible manipulators. These include using genetic algorithm (Siddique and Tokhi, 2002), particle swarm optimisation (Elkaranshway et al., 2011; Yatim and Darus, 2014), bacteria foraging algorithm (BFA) (Alavandar et al., 2010; Nasir and Tokhi, 2012; Supriyono et al., 2010), and hybrid BFA with spiral dynamics (Nasir and Tokhi, 2015). To the best knowledge of the author, FA and IWO algorithms have not been applied to control of flexible manipulators.

Therefore, this section presents the development of an FLC mechanism optimised by the algorithms for a flexible manipulator. The focus of this section is to investigate the control of a flexible manipulator with an FLC mechanism optimised by the algorithms. The performance of the proposed algorithms as well as comparison with FA and IWO algorithms in tuning parameters of a fuzzy logic controller for a flexible manipulator are evaluated.

6.3.1 Control Mechanism of a Single-link Flexible Manipulator

The main parts of the flexible manipulator considered in this work comprise a drive motor, a flexible arm and measuring devices. A shaft encoder and tachometer placed at the hub are used to measure the hub-angular position and hub-angular velocity. An accelerometer placed at end-point of the manipulator is used for measurement of end-point acceleration. A schematic diagram of the laboratory scale planar-constrained single-link flexible manipulator (Azad, 1994) used in this work is shown in Figure 6.12. The experimental installation's mechanism can be described using Figures 6.12 and 6.13. The POQ and $P'OQ'$ represent the stationary and moving coordinates. τ represents the motor torque applied at the hub resulting the angular displacement θ. l is the length of the beam, u is linear displacement of a point x along the beam, and M_p and I_p represent a payload mass with associated inertia at the end-point of the manipulator. The manipulator is considered with hub inertia I_h and moment of inertia I_b. The maximum angular range of the manipulator is $[-80, 80]$ degrees.

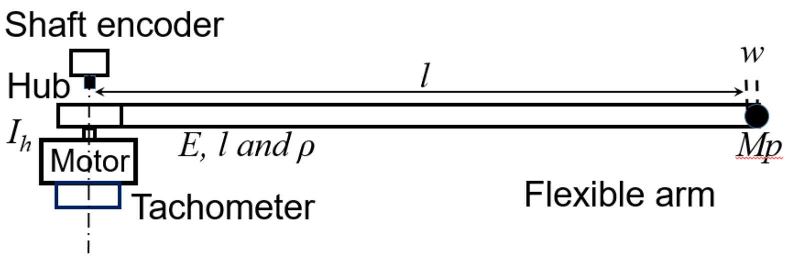


Figure 6.12: Schematic diagram of flexible manipulator system

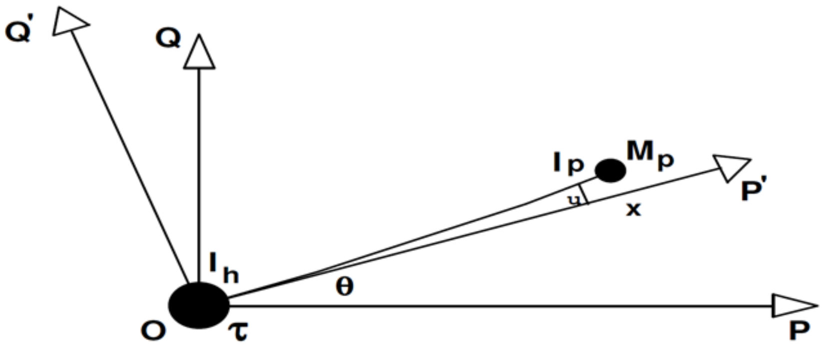


Figure 6.13: Single-link flexible manipulator representation (Azad, 1994)

According to Figure 6.13, the angular position $y(x,t)$ of a point at distance x from the hub is represented as follows:

$$y(x,t) = x\theta(t) + u(x,t) \tag{6.14}$$

where $\theta(t)$ and $u(x,t)$ denote the rigid body motion and elastic deflection respectively. The dynamic equation for the single-link flexible manipulator was developed by Azad (1994) and Poerwanto (1998) and can be formulated as follows:

$$El\frac{\partial^4 u(x,t)}{\partial x^4} + \rho\frac{\partial^2 u(x,t)}{\partial x^2} - D_s\frac{\partial^3 u(x,t)}{\partial x^2 \partial t} = -\rho x\theta \tag{6.15}$$

where $El\frac{\partial^4 u(x,t)}{\partial x^4}$ is a damping moment which is dissipated in flexible manipulator structure. To solve the mathematical model in equation (6.15) the finite difference method is used. A state-space model has been derived and represented in simulink based on parameters of the real experimental setup (Azad, 1994; Poerwanto, 1998). The developed models proposed by Azad (1994) and Poerwanto (1998) are configured into one model and used in simulating the system in this work.

Due to the non-linear character of mathematical model of flexible manipulator system, it will prove difficult to satisfactorily achieve performance requirements such as accuracy, speed, quality of transient with classical control techniques. An effective solution can be obtained by combining the system with fuzzy logic control and an optimisation algorithm. Optimisation of fuzzy controller in this work is carried out using the bio-inspired approaches used in this research. In the controller optimisation process, tuning of parameters of scaling factors and membership functions of each linguistic variable are considered. The flexible manipulator model and the controller design are implemented and simulated in MATLAB / Simulink environment.

The research is focused on tracking control for hub-angular position based on pre-determined set point. The fuzzy control scheme for hub-angular position control is shown in Figure 6.14. In this FMS, the output variables of end-point acceleration, hub-velocity and hub-angle are taken into account.

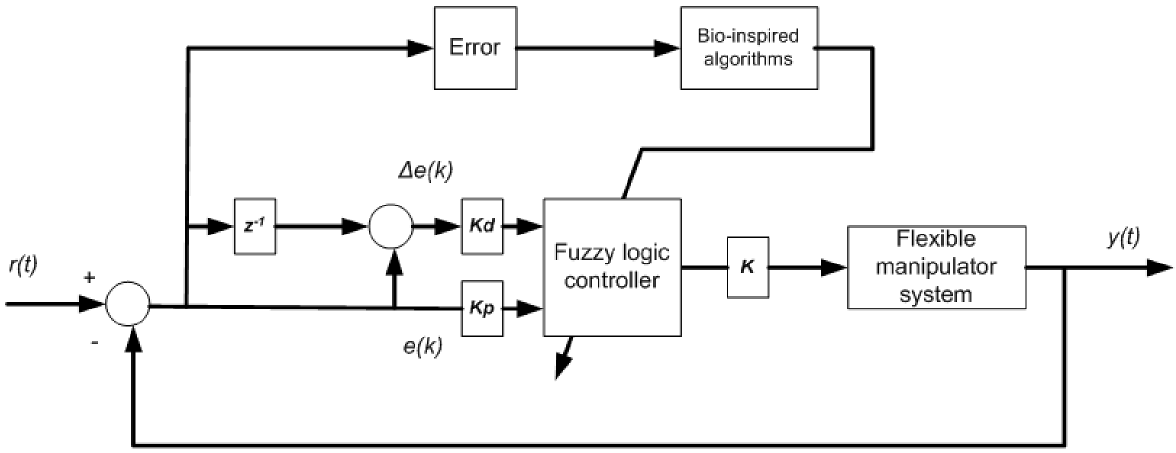


Figure 6.14: The block diagram of PD-fuzzy logic control for single-link flexible manipulator

A zero-order Takagi-Sugeno fuzzy inference system (FIS) is used to design the fuzzy controller. Two inputs and one output PD type fuzzy logic controller is developed as illustrated in Figure 6.14. Thus, the input to flexible manipulator can be represented as follows:

$$u(k) = K_p e(k) + K_d \Delta e(k) \tag{6.16}$$

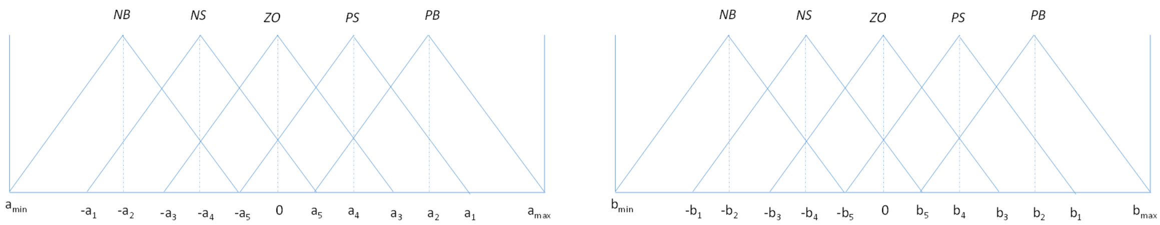
where K_p and K_d are proportional and derivative gains, $e(k)$ is error and $\triangle\ e(k)$ is change-in-error at sample number, k defined as:

$$e(k) = \quad \theta_d(k) - \theta(k) \tag{6.17}$$
$$\Delta e(k) = \quad e(k) - e(k-1) \tag{6.18}$$

where $\theta_d(k)$ is reference input and $\theta(k)$ is derived output.

In the membership function of zero-order Sugeno, two inputs (error and change of error) are represented in the form of linguistic variables and they are characterized via triangularshaped membership functions. There are five linguistic variables for both inputs: negative big (NB), negative small (NS), zero (ZO), positive small (PS) and positive big (PB). The range of each input is in [-1, 1]. The optimisation technique is used to estimate these 10 values in the specified range. Figure 6.15 shows the fuzzy input terms that represent the parameters to be optimised. The values a_1 ... a_5, b_1 ... b_5 will be notated as x_1 ... x_5, x_6 ... x_{10} respectively. a_{max}, b_{max} are large positive values and a_{min}, b_{min} are large negative values. The membership functions are set in symmetrical about zeroth axis.

The parameters of membership functions x_1 ... x_5 are associated with fuzzy error input and x_6 ... x_{10} fuzzy with change of error input. In the experiment, pre-determined scaling factors are considered. For the input scaling factors (K_p and K_d), proportional gain, Kp is set to 0.006, derivation gain, K_d = 0.03. The output fuzzy scaling factor, K is chosen to be large number and is set in this experiment as K = 500.



(a) error, e(k) (b) change of error, $\triangle\ e(k)$

Figure 6.15: The description of fuzzy input term

The fuzzy rule-base represent a correspondence between the particular membership functions of inputs and output. In the experiment, the 5 terms used for each input corresponding to 25 control rules as shown in Figure 6.16 constitute the fuzzy rule-base containing fuzzy statements in If-Then form.

		Change of error				
		NB	***NS***	***ZO***	***PS***	***PB***
error	***NB***	*PB*	*PB*	*PB*	*PS*	*ZO*
	NS	*PB*	*PS*	*PS*	*ZO*	*NS*
	ZO	*PS*	*ZO*	*ZO*	*Z*	*NS*
	PS	*PS*	*ZO*	*NS*	*NS*	*NB*
	PB	*ZO*	*NS*	*NB*	*NB*	*NB*

Figure 6.16: The fuzzy rule-base

The zero-order Takagi-Sugeno FIS output is selected in the range [0, 1]. The constant value for each term is shown in Table 6.2.

Table 6.2: The fuzzy rule-based output

Torque	Constant value
NB	-1
NS	-0.5
ZO	0
PS	0.5
PB	1

The optimisation algorithms are used in this research to minimise the error between the reference and actual hub-angle displacement. Thus, a cost function to be minimised based on the error can be formed, and here the integral of absolute error is selected

$$\min_{e(k)} f = min \int |e(k)|dt \tag{6.19}$$

where $e(k)$ is the error calculated between the reference and actual output.

6.3.2 Experiments

In the simulations presented here, the flexible manipulator is excited with a bang-bang input. Since membership functions of each input are described symmetrically with respect to zeroth axis. There were five variables needed to optimise the membership values. Hence, a ten-dimensional problem with respect to two inputs was considered. For all the tests, the algorithms used the same population size, n and the maximum number of iterations for a fair comparative evaluation. The basic criteria thus used are as follows:

- Maximum number of population, $n_{max} = 30$.
- Maximum number of iterations, $it_{max} = 30$ (NFE = 900).

The resultant cost functions, $f(x)$ values and desired gains are given in Table 6.3.

Based on Table 6.3, the obtained numerical values can be transcribed into membership function values of the algorithms. Samples of triangular membership function for FA variants are shown in Figure 6.17. The NB and PB values are infinite and all the other membership functions are in the range [-1, 1].

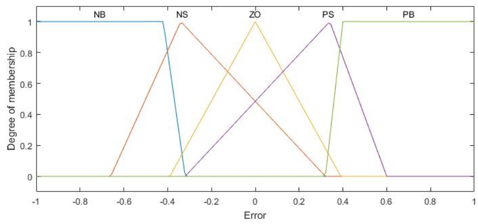
Figure 6.18 shows the convergence graph of the cost function value obtained from the simulation output of the PD fuzzy logic control experiment. As noted, all the proposed algorithms were able to converge to low level as compared to FA and IWO algorithm, except

Table 6.3: Optimised control parameters of membership functions

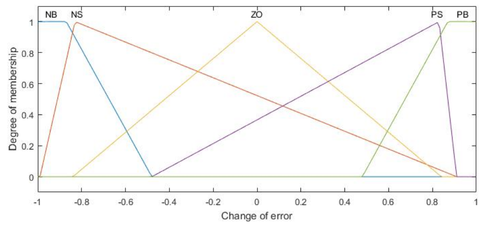
Method		$e(k)$		$\Delta e(k)$	f(x)	time, t
FA	A1	0.66558	B1	0.91496	7.95E+06	1.08E+04
	A2	0.42546	B2	0.87175		
	A3	0.3944	B3	0.83992		
	A4	0.34353	B4	0.83261		
	A5	0.32589	B5	0.48597		
FA-NSF	A1	0.6211	B1	0.99592	3.91E+06	1.10E+04
	A2	0.50674	B2	0.98436		
	A3	0.40401	B3	0.84515		
	A4	0.35146	B4	0.79465		
	A5	0.33532	B5	0.43918		
FA-eSF	A1	1	B1	1	3.94E+06	**1.07E+04**
	A2	0.4872	B2	0.9473		
	A3	0.4525	B3	0.8822		
	A4	0.3837	B4	0.6524		
	A5	0.3271	B5	0.3984		
IWO	A1	0.92377	B1	0.99826	8.01E+06	1.96E+04
	A2	0.92132	B2	0.98533		
	A3	0.63919	B3	0.82928		
	A4	0.47635	B4	0.82503		
	A5	0.4753	B5	0.60572		
IWO-eSSF	A1	1	B1	0.95761	**3.85E+06**	1.77E+04
	A2	0.37789	B2	0.91537		
	A3	0.37443	B3	0.74399		
	A4	0.35766	B4	0.68861		
	A5	0.35647	B5	0.48492		
MIWO-eSSF	A1	0.9894	B1	1	7.90E+06	1.81E+04
	A2	0.731	B2	0.9962		
	A3	0.3621	B3	0.9435		
	A4	0.3429	B4	0.806		
	A5	0.3314	B5	0.4242		
HIWFO	A1	0.80902	B1	0.93263	**3.85E+06**	**1.07E+04**
	A2	0.36178	B2	0.83276		
	A3	0.36045	B3	0.73869		
	A4	0.35841	B4	0.73034		
	A5	0.35746	B5	0.4841		
HIWFO-SF	A1	0.8945	B1	1	4.01E+06	1.07E+04
	A2	0.85898	B2	1		
	A3	0.84402	B3	0.84011		
	A4	0.46384	B4	0.83485		
	A5	0.46274	B5	0.59213		

MIWO-eSSF algorithm. FA-NSF and IWO-eSSF achieved faster convergence and FA had slower convergence in determining the optimal value point.

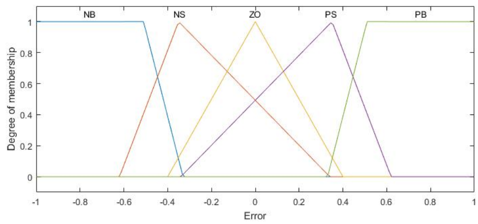
In Figure 6.18, the maximum number of iterations was set small but enough to obtain the optimal solution and avoid time-consuming computations. The derived time domain specifi-



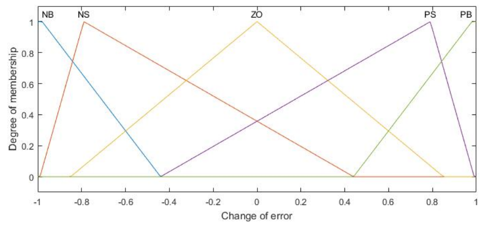
(a) FA error



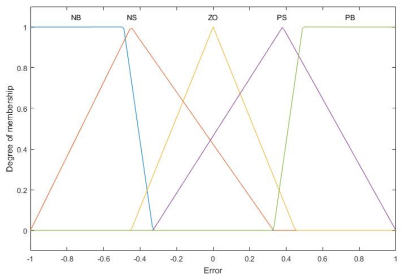
(b) FA change of error



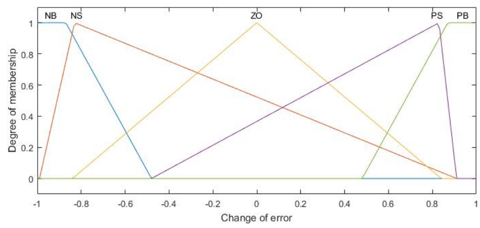
(c) FA-NSF error



(d) FA-NSF change of error



(e) FA-eSF error



(f) FA-eSF change of error

Figure 6.17: Fuzzy error and change of error for FA variants

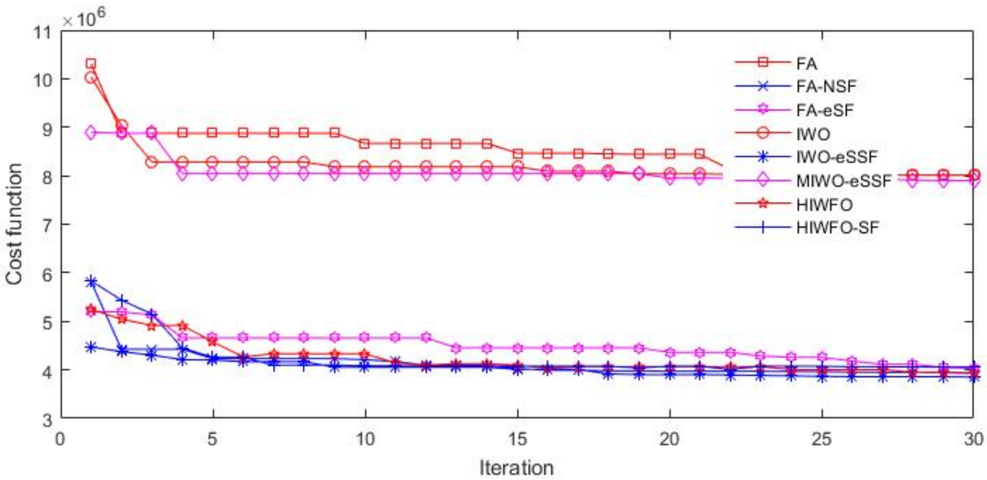
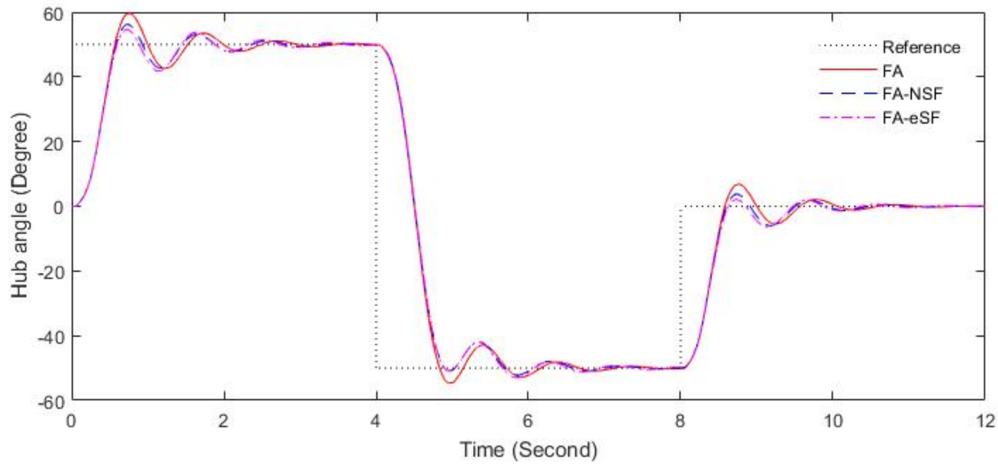
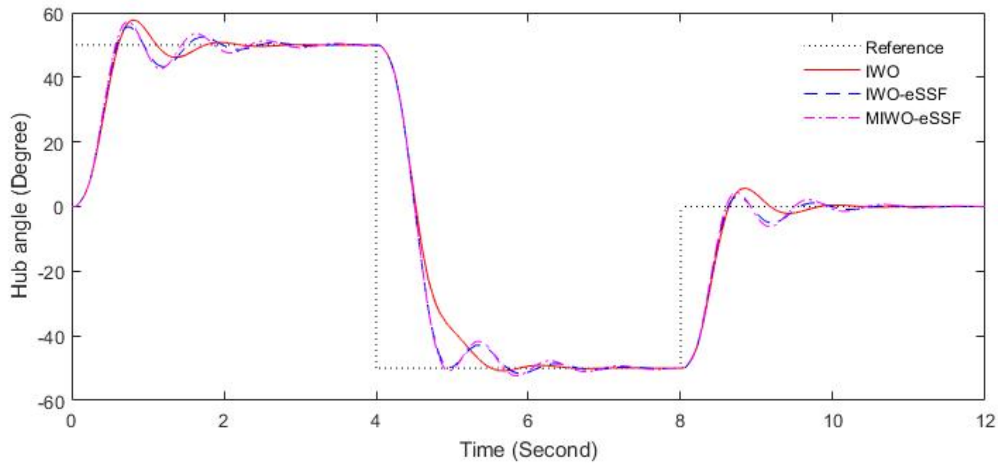


Figure 6.18: Convergence plot for the FMS

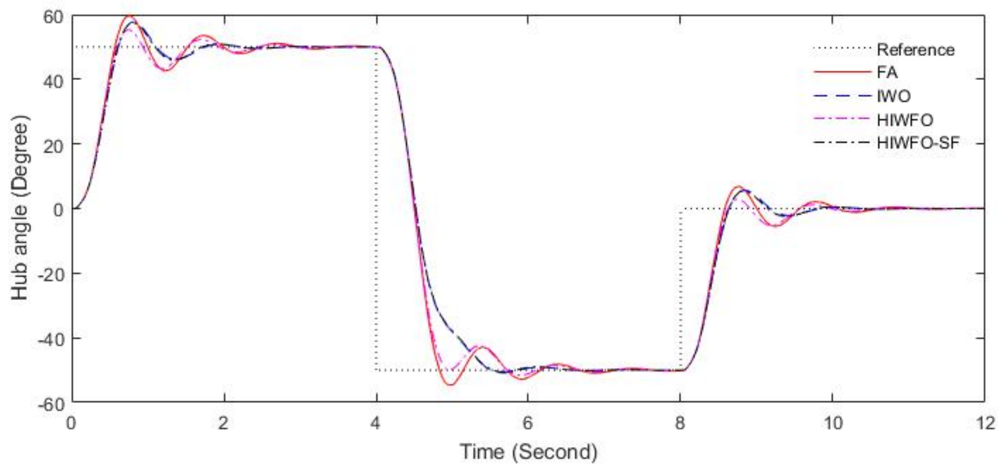
cation results are presented in Table 6.4. As noted FA produced faster rise time but showed the highest response overshoot value. On the other hand, HIWFO, IWO-eSSF and FA-eSF produced competitive rise time value and lower overshoot value than other algorithms. They also managed to obtain shorter settling time.



(a) FA variants



(b) IWO variants



(c) HIWFO and HIWFO-SF

Figure 6.19: Hub-angle response of the single-link flexible manipulator

Table 6.4: Results of time domain parameters

Parameters	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
RiseTime, (tr1,s)	**0.33**	**0.34**	**0.34**	0.36	0.35	**0.34**	0.35	0.36
SettlingTime (ts1,s)	2.76	2.64	2.62	1.66	2.32	2.66	2.31	**1.63**
Overshoot (PO1,%)	19.28	12.43	**9.28**	15.36	**11.26**	14.31	**10.63**	15.06
RiseTime, (tr2,s)	**0.48**	**0.49**	**0.49**	0.82	0.5	**0.49**	0.5	0.83
SettlingTime (ts2,s)	6.02	6.34	**5.96**	**5.39**	**5.64**	6.36	**5.64**	5.4
Overshoot (PO2,%)	9.18	4.37	5.9	1.68	3.28	4.67	3.14	1.16
Undershoot (PU2,%)	99.86	99.59	99.94	100.01	99.84	99.54	99.83	100.03
RiseTime, (tr3,s)	**0.34**	**0.36**	**0.36**	0.38	**0.36**	**0.35**	**0.36**	0.38
SettlingTime (ts3,s)	10.36	10.29	10.22	9.66	**9.81**	10.29	**9.81**	9.64
Overshoot (PO3,%)	2.90E+08	3.00E+08	3.30E+10	3.50E+09	1.70E+08	1.10E+11	**1.60E+08**	4.90E+09
Undershoot (PU3,%)	2.20E+09	4.00E+09	7.80E+11	3.90E+08	2.70E+09	9.90E+09	2.70E+09	5.30E+08

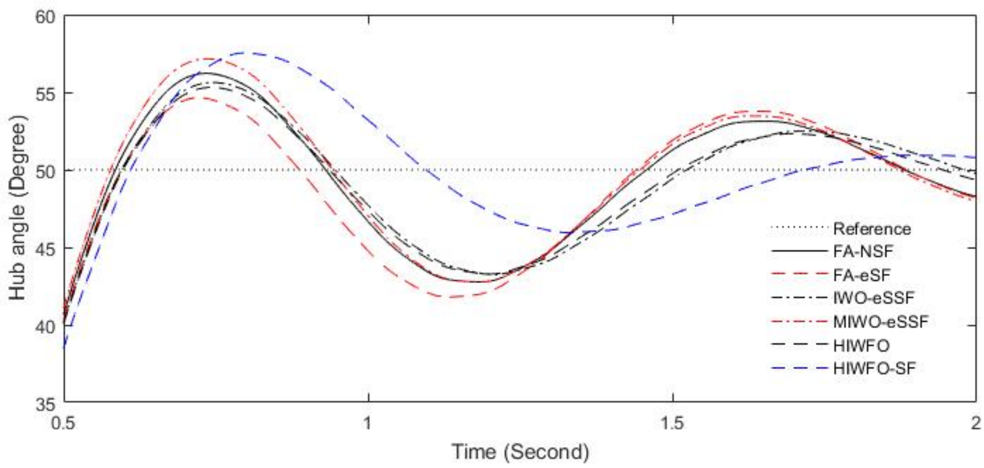


Figure 6.20: The hub-angle response in time range 0.5 - 2.0 seconds

It can be seen in Figure 6.19 and 6.20 that the proposed algorithms achieved better results and were able to damp overshoots and undershoots as compared to FA and IWO, although small oscillations were also present. Figure 6.20 compares responses produced by the proposed algorithms to part first step of the bang-bang input. FA-eSF achieved better result as it produced decent rise time and lowest response overshoot. IWO-eSSF and HIWFO also resulted in competitive results.

In this experiment, all of the algorithms were able to determine the membership function values. Further tuning of scaling gains can result in shorter rise time and lower overshoot value. Based on the results obtained, the steady state error was zero, thus resulting in desired steady-state system response.

6.4 Application to Exoskeleton Control of Upper and Lower Limb Models

Assistive robotic devices are increasingly needed to facilitate mobility and rehabilitation requirements of elderly and disabled (Moubarak et al., 2009). Therefore, research interest in upper and lower extremities robot assistance has intensified in the academic and industrial sectors. Exoskeleton is an assistive device designed for mobility and for rehabilitation purpose (Ghassaq et al., 2015). Significant research within academic and industrial sectors in the area of exoskeleton mobility and robot assistance for medical and rehabilitation applications (Ali et al., 2015; Ghassaq et al., 2015; Glowinski et al., 2015; Moubarak et al., 2009).

Human arm model of upper extremities and lower limb exoskeleton model for lower extremities are used in this experiment. The proposed algorithms are used to devise control mechanisms for lower and upper extremities. A set-point tracking position control using proportional, integral and derivative (PID) control is developed as the control mechanism. The bio-inspired algorithms are applied to optimise the controller to achieve preferable manoeuvrability of the model. Figure 6.21 shows a block diagram of the control mechanism used in this experiment. Performances of the proposed algorithms with the control strategy are evaluated and analysed.

6.4.1 Human Arm Movement

In this section, the proposed algorithms are employed for upper limb exoskeleton exercise. The significant research and development efforts have been placed into upper limb exoskeleton such as ARMin III (Nef et al., 2009) and recently NTUH-II (Lin et al., 2014). Most of the applications of upper limb exoskeletons include rehabilitation such as to help stroke and disability patients.

A lot of the research work has been reported using bio-inspired algorithms in the design and development of upper limb exoskeleton. For example, Hassan and Karam (2015) used

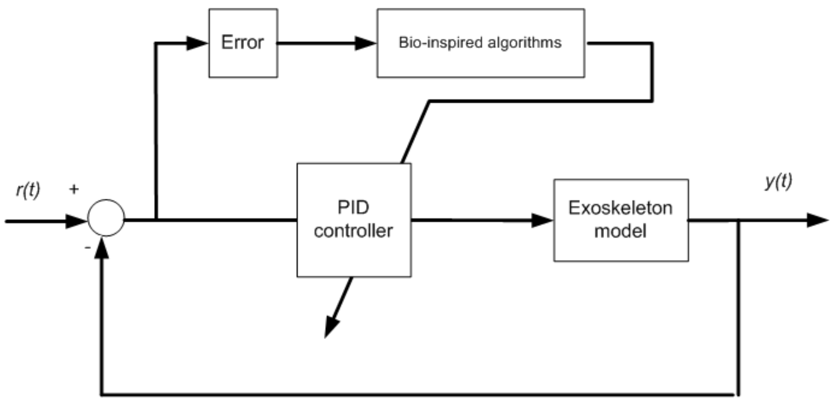


Figure 6.21: PID-based bio-inspired control mechanism of exoskeleton system

PSO in designing the structure of rehabilitation robot arm. Khan et al. (2015) also employed PSO in determining the control gains of upper limb assist exoskeleton robot. Furthermore, Bryson et al. (2015) have reported the use of PSO in the study of optimal design of arm manipulator. Wu et al., (2012) have proposed ABC algorithm to tune the controller of a rehabilitation robot arm.

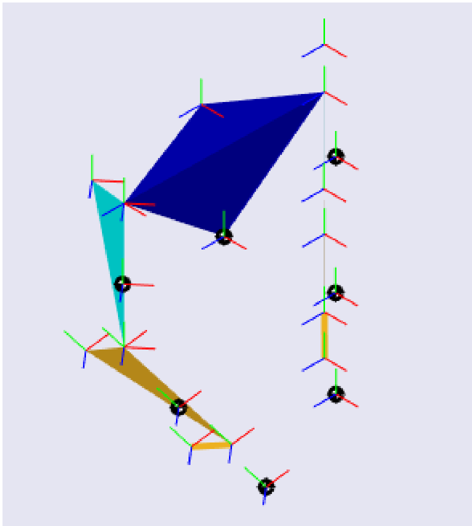
Control Design Mechanism

In the research of upper limb exoskeleton, bio-mechanical arm prototype can be developed in order to represent humanoid upper limb. Bio-mechanical models of the upper limb have been well described by Moubarak et al. (2009). The model can be used to design controller for evaluation and diagnosis for rehabilitation purposes. In this experiment, human model is used as a plant and a controller is developed in order to evaluate the performance of the arm movement. A human arm model proposed by Ali et al. (2015) is used in this experiment. The model is part of research in upper limb exoskeleton. The human parameters used in the human arm model are from Głowiński et al. (2015) and sim-mechanics / MATLAB is used as the software platform. Figure 6.22 show the human arm model developed by Ali et al. (2015) and used in this experiment. Human arm movement is as shown in Figure 6.22a. The work here focuses on the motion of shoulder joint from static to external movement. Elbow joint is set to move in flexion and extension. The movements of elbow and shoulder are shown in Figures 6.22b and 6.22b and the actual range of motion of the shoulder and elbow given in Table 6.5 are as provided by Moubarak et al. (2009).

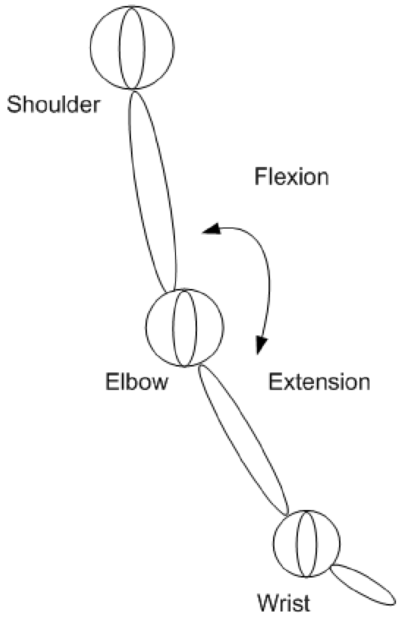
Table 6.5: The human arm movement

		Shoulder rotation	Elbow flexion
Range of motion	Actual	$-80^o - 100^o$	$0^o - 145^o$
	Experiment	$0^o - 40^o$	$0^o - 45^o$

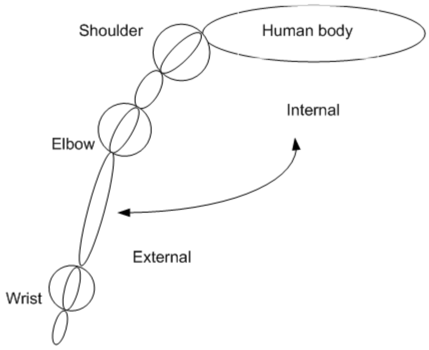
The bio-mechanical arm model can be used for the design and control of a prototype to



(a) Human arm model



(b) Elbow flexion movement



(c) Shoulder rotation movement

Figure 6.22: Human arm model and its basic movement

simplify the interaction mechanism between human and exoskeleton (Moubarak et al., 2009). For example, it can be used to control the forces generated by the upper limb movements during assistive rehabilitation exercise. Therefore, this works only focuses on set point tracking control of the human arm model. PID control is used to reduce the error in positional the human arm. The controller is developed to control the arm movement so that, it follows the desired position. The PID control law used in this experiment is

$$u(t) = k_p e(t) + k_i \int_0^t e(t)dt + k_d \frac{de(t)}{dx} \tag{6.20}$$

where $e(t)$ is the error between actual reading and desired trajectory, k_p, k_i and k_d are gains of proportional, integral and derivative terms of the controller, respectively. The gains of the controller will be optimised by the proposed optimisation algorithms to minimise the error, $e(t)$. The block diagram of the PID controller used in the human arm control is shown in Figure 6.23.

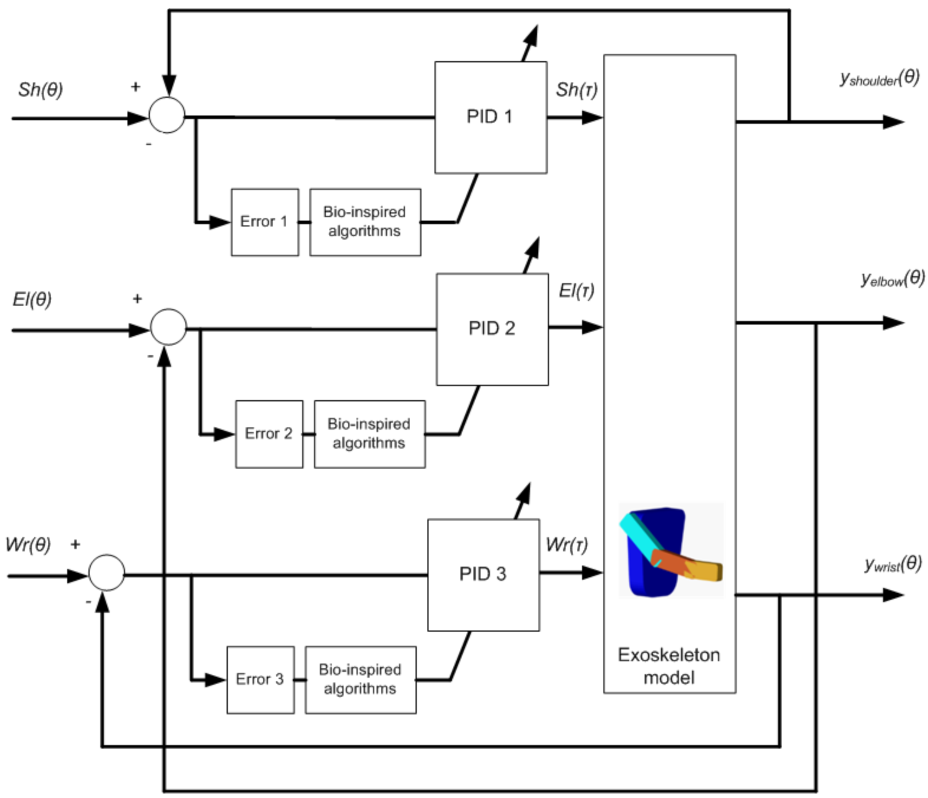


Figure 6.23: The PID controller for the human arm system

As seen in Figure 6.23, the control mechanism consists of three parts. PID 1 refers to the control mechanism of the shoulder, PID 2 is of elbow and PID 3 is of the wrist. $El(\theta)$, $Sh(\theta)$ and $Wr(\theta)$ represent the reference trajectories elbow, shoulder and wrist, respectively and $y_{elbow}(\theta)$, $y_{shoulder}(\theta)$, and $y_{wrist}(\theta)$, represent the actual trajectories of the human arm model. The output of each controller is the produced torque for the respective part.

The bio-inspired algorithms are used to minimise the error of each control loop. The MSE is used as cost function. The e^2 in the MSE represents the square of error between the reference trajectory and the actual trajectory measured from the human arm model. The overall cost function of the system is as follows:

$$f_{human\ \ arm}(x) = \omega_1 MSE_1 + \omega_2 MSE_2 + \omega_3 MSE_3 \tag{6.21}$$

where $f_{humanarm}$ (x) is the cost function, MSE_1 is the mean squared error of the shoulder control loop, MSE_2 is for the elbow and MSE_3 is for the wrist. The aggregation method mentioned in the previous chapter is implemented here. Therefore, the weights ω_1, ω_2 and ω_3 are carefully selected so that their sum is equal to 1.

Experiments

Simulation were performed for only one human arm. The algorithms were used to tune and optimise the controller to determine the best gains for the system. The performance of the control system with the used optimisation algorithms is evaluated. The error and torque characteristics are also monitored.

For all the tests, the algorithms used the same population size, n and the maximum number of iterations for a fair comparative evaluation. This basic criteria used were thus as follows:

- Maximum number of population, n_{max} = 30.
- Maximum number of iterations, it_{max} = 30 (NFE = 900).

The minimum cost functions, $f(x)$ values achieved with the respective desired gains of each controller are as shown in Table 6.6.

Table 6.6: Optimised control parameters of the human arm model

	FA	FA-NSF	FA-eSF	IWO	IWO-eSSF	MIWO-eSSF	HIWFO	HIWFO-SF
f(x)	4.39E-02	**3.07E-02**	3.45E-02	3.63E-02	4.01E-02	3.20E-02	3.63E-02	3.50E-02
$t, (sec)$	1.08E+03	1.15E+03	1.05E+03	1.14E+03	**1.02E+03**	1.08E+03	1.05E+03	1.05E+03
x_1	483.321	**397.505**	308.509	352.29	473.254	496.29	497.451	304.583
x_2	484.666	**38.042**	333.82	355.445	459.841	83.038	77.076	280.149
x_3	258.553	**499.949**	462.484	218.787	22.408	478.965	498.341	372.232
x_4	326.401	**477.913**	488.942	404.934	283.08	335	499.915	500
x_5	449.641	**378.743**	348.925	482.203	27.298	413.828	259.328	241.977
x_6	64.304	**463.003**	496.74	1.084	411.665	442.146	422.946	389.404
x_7	308.661	**50.335**	29.771	179.289	151.019	458.518	71.798	272.79
x_8	440.777	**128.142**	412.867	120.782	110.615	445.179	212.872	191.365
x_9	103.62	**188.722**	314.436	108.247	147.745	141.577	445.296	57.438

Table 6.6 shows the fitness value as well as the respective optimised parameters of control mechanism for the human arm model. Figure 6.24 shows the convergence plots of the best fitness values of the algorithms. It is noted that, FA-NSF and MIWO-eSSF achieved the lowest fitness value as compared to the other algorithms. However, all the algorithm did not appear to converge further after 15 iterations. The fitness values of the algorithms were also low to the optimal value point.

In the simulations, the elbow and shoulder joints were actuated individually. In this experiment, wrist was considered static and hence, the wrist movement was followed based on the elbow movement. Figure 6.25a shows the trajectory based on the movements from Table 6.5. In this case, the tracking was based on the movement of elbow and arm. The starting point was standing position in normal condition. Both shoulder and elbow were initialised to zero position. Zero position refers to human in standing where both elbow and shoulder are in straight downward position. For this experiment, the shoulder moved in outward rotation while the elbow was raised by moving in flexion and extension condition.

The actual trajectory and the output tracking of the controller optimised by the algorithms are also compared in Figure 6.25. For all the algorithms, the PID control successfully tracked

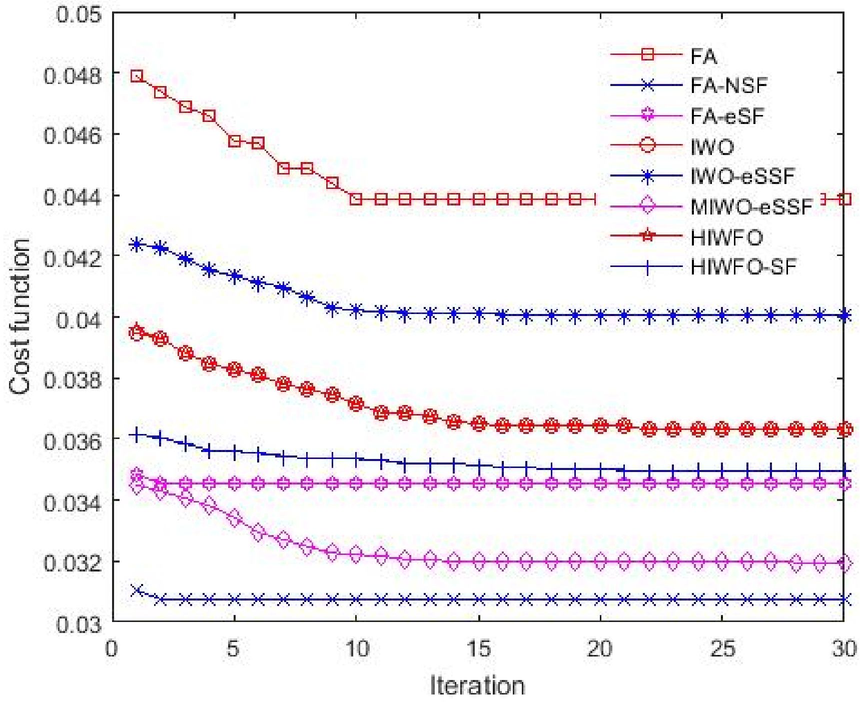
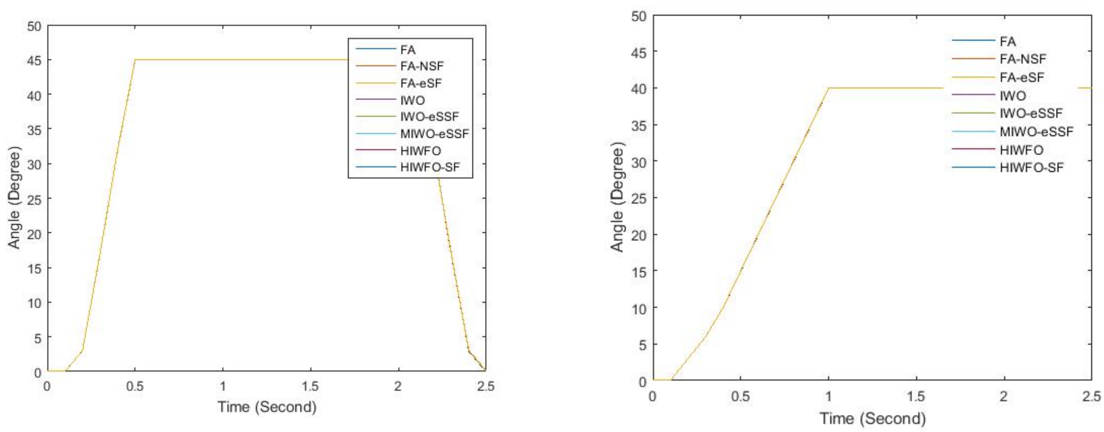


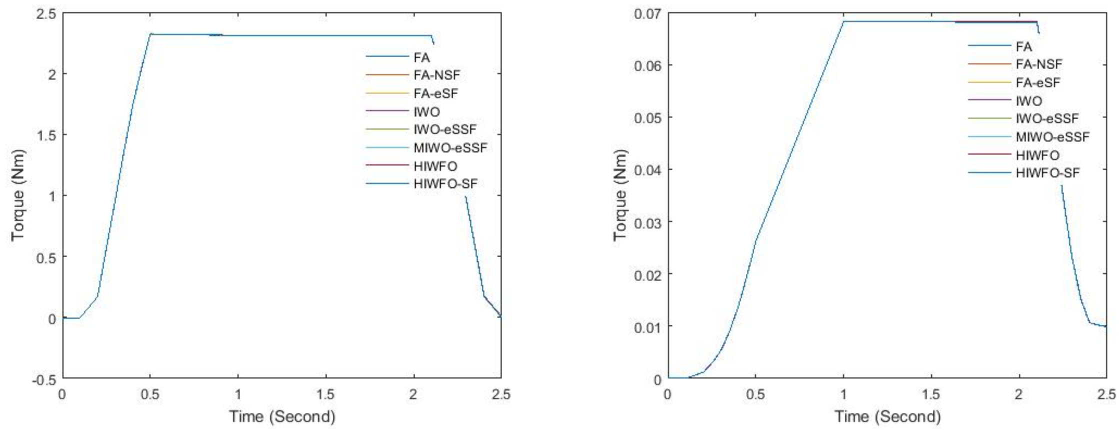
Figure 6.24: The convergence plot of human arm model



(a) Elbow

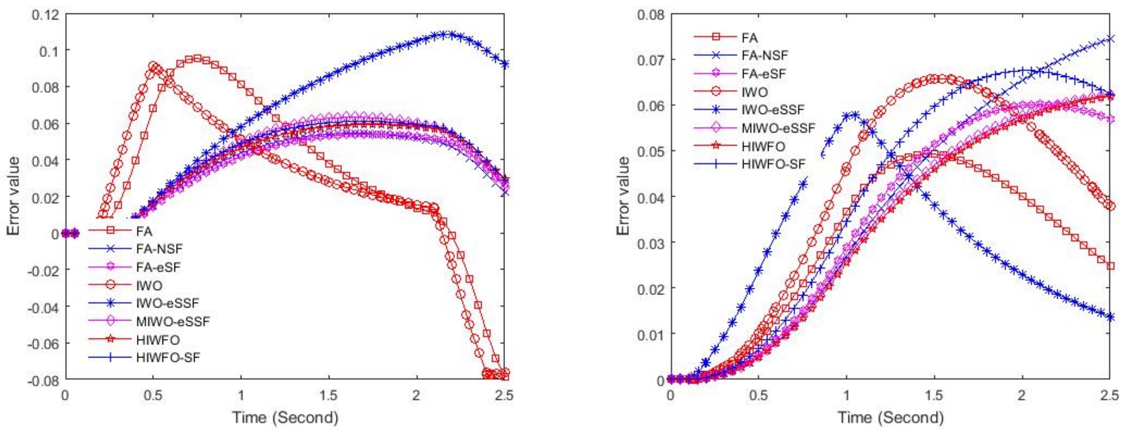
(b) Shoulder

Figure 6.25: The actual and desired movements



(a) Elbow (b) Shoulder

Figure 6.26: The torque value of the human arm model



(a) Elbow (b) Shoulder

Figure 6.27: Positional errors

the movement. The torque profiles in Figure 6.26 also shows the same pattern. The movement was successfully achieved with the range of torque not more than 2.5 Nm for the elbow and less than 0.07 for the shoulder. The shape of the torque followed the tracking point similar to that noted by Glowinski (2015). As noted in 6.26, after 2 seconds, the torque decreased as the elbow begun to extend to normal condition (straight downward) and the shoulder movement remained in place.

The response errors for elbow and shoulder are shown in Figures 6.27a and 6.27b, respectively. As noted FA-eSF, FA-NSF, MIWO-eSSF, HIWFO and HIWFO-SF achieved low range of error for elbow. On the other hand, HIWFO-SF, FA-NSF and IWO produced higher range of response error for shoulder. However, in this case, all the error values were insignificant as they were less than 0.08. As the range of torque for elbow was higher, the movement of the arm depended on elbow movement. All the optimisation algorithms successfully produced desired parameters resulting good control and tracking of the movement. It can be conclude that the proposed algorithms produced low error in the elbow and could be considered better than other algorithms.

6.4.2 Lower Limb Exoskeleton Movements

In this section, control mechanisms for lower-extremities exoskeleton assistance are devised and evaluated with the proposed algorithms. The lower limb exoskeleton system model as described by Ghassaq et al. (2015) is used in this experiment. The exoskeleton system is to control and balance both lower limb exoskeleton and humanoid movement in a walking cycle.

The humanoid model structure has been developed by Ghasaq et al. (2015) in Visual Nastran 4D (VN4D) environment. The model segmentation parameters have been built in the humanoid model based on Winter (2009). The evaluations are performed using MATLAB 2012 / Simulink linked with VN4D. In VN4D, simulation of a complex mechanical system is easily developed (Ghasaq et al., 2015; Shih-Liang et al., 2001). For simulated walking, a specific trajectory of the knee joint movement is set using Clinical Gait Analysis (CGA) data with reference to Kirtley (2006). The exoskeleton model is with reference to the Proyecto Control Montaje (PCM) exoskeleton model developed by Virk et al. (2014) and later simplified by Miranda-Linares et al. (2015).

The use of bio-inspired algorithms in the design and development of lower limb exoskeleton has great potential. The algorithm can be used to find optimal parameters of exoskeleton design and also to fine tune the control structure used for the exoskeleton. Lui et al. (2012) used PSO enhanced with simulated annealing to enhance the lower limb exoskeleton design. Long et al. (2016), used GA to optimise a sliding mode controller in lower limb exoskeleton application.

Control design mechanism

In the exoskeleton design, development of control approach play an important role to ensure the exoskeleton always follow the human movement. The main issue is that either the exoskeleton could support the user's body weight in the self-balancing control, which make the exoskeleton system more complex (Ghasaq et al., 2015). Hence, an assistive torque is to be provided by exoskeleton to enhance the ability of human to walk. Considering, for example, an elderly person with exoskeleton for upright walking, the exoskeleton control system will provide the necessary response for appropriate support on the lower limb for the elderly person to walk.

In this research, PID control is developed for knee joint movement. The bio-inspired algorithms are used to optimise and minimise the orientation error for the knee joints while the exoskeleton system is in walking phase. The classical PID control law used is as mentioned in equation (6.20).

The value of $e(t)$ represents the error between actual and reference trajectory, k_p, k_i and k_d are proportional, integral and derivative gains of the PID controller, respectively. The proposed optimisation algorithms are used to fine tune the controller gains to minimise the error, $e(t)$. The block diagram of the PID control used for the lower limb exoskeleton is shown in Figure 6.28.

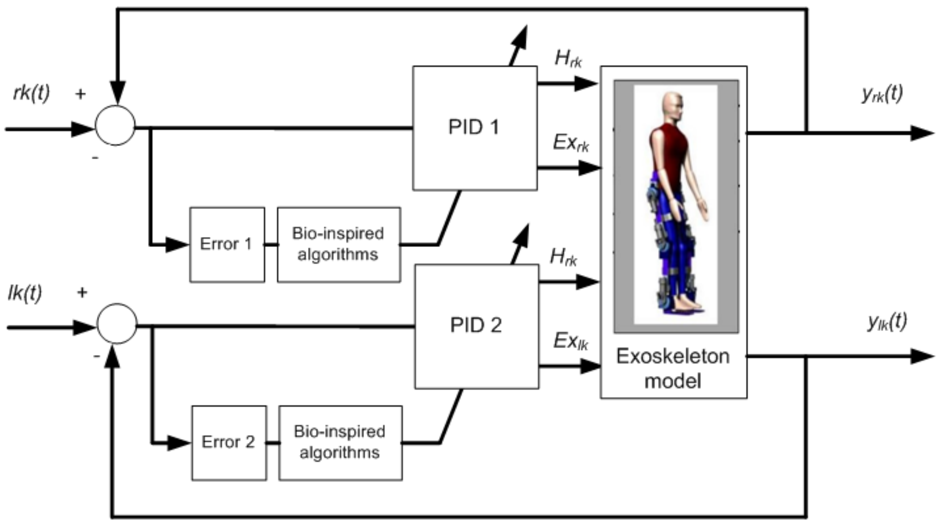


Figure 6.28: Lower limb exoskeleton with PID control

The simulation focused on the trajectory of right and left knee and the ability of the controller to move the exoskeleton model accordingly. The output of the controller is the knee torque of the model which is fed to the humanoid. In Figure 6.28, the Error 1 and 2 need to be minimised, and the proposed algorithms as the optimisation tools in the controller. The performance index of the controller set as the cost function is the MSE as mentioned in equation (6.5). The value of e^2 in the MSE represents the square of error captured between the reference trajectory and the actual trajectory measured from VN4D simulation output of the exoskeleton model. The right knee and left knee joints are controlled based on the cost function:

$$f_{knee}(x) = \omega_1 MSE_1 + \omega_2 MSE_2 \tag{6.22}$$

where $f_{knee}(x)$ is the cost function, MSE_1 is the MSE for the left knee and MSE_2 is for the right knee, ω_1 and ω_2 are the weights selected for the knee joints.

Experiments

Typical walking comprises repeated gait cycle. In this investigations here, simulations are performed for one gait cycle. The controller gains are tuned using the proposed optimisation algorithms as well as FA and IWO algorithms. The performances of the algorithms are assessed based on the extent of MSE achieved and ability to smoothly follow a pre-defined trajectory.

For a fair comparative evaluations of the algorithms, the same population size, n and the maximum number of iterations are used for each. The basic criteria thus used were as follows:

- Maximum size of population, n_{max} = 30.
- Maximum number of iterations, it_{max} = 30 (NFE = 900).

The resulting cost function, $f(x)$ values and desired gains are as given in Table 6.7.

Table 6.7: Optimised control parameters of lower limb exoskeleton

Algorithm	f(x)	$t, (sec)$	x_1	x_2	x_3	x_4	x_5	x_6
FA	1.19277	1.22E+05	5	2.12	0.19	5	1.4	0.23
FA-NSF	**1.1588**	2.36E+05	4.14	1.8	0.22	4.79	2.82	0.28
FA-eSF	**1.14242**	1.99E+05	3.66	1.79	0.23	4.71	1.31	0.25
IWO	1.29199	1.99E+05	3.4	1.24	0.21	2.79	2.22	0.28
IWO-eSSF	1.16381	2.16E+05	4.17	1.79	0.2	4.82	1.7	0.24
MIWO-eSSF	1.18274	2.79E+05	4.17	0.76	0.19	4.44	2.84	0.26
HIWFO	1.1837	3.47E+05	4.1	1.42	0.19	4.76	1.02	0.25
HIWFO-SF	1.19786	4.12E+05	3.84	2.38	0.21	4.25	1.25	0.25

Figure 6.29 shows the convergence graphs of the algorithms. As noted, all the algorithms were able to converge to the optimal value point. FA-eSF, FA-NSF and IWO-eSSF performed better than other algorithms. Other algorithms also achieved decent performance, however, IWO achieved the worst cost function value.

The performance of the controller tuned by the algorithms was also evaluated based on reference tracking performance for both knee joints. The angle trajectories of the knee joints were measured by an angle sensor attached to the exoskeleton model provided by VN4D software. Figure 6.30 shows the achieved angle trajectories of the right and left knee joints.

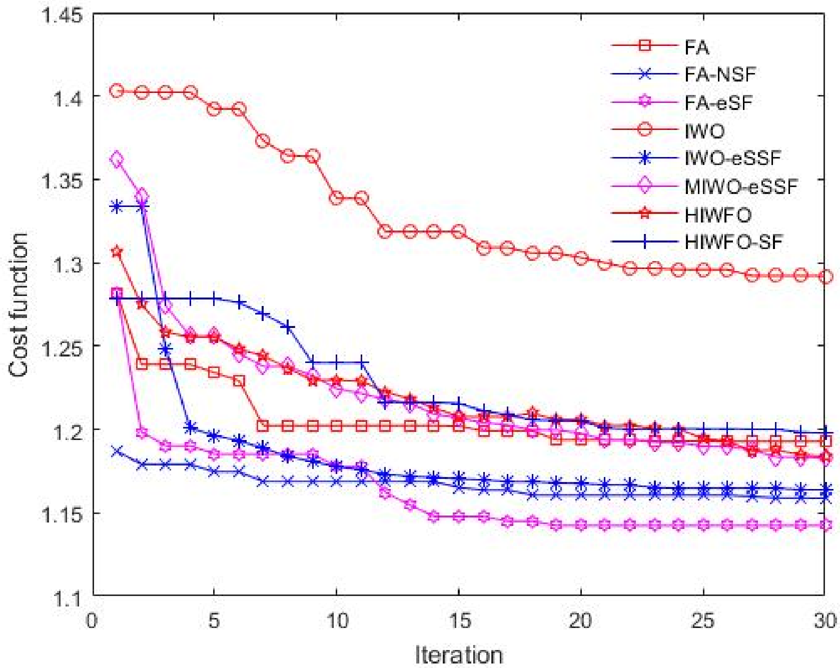
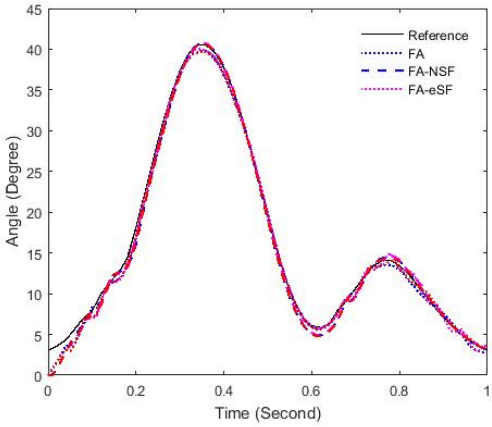


Figure 6.29: The convergence plot of algorithms in lower limb exoskeleton control

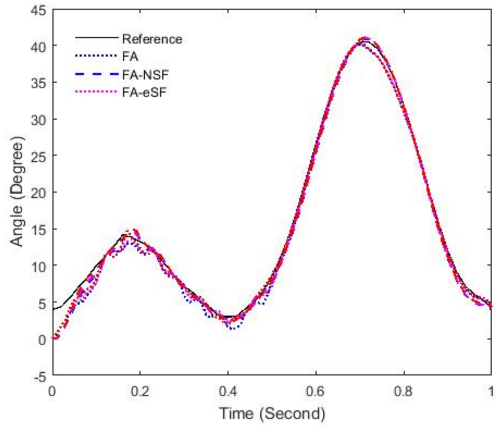
Figure 6.30 provides comparison of the reference angle trajectory with the actual outputs. It is noted that all the algorithms seems to have struggled at the early stage of the walking cycle for both left and right knee joints. After 0.4 seconds, all the algorithms manage to achieve smoother angle response. Figure 6.31 shows zoomed-in actual and desired movements. As noted in Figure 6.31, the control mechanism with IWO variants and both proposed hybrid algorithms performed better than other algorithms. As noted in Figures 6.31a and 6.31b, FA slightly struggled to achieve the reference trajectory and the response fluctuated. The IWO also struggled more than the proposed IWO variants, as noted in Figures 6.31c and 6.31d. Although the response due to the proposed algorithms also showed fluctuation, they followed the angle trajectories and hence, performed better than their respective predecessors.

Table 6.8 shows the minimum and maximum torque profiles of the right and left knee joints of both humanoid and exoskeleton during walking phase. Samples of the torque profile are shown in Figure 6.32 for the case of torque profiles for both humanoid and exoskeleton of the FA-NSF and HIWFO-SF, respectively. Most of the algorithms achieved good results as they showed that the average torque of exoskeleton was less than 30 Nm. This is because, according to Low (2011), the maximum assistive torque for exoskeleton in the application to support knee joint is advised to be lower than 60 Nm.

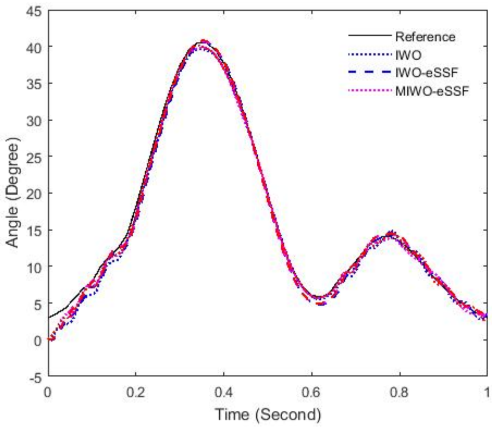
From Table 6.8, it can concluded that FA-NSF, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF resulted lower torque for both right and left knee in the exoskeleton section. Above all, the HIWFO needed the smallest torque. The simulation result justified that the HIWFO and HIWFO-SF result in smoother trajectory and lower response fluctuation.



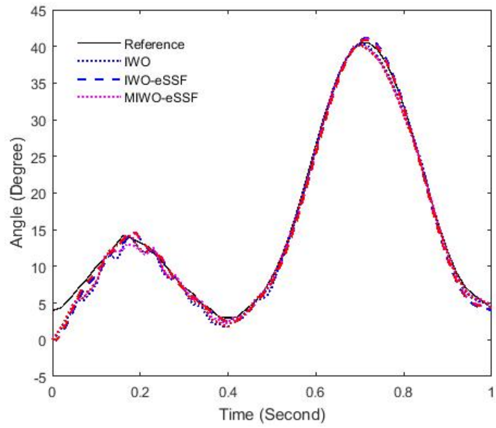
(a) FA



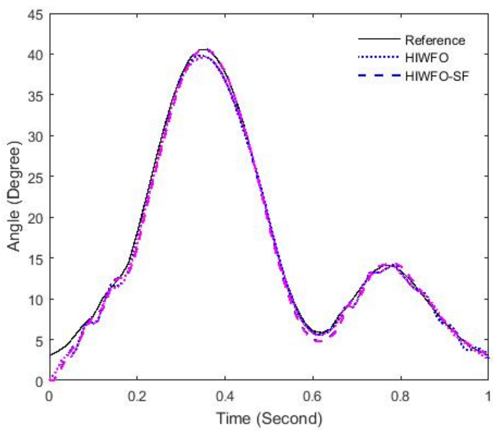
(b) FA



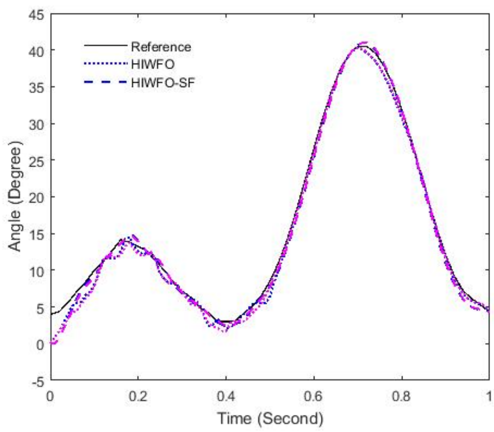
(c) IWO



(d) IWO

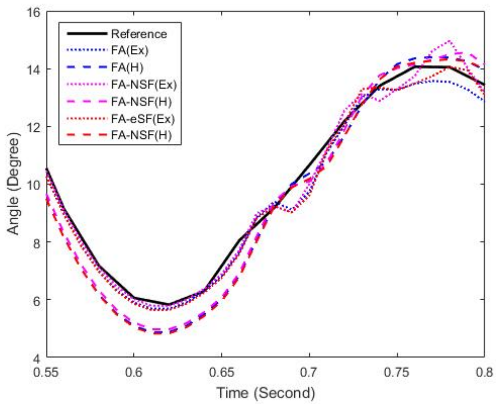


(e) HIWFO and HIWFO-SF

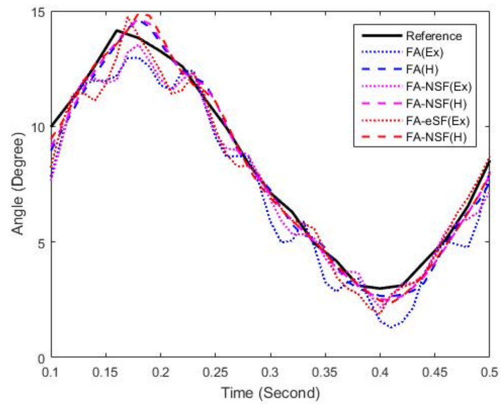


(f) HIWFO and HIWFO-SF

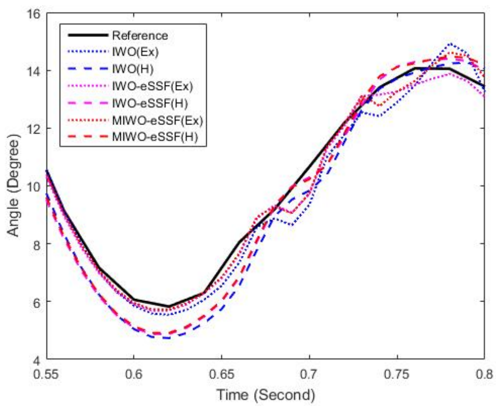
Figure 6.30: Actual and desired movements of right-knee and left-knee



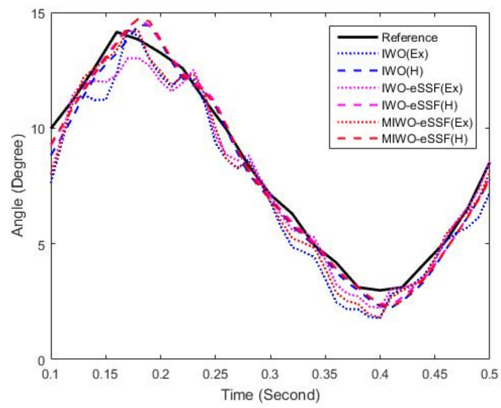
(a) FA



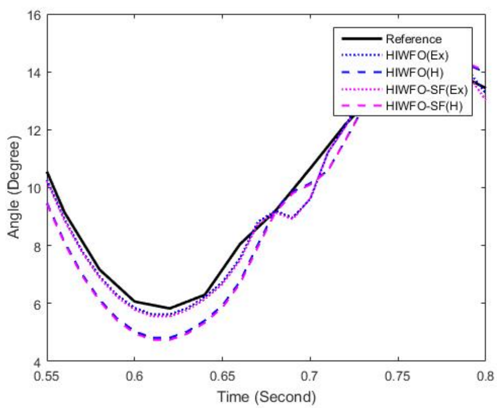
(b) FA



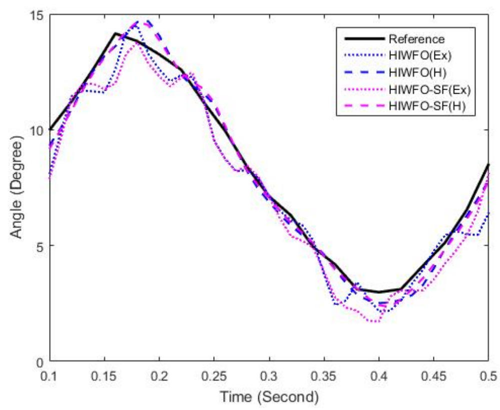
(c) IWO



(d) IWO



(e) HIWFO and HIWFO-SF

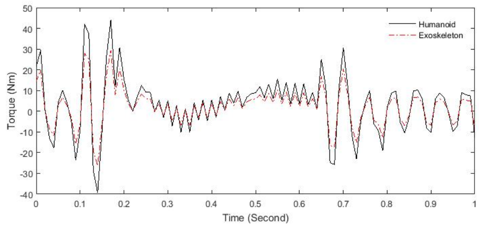


(f) HIWFO and HIWFO-SF

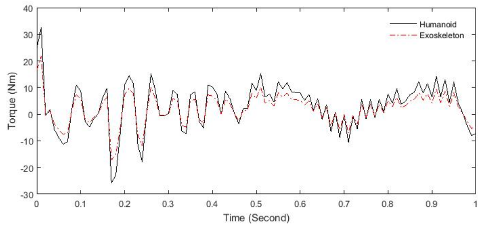
Figure 6.31: Actual and desired movements of right-knee and left-knee (Zoomed-in)

Table 6.8: The min-max (minimum and maximum) torque profile of right and left knee joints

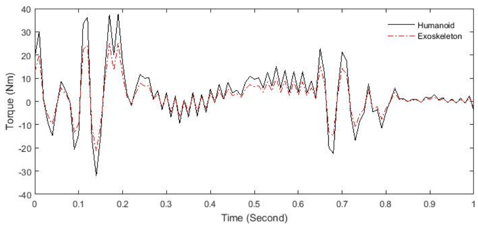
	Humanoid				Exoskeleton			
	Right		Left		Right		Left	
Algorithm	min	max	min	max	min	max	min	max
FA	-21.317	41.858	-14.211	27.905	-32.896	30	-21.93	20
FA-NSF	-39.39	44.086	-26.26	29.391	-25.812	32.529	-17.208	21.686
FA-eSF	-30.964	39.208	-20.643	26.138	-29.706	33.873	-19.804	22.582
IWO	-34.732	40.856	-23.154	27.237	-27.487	33.518	-18.325	22.346
IWO-eSSF	-24.355	41.737	-16.236	27.825	-26.01	32.023	-17.34	21.348
MIWO-eSSF	-32.845	39.545	-21.896	26.363	-25.541	31.76	-17.027	21.173
HIWFO	-30.435	39.216	-20.29	26.144	-24.3	31.584	-16.2	21.056
HIWFO-SF	-32.055	37.727	-21.37	25.152	-23.878	32.766	-15.918	21.844



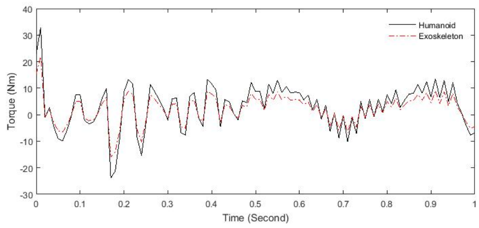
(a) FA-NSF: Elbow



(b) FA-NSF: Shoulder



(c) HIWFO-SF: Elbow



(d) HIWFO-SF: Shoulder

Figure 6.32: The torque value of lower limb exoskeleton system for FA-NSF and HIWFO-SF algorithms

6.5 Summary

In this chapter, the proposed bio-inspired optimisation algorithms have been employed in four engineering applications. The applications comprises system modelling of a TRS, tracking control of an FMS and controller design for two exoskeleton applications.

In the case of system modelling, dynamic models of vertical channels for the TRS have been developed and compared based on the optimisation algorithms. The modelling process is based on input-output data taken in the preliminary experimental work on the actual system. The validation of the derived model produced has been assessed for a given new real input data. The stability and correlation tests have also been carried out to validate the predicted model. IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF have shown better convergence value and have outperformed other algorithms. The algorithms have also produced predicted models that satisfy all the validation tests and the models have been acceptable for the system.

Application of the optimisation algorithms to optimise parameters of a PD fuzzy logic controller (PD-FLC) for position tracking control of the FMS and PID control for trajectory tracking of upper and lower extremity exoskeletons. The performance comparison has been made based on ability of the algorithms to achieve best fitness and convergence to optimal solution.

Chapter 7

Conclusions and Future Work

7.1 Conclusions

In this research, a swarm intelligence firefly algorithm (FA) and ecology-based invasive weed optimisation (IWO) algorithm have been investigated and enhanced to solve global optimisation problems. The research has led to two improved variants of FA algorithm by using non-linear and exponential adaptation and enhancements with spread factor mechanism. The two proposed algorithms are referred to as FA-NSF and FA-eSF. Two new variants of invasive weed algorithm, IWO-eSSF and MIWO-eSSF have also been proposed. A new concept, referred to as normalized seed-spread factor (SSF) has been introduced into the algorithms to examine the rate of evolution of seeds spreading, and this enhances the search process by using local knowledge during spatial dispersal process. Furthermore, two new hybrid algorithms (HIWFO and HIWFO-SF) based on hybridization of the invasive weed and firefly algorithms have been proposed in this work. The hybridization of the algorithms has been achieved by embedding the FA method into IWO algorithm structure to enhance the local search capability of IWO complimenting its already very good exploration capability. HIWFO has been proposed by combining the FA and IWO algorithms, whereas HIWFO-SF is an enhanced version of HIWFO with spread factor mechanism placed in the seed distribution movement and randomization of the firefly section of the algorithm. The algorithms thus formulated have been proposed in an attempt to improve the exploration and exploitation abilities of the search space to avoid premature convergence and achieve better optimum solution. The proposed algorithms have rigorously tested and evaluated with single-objective, constrained and multi-objective optimisation problems. The algorithms have further been tested and evaluated in a set of practical constrained problems and engineering applications. Comparative assessments of performances of the proposed algorithms with the predecessor FA and IWO algorithms have been presented.

Performance evaluations of the proposed algorithms on the single-objective optimisation problems have included unconstrained and constrained problems. In these evaluations, ten standard benchmark functions and 16 CEC 2014 test functions exemplifying single-objective unconstrained optimisation problems have been used. The problems have been formulated with various problem dimensions. The constrained optimisation problems considered have included CEC 2006 test functions and practical design problems with continuous design variables. The practical constrained design problems included the pressure vessel design, spring design, welded beam design, speed reducer and gain design problems. The experimental results have been analysed based on numerical results, convergence plot, success rate of the each algorithm and statistical analysis using Kruskal-Wallis tests. The analyses carried out have shown that the proposed algorithms especially IWO-eSSF, MIWO-eSSF and HIWFO-SF achieve better performance among the algorithms. Although some of the algorithms have performed better in different test functions, HIWFO-SF has been shown to be more efficient among the algorithms. The SF concept has been shown to have significant impact in performance of proposed adaptive FA and IWO algorithms and the HIWFO-SF algorithms, as they have achieved better solution accuracy and faster convergence rate in solving single-objective optimisation problems.

The proposed single-objective optimisation algorithms have successfully been extended to solve multi-objective problems. The proposed adaptive FA and IWO variants have shown potential in finding Pareto-optimal solutions of multi-objective problems. Eight multi-objective problems consisting of unconstrained and constrained multi-objective problems have been considered in the evaluations of the algorithms. A practical design problem related to multi-objective optimisation problem has also been considered. In this research, three performance measurements, namely hypervolume (HV), spacing (SP) and maximum spread (MS) have been used in the analysis of results. HV has been used for evaluation of the convergence criteria, whereas SP and MS have been used for evaluation of the coverage criteria. Analyses of the results have shown that MIWO-eSSF and HIWFO-SF have superior performance among the algorithms in solving multi-objective optimisation problems, by finding better Pareto-optimal solutions with improved coverage and convergence of the Pareto sets to the Pareto front.

The proposed algorithms have been exposed to four engineering applications, namely system modelling of a twin rotor system (TRS), tracking control of a flexible manipulator system (FMS) and controller design for upper extremity and lower extremity exoskeleton applications. For system modelling, the identification of dynamic models in linear parametric form of the TRS have been carried out using the proposed algorithms. Dynamic models of vertical channels for the TRS have been developed and a comparative assessment of the models in replicating the behaviour of the TRS and hence performances of the algorithms has been carried out. The results have shown that, while all algorithms achieved acceptable models for the system, IWO-eSSF, MIWO-eSSF, HIWFO and HIWFO-SF algorithms have better convergence value and superior performance among the algorithms.

The proposed algorithms have further been evaluated in the design of controllers for hub-angle position tracking of the FMS and for tracking control of upper and lower extremity exoskeletons. The performance comparison have been made based on the capability of the algorithms to achieve best fitness and convergence speed. The results have shown that FA-NSF, IWO-eSSF and both proposed new hybrid algorithms have faster convergence among the algorithms. However, HIWFO, IWO-eSSF and FA-eSF outperform other algorithms by achieving competitive response rise time, faster settling time and lower overshoot.

The application of upper limb exoskeleton is referred to the control of human arm model with the exoskeleton. The model consists of shoulder, elbow and wrist controlled for certain arm movement. The results have shown that all the algorithms successfully tuned the controller to track the human arm movement. Among the algorithms, FA-eSF, FA-NSF, MIWO-eSSF, HIWFO and HIWFO-SF are considered more competitive. On the other hand, in the application of the lower limb exoskeleton, the investigations have focused on trajectory control of right and left knee during walking cycle. The results have shown that all the algorithms have converged to acceptable fitness values, resulting in good reference tracking. The IWO-eSSF algorithms have achieved competitive convergence values among the algorithms. IWO-eSSF, HIWFO and HIWFO-SF have performed better in position tracking control and with lower torque profile for right and left knee joints among the algorithms.

In conclusion, the proposed algorithms have shown to be capable in solving single and multi-objective optimisation problems of unconstrained and constrained types in various dimensions. The algorithms can deal with complex multi-objective unconstrained and constrained problems by achieving diverse Pareto optimal solutions. The proposed algorithms have shown competitive performances relative to one another and have outperformed their predecessor, FA and IWO algorithms in solving global optimisation problems.

7.2 Future Work

Potential areas of research that could not be carried out due to time constraints, and may be explored in the future, include;

1. Include knowledge sharing and memory into the research algorithm.

   The proposed algorithms mainly aimed to countermeasure the weaknesses of the basic algorithms. The potential to use information of local knowledge during iteration process is still huge to be explored to enhance the original algorithm. So far, IWO and FA algorithms are memory-less optimisation algorithms and have tendency to get stuck at local optima in the early iterations. In the nature, fireflies always move in groups and weeds are often found in more than one plant, the need to implement memory-based algorithm for these algorithms is one of the future work to be explored. The aim is not only to maintain the natural life phenomenon in each algorithm, but to help the algorithm to avoid unnecessary pace at any local extreme point in local exploration and be able to accelerate to the global optimum point at the end of the search process.

2. Apply the research algorithm for advanced control of exoskeleton system

Developing control system for exoskeleton application is a challenge especially in a real system. Various types of control system approaches have been developed to control an exoskeleton device. The development of advanced non-linear control strategies such as fuzzy type-II control, sliding mode control and computed torque control are essential to handle the non-linear properties of the system. The optimisation algorithms developed in this research can be applied to design such advanced controller strategies.

3. Application to industrial multi-objective optimisation problems

   The proposed algorithms have potential in solving high dimensional single-objective problems as well as multi-objective optimisation problems. Therefore, the need to improve the algorithm to adapt and meet the real life test environment is one of the potential future works. Real life industrial applications such as water distribution problems, smart–grid distribution problems and other real life industrial applications usually have more than one conflicting objectives that not only affect the solutions, they also affect the human and environment. The proposed optimisation algorithms can be used to solution of such complex industrial problems.

4. Implementation of the algorithms in real-time applications

   Most of the works presented in the literature relating to FA and IWO algorithms are reported in simulation tests. Hence, there is some huge potential for the future work for the algorithms to work in solving real time optimisation problems. Current methods such as parallel computing, high performance computerization and so on could be applied to modify the algorithm in adopting and solving real-time problems. The real-time applications may include robotics, industrial and biomedical engineering problems.

References

Abbass, H. A., & Sarker, R. (2002). The pareto differential evolution algorithm. *International Journal on Artificial Intelligence Tools*, *11*(04), 531–552.

Abdullah, A., Deris, S., Mohamad, M. S., & Hashim, S. Z. M. (2012). A new hybrid firefly algorithm for complex and nonlinear problem. *Distributed Computing and Artificial Intelligence*, (pp. 673–680), Springer.

Abedinia, O., Amjady, N., & Naderi, M. S. (2012). Multi-objective environmental/economic dispatch using firefly technique. *2012 11th International Conference on Environment and Electrical Engineering (EEEIC), Venice, Italy, May 2012*, (pp. 461–466), IEEE.

Abu-Al-Nadi, D. I., Alsmadi, O. M., Abo-Hammour, Z. S., Hawa, M. F., & Rahhal, J. S. (2013). Invasive weed optimization for model order reduction of linear mimo systems. *Applied Mathematical Modelling*, *37*(6), 4570–4577.

Aguirre, A. H., Zavala, A. M., Diharce, E. V., & Rionda, S. B. (2007). Copso: Constrained optimization via pso algorithm. *Center for Research in Mathematics (CIMAT). Technical Report No. I-07-04/22-02-2007*.

Ahmad, S. M., Chipperfield, A. J., & Tokhi, M. O. (2001). Parametric modelling and dynamic characterization of a two-degree-of-freedom twin-rotor multi-input multi-output system. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *215*(2), 63–78.

Ahmadi, M., & Mojallali, H. (2012). Chaotic invasive weed optimization algorithm with application to parameter estimation of chaotic systems. *Chaos, Solitons & Fractals*, *45*(9), 1108–1120.

Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing*, *23*(4), 1001–1014.

Akbari, R., & Ziarati, K. (2011). A cooperative approach to bee swarm optimization. *Journal of Information Science and Engineering*, *27*(3), 799–818.

Alam, M. S., & Tokhi, M. O. (2007). Modelling of a twin rotor system: a particle swarm optimization approach. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *221*(3), 353–375.

Alavandar, S., Jain, T., & Nigam, M. J. (2010). Hybrid bacterial foraging and particle swarm optimisation for fuzzy precompensated control of flexible manipulator. *International Journal of Automation and Control*, *4*(2), 234–251.

Aldebrez, F. M., Darus, I. Z. M., & Tokhi, M. O. (2004). Dynamic modelling of a twin rotor system in hovering position. *First International Symposium on Control, Communications and Signal Processing, March 2004*, (pp. 823–826), IEEE.

Ali, M. M., Khompatraporn, C., & Zabinsky, Z. B. (2005). A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems. *Journal of Global Optimization*, *31*(4), 635–672.

Ali, S. K., Tokhi, M. O., Ishak, A. J., & Ghasaq, A. L. (2015). Pid and adaptive spiral dynamic

algorithm in controlling human arm movements. *Assistive Robotics: Proceedings of the 18th International Conference on CLAWAR 2015*, (pp. 87–94), World Scientific.
Al-Ta'i, Z., & Al-Hameed, O. A. (2013). Comparison between pso and firefly algorithms in fingerprint authentication. *International Journal of Eng. and Innovative Technology (IJEIT)*, *3*, 421–425.
Apostolopoulos, T., & Vlachos, A. (2010). Application of the firefly algorithm for solving the economic emissions load dispatch problem. *International Journal of Combinatorics*, *2011*.
Arora, J. (2004). *Introduction to optimum design*. Academic Press.
Auger, A., Bader, J., Brockhoff, D., & Zitzler, E. (2009). Theory of the hypervolume indicator: optimal μ-distributions and the choice of the reference point. *Proceedings of the tenth ACM SIGEVO workshop on Foundations of genetic algorithms, Orlando USA, January 2009*, (pp. 87–102), ACM.
Azad, A. (1994). *Analysis and design of control mechanism for flexible manipulator systems* (Unpublished doctoral dissertation). Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.
Babu, B. V., & Gujarathi, A. M. (2007). Multi-objective differential evolution (mode) for optimization of supply chain planning and management. *IEEE Congress on Evolutionary Computation, 2007. CEC 2007, Singapore, September 2007*, (pp. 2732–2739).
Banati, H., & Bajaj, M. (2011). Fire fly based feature selection approach. *IJCSI International Journal of Computer Science Issues*, *8*(4).
Barisal, A. K., & Prusty, R. C. (2015). Large scale economic dispatch of power systems using oppositional invasive weed optimization. *Applied Soft Computing*, *29*, 122–137.
Basak, A., Maity, D., & Das, S. (2013). A differential invasive weed optimization algorithm for improved global numerical optimization. *Applied Mathematics and Computation*, *219*(12), 6645–6668.
Belegundu, A. D. (1983). Study of mathematical programming methods for structural optimization. *Dissertation Abstracts International Part B: Science and Engineering[DISS. ABST. INT. PT. B- SCI. & ENG.],*, *43*(12), 1983.
Belegundu, A. D., & Arora, J. S. (1985). A study of mathematical programming methods for structural optimization. part i: Theory. *International Journal for Numerical Methods in Engineering*, *21*(9), 1583–1599.
Bernstein, D. S. (2006). Optimization r us [from the editor]. *Control Systems, IEEE*, *26*(5), 6–7.
Bhattacharya, R., Bhattacharyya, T. K., & Saha, S. (2011). Sidelobe level reduction of aperiodic planar array using an improved invasive weed optimization algorithm. *2011 IEEE Applied Electromagnetics Conference (AEMC), Kolkata, India, December 2011*, (pp. 1–4), IEEE.
Billings, S. A., & Zhu, Q. M. (1994). Nonlinear model validation using correlation tests. *International Journal of Control*, *60*(6), 1107–1120.
Binh, T. T., & Korn, U. (1997). Mobes: A multiobjective evolution strategy for constrained optimization problems. In *The third international conference on genetic algorithms (mendel 97)* (Vol. 25, p. 27).
Binitha, S., & Sathya, S. S. (2012). A survey of bio inspired optimization algorithms. *International Journal of Soft Computing and Engineering*, *2*(2), 137–151.
Blum, C., & Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, *35*(3), 268–308.
Bojic, I., Podobnik, V., Ljubi, I., Jezic, G., & Kusek, M. (2012). A self-optimizing mobile network: Auto-tuning the network with firefly-synchronized agents. *Information*

Sciences, *182*(1), 77–92.

Bracken, J., & McCormick, G. P. (1968). *Selected applications of nonlinear programming* (Tech. Rep.). DTIC Document.

Bryson, J. T., Jin, X., & Agrawal, S. K. (2015). Optimal design of cable-driven manipulators using particle swarm optimization. *Journal of Mechanisms and Robotics*, *8*(4). doi: 10.1115/1.4032103

Bulatović, R. R., Bošković, G., Savković, M. M., & Gašić, M. M. (2014). Improved cuckoo search (ics) algorthm for constrained optimization problems. *Latin American Journal of Solids and Structures*, *11*(8), 1349–1362.

Ch, S., Sohani, S., Kumar, D., Malik, A., Chahar, B. R., Nema, A. K., … Dhiman, R. C. (2014). A support vector machine-firefly algorithm based forecasting model to determine malaria transmission. *Neurocomputing*, *129*, 279–288.

Chandrasekaran, K., & Simon, S. P. (2012). Firefly algorithm for reliable/emission/economic dispatch multi objective problem. *International Review of Electrical Engineering*, *7*(1).

Chandrasekaran, K., & Simon, S. P. (2013). Optimal deviation based firefly algorithm tuned fuzzy design for multi-objective ucp. *IEEE Transactions on Power Systems*, *28*(1), 460–471.

Chandrasekaran, K., Simon, S. P., & Padhy, N. P. (2014). Scuc problem for solar/thermal power system addressing smart grid issues using ff algorithm. *International Journal of Electrical Power & Energy Systems*, *62*, 450–460.

Chaves-González, J. M., & Vega-Rodríguez, M. A. (2014). A multiobjective approach based on the behavior of fireflies to generate reliable dna sequences for molecular computing. *Applied Mathematics and Computation*, *227*, 291–308.

Chen, H., Zhou, Y., He, S., Ouyang, X., & Guo, P. (2013). Invasive weed optimization algorithm for solving permutation flow-shop scheduling problem. *Journal of Computational and Theoretical Nanoscience*, *10*(3), 708-713. doi: doi:10.1166/jctn.2013.2759

Chen, K., Zhou, Y., Zhang, Z., Dai, M., Chao, Y., & Shi, J. (2016). Multilevel image segmentation based on an improved firefly algorithm. *Mathematical Problems in Engineering*, *2016*.

Chen, Z., Wang, S., Deng, Z., & Zhang, X. (2011). Tuning of auto-disturbance rejection controller based on the invasive weed optimization. *2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications (BIC-TA), 27-29 September 2011, Penang, Malaysia*, (pp. 314–318), IEEE.

Chifu, V. R., Salomie, I., Chifu, E. Ş., Pop, C. B., Valea, D., Lupu, M., & Antal, M. (2016). Hybrid invasive weed optimization method for generating healthy meals. *Soft Computing Applications*, (pp. 335–351), Springer.

Chinchuluun, A., & Pardalos, P. M. (2007). A survey of recent developments in multiobjective optimization. *Annals of Operations Research*, *154*(1), 29–50.

Coelho, L. D. S., de Andrade Bernert, D. L., & Mariani, V. C. (2011). A chaotic firefly algorithm applied to reliability-redundancy optimization. *2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, 5-8 June*, (pp. 517–521), IEEE.

Coelho, L. D. S., & Mariani, V. C. (2012). Firefly algorithm approach based on chaotic tinkerbell map applied to multivariable pid controller tuning. *Computers & Mathematics with Applications*, *64*(8), 2371–2382.

Coello, C. A. C. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. *Computer Methods in Applied Mechanics and Engineering*, *191*(11), 1245–1287.

Coello, C. A. C. C. (2001). A short tutorial on evolutionary multiobjective optimization. *Evolutionary Multi-Criterion Optimization*, (pp. 21–40), Springer.

Coello Coello, C. A. (2000). Use of a self-adaptive penalty approach for engineering optimization problems. *Computers in Industry*, *41*(2), 113–127.

Das, A. K., Majumdar, R., Krishnanand, K., & Panigrahi, B. K. (2011). Economic load dispatch using hybridized differential evolution and invasive weed operation. *2011 International Conference on Energy, Automation, and Signal (ICEAS), Bhubaneswar, Odisha, December 2011*, (pp. 1–5), IEEE.

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

Deb, K., & Goel, T. (1999). Evolutionary multi-criterion optimization. *Evolutionary Algorithms in Engineering and Computer Science*, 135–162.

Debbarma, S., Saikia, L. C., & Sinha, N. (2014). Solution to automatic generation control problem using firefly algorithm optimized i λ d μ controller. *ISA transactions*, *53*(2), 358–366.

Dorigo, M., Maniezzo, V., & Colorni, A. (1996). Ant system: optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *26*(1), 29–41.

Eberhart, R. C., & Kennedy, J. (1995). A new optimizer using particle swarm theory. *Proceedings of The Sixth International Symposium on Micro Machine and Human Science*, *1*, (pp. 39–43), New York, NY.

Efren, M.-M. (2009). Constraint-handling in evolutionary optimization. *Pland: Springer*.

EI-Sawya, A. A., Zakib, E. M., & Rizk-Allhb, R. M. (2013). A novel hybrid ant colony optimization and firefly algorithm for multi-objective problems. *International Journal of Mathematical Archive, IJMA, ISSN 2229-5046*, *4*(1).

Elkaranshway, H. A., Bayoumi, E. H. E., & Soliman, H. M. (2011). Pso-based robust pid control for flexible manipulator systems. *International Journal of Modelling, Identification and Control*, *14*(1-2), 1–12.

El-Sawy, A. A., Zaki, E. M., & Rizk-Allah, R. M. (2012). A novel hybrid ant colony optimization and firefly algorithm for solving constrained engineering design problems. *Journal of Natural Sciences and Mathematics*, *6*(1).

Farahani, S. M., Abshouri, A. A., Nasiri, B., & Meybodi, M. R. (2011). A gaussian firefly algorithm. *International Journal of Machine Learning and Computing*, *1*(5), 448–453.

Farahani, S. M., Abshouri, A. A., Nasiri, B., & Meybodi, M. R. (2012). Some hybrid models to improve firefly algorithm performance. *International Journal of Artificial Intelligence*, *8*(S12), 97–117.

Farhang-Mehr, A., & Azarm, S. (2002). Diversity assessment of pareto optimal solution sets: an entropy approach. *Proceedings of the World on Congress on Computational Intelligence, Honolulu USA, May 2002*, (pp. 723–728), IEEE.

Farhoodnea, M., Mohamed, A., Shareef, H., & Zayandehroodi, H. (2014). Optimum placement of active power conditioners by a dynamic discrete firefly algorithm to mitigate the negative power quality effects of renewable energy-based generators. *International Journal of Electrical Power & Energy Systems*, *61*, 305–317.

Fateen, S.-E. K., Bonilla-Petriciolet, A., & Rangaiah, G. P. (2012). Evaluation of covariance matrix adaptation evolution strategy, shuffled complex evolution and firefly algorithms for phase stability, phase equilibrium and chemical equilibrium problems. *Chemical Engineering Research and Design*, *90*(12), 2051–2071.

Fister Jr, I., Yang, X.-S., Fister, I., & Brest, J. (2012). Memetic firefly algorithm for combinatorial optimization. *arXiv preprint arXiv:1204.5165*.

Fogel, D. B. (1995). A comparison of evolutionary programming and genetic algorithms on selected constrained optimization problems. *Simulation*, *64*(6), 397–404.

Fogel, L. J., Owens, A. J., & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution*. New York: Wiley Publishing.

Fraser, A. S. (1957). Simulation of genetic systems by automatic digital computers. *Australian Journal of Biological Science*, *10*(1), 484–491.

Frohlich, R. W. (2009). *Evolutionary intelligence*. Xlibris Corporation.

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2011). Mixed variable structural optimization using firefly algorithm. *Computers & Structures*, *89*(23), 2325–2336.

Gandomi, A. H., Yang, X.-S., & Alavi, A. H. (2013). Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems. *Engineering with Computers*, *29*(1), 17–35.

Gandomi, A. H., Yang, X.-S., Talatahari, S., & Alavi, A. H. (2013). Firefly algorithm with chaos. *Communications in Nonlinear Science and Numerical Simulation*, *18*(1), 89–98.

García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the cec'2005 special session on real parameter optimization. *Journal of Heuristics*, *15*(6), 617–644.

George, G., & Parthiban, L. (2015). Multi objective hybridized firefly algorithm with group search optimization for data clustering. *2015 IEEE International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN), Kolkata, India, Nov 2015*, (pp. 125–130), IEEE.

Ghalenoei, M., Hajimirsadeghi, H., & Lucas, C. (2009). Discrete invasive weed optimization algorithm: application to cooperative multiple task assignment of uavs. *Proceedings of the 48th IEEE Conference on Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009, December 2011, Shanghai*, (pp. 1665–1670), IEEE.

Ghasaq, A. L., Tokhi, M. O., & Ali, S. K. (2015). Design and control of exoskeleton for elderly mobility. *Assistive Robotics: Proceedings of the 18th International Conference on CLAWAR 2015*, (pp. 67–74), World Scientific.

Ghasemi, M., Ghavidel, S., Aghaei, J., Gitizadeh, M., & Falah, H. (2014). Application of chaos-based chaotic invasive weed optimization techniques for environmental opf problems in the power system. *Chaos, Solitons & Fractals*, *69*, 271–284.

Ghasemi, M., Ghavidel, S., Ghanbarian, M. M., & Habibi, A. (2014). A new hybrid algorithm for optimal reactive power dispatch problem with discrete and continuous control variables. *Applied Soft Computing*, *22*, 126–140.

Gholizadeh, S. (2015). Performance-based optimum seismic design of steel structures by a modified firefly algorithm and a new neural network. *Advances in Engineering Software*, *81*, 50–65.

Ghosh, A., Das, S., Chowdhury, A., & Giri, R. (2011). An ecologically inspired direct search method for solving optimal control problems with bézier parameterization. *Engineering Applications of Artificial Intelligence*, *24*(7), 1195–1203.

Giri, R., Chowdhury, A., Ghosh, A., Das, S., Abraham, A., & Snasel, Y. (2010a). Automatic shell clustering using a metaheuristic approach. *2010 IEEE International Conference on Systems Man and Cybernetics (SMC), 10-13 October 2010, Istanbul*, (pp. 2579–2586), IEEE.

Giri, R., Chowdhury, A., Ghosh, A., Das, S., Abraham, A., & Snasel, Y. (2010b). A modified invasive weed optimization algorithm for training of feed-forward neural networks.

2010 IEEE International Conference on Systems Man and Cybernetics (SMC), 10-13 October 2010, Istanbul, (pp. 3166–3173), IEEE.

Głowiński, S., Krzyżyński, T., Pecolt, S., & Maciejewski, I. (2015). Design of motion trajectory of an arm exoskeleton. *Archive of Applied Mechanics*, *85*(1), 75–87.

Gokhale, S. S., & Kale, V. S. (2016). An application of a tent map initiated chaotic firefly algorithm for optimal overcurrent relay coordination. *International Journal of Electrical Power & Energy Systems*, *78*, 336–342.

Guo, L., Wang, G.-G., Wang, H., & Wang, D. (2013). An effective hybrid firefly algorithm with harmony search for global numerical optimization. *The Scientific World Journal*, *2013*, 9 pages. doi: 10.1155/2013/125625

Gupta, S., & Arora, S. (2016). A hybrid firefly algorithm and social spider algorithm for multimodal function. *Intelligent Systems Technologies and Applications*, (pp. 17–30), Springer.

Haider, U., Das, S., Maity, D., Abraham, A., & Dasgupta, P. (2011). Self adaptive cluster based and weed inspired differential evolution algorithm for real world optimization. *2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, June 2011*, (pp. 750–756), IEEE.

Hansen, N., Auger, A., Finck, S., & Ros, R. (2010, 01). *Real-parameter black-box optimization benchmarking 2010: Experimental setup* (Tech. Rep. No. RR-7215). INRIA. Retrieved from http://coco.gforge.inria.fr/

Hassan, M. Y., & Karam, Z. A. (2015). Force-position controller design for 4dofs rehabilitation robot arm using pso algorithm. *International Journal of Information Technology and Business Management*, *39*(1), 84–98.

Hassanzadeh, T., & Meybodi, M. R. (2012a). A new hybrid algorithm based on firefly algorithm and cellular learning automata. *2012 20th Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, May 2012*, (pp. 628–633), IEEE.

Hassanzadeh, T., & Meybodi, M. R. (2012b). A new hybrid approach for data clustering using firefly algorithm and k-means. *2012 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), Shiraz, Fars, May 2012*, (pp. 007–011), IEEE.

He, Q., & Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. *Engineering Applications of Artificial Intelligence*, *20*(1), 89–99.

Holland, J. H. (1975). *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. U Michigan Press.

Homaifar, A., Qi, C. X., & Lai, S. H. (1994). Constrained optimization via genetic algorithms. *Simulation*, *62*(4), 242–253.

Horng, M.-H., & Liou, R.-J. (2011). Multilevel minimum cross entropy threshold selection based on the firefly algorithm. *Expert Systems with Applications*, *38*(12), 14805–14811.

Hu, Z., & Cai, X. (2012). Enhancing multi-objective invasive weed optimization with information exchange in intra-and inter-communities. *2012 IEEE Fifth International Conference on Advanced Computational Intelligence (ICACI), Nanjing, October 2012*, (pp. 360–364), IEEE.

Hu, Z., Cai, X., & Fan, Z. (2014). An improved memetic algorithm using ring neighborhood topology for constrained optimization. *Soft Computing*, *18*(10), 2023–2041.

Huang, F.-z., Wang, L., & He, Q. (2007). An effective co-evolutionary differential evolution for constrained optimization. *Applied Mathematics and Computation*, *186*(1), 340–356.

Hughes, E. J. (2001). Constraint handling with uncertain and noisy multi-objective evolution. *Proceedings of the 2001 Congress on Evolutionary Computation, Seoul, Korea, 2001*, *2*, (pp. 963–970), IEEE.

Hung, H.-L., Chao, C.-C., Cheng, C.-H., & Huang, Y.-F. (2010). Invasive weed optimization method based blind multiuser detection for mc-cdma interference suppression over multipath fading channel. *2010 IEEE International Conference on Systems Man and Cybernetics (SMC), 10-13 October 2010, Istanbul*, 2145–2150.

Jamil, M., & Yang, X.-S. (2013). A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, *4*(2), 150–194.

Jariyatantiwait, C., & Yen, G. G. (2014). Fuzzy multiobjective differential evolution using performance metrics feedback. *2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, July 2014*, (pp. 1959–1966), IEEE.

Jayabarathi, T., Yazdani, A., & Ramesh, V. (2012). Application of the invasive weed optimization algorithm to economic dispatch problems. *Frontiers in Energy*, *6*(3), 255–259.

Jordehi, A. R. (2015). A review on constraint handling strategies in particle swarm optimisation. *Neural Computing and Applications*, *26*(6), 1265–1275.

Kanimozhi, T., & Latha, K. (2015). An integrated approach to region based image retrieval using firefly algorithm and support vector machine. *Neurocomputing*, *151*, 1099–1111.

Kannan, B., & Kramer, S. N. (1994). An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. *Journal of Mechanical Design*, *116*(2), 405–411.

Karaboga, D. (2005). *An idea based on honey bee swarm for numerical optimization* (Tech. Rep.). Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.

Karaboga, D., & Akay, B. (2011). A modified artificial bee colony (abc) algorithm for constrained optimization problems. *Applied Soft Computing*, *11*(3), 3021–3031.

Karimkashi, S., & Kishk, A. A. (2010). Invasive weed optimization and its features in electromagnetics. *IEEE Transactions on Antennas and Propagation*, *58*(4), 1269–1278.

Kaveh, A., & Talatahari, S. (2009). Engineering optimization with hybrid particle swarm and ant colony optimization. *Asian Journal of Civil Engineering*, *10*(6), 611–628.

Kaveh, A., & Talatahari, S. (2010). An improved ant colony optimization for constrained engineering design problems. *Engineering Computations*, *27*(1), 155–182.

Kaveh, A., & Talatahari, S. (2011). Hybrid charged system search and particle swarm optimization for engineering design problems. *Engineering Computations*, *28*(4), 423–440.

Kazemzadeh-Parsi, M. J. (2014). A modified firefly algorithm for engineering design optimization problems. *Iranian Journal of Science and Technology*, *38*(M2), 403–421.

Khadwilard, A., Chansombat, S., Thepphakorn, T., Thapatsuwan, P., Chainate, W., & Pongcharoen, P. (2011). Application of firefly algorithm and its parameter setting for job shop scheduling. *First Symposium on Hands-On Research and Development*, *1*.

Khan, A. M., Ji, Y. H., Ali, M. A., Han, J. S., & Han, C. S. (2015). Passivity based adaptive control and its optimization for upper limb assist exoskeleton robot. *Journal of the Korean Society for Precision Engineering*, *32*(10), 857–863.

Kim, G., & Kim, K. (2016). A study on the adaptation of the firefly algorithm for the synchronization between multiple controllers in sdn environment. *2016 International*

Conference on Information Networking (ICOIN), Kota Kinabalu, Malaysia, 13-15 January 2016, (pp. 308–311), IEEE.

Kirtley, C. (2006). *Clinical gait analysis: theory and practice*. Elsevier Health Sciences.

Klausen, A., Tørdal, S. S., Karimi, H. R., Robbersmyr, K. G., Jecmenica, M., & Melteig, O. (2014). Firefly optimization and mathematical modeling of a vehicle crash test based on single-mass. *Journal of Applied Mathematics*, *2014*.

Knowles, J., & Corne, D. (1999). The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation. *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99, Washington DC, July 1999*, *1*.

Krawczyk, B. (2015). One-class classifier ensemble pruning and weighting with firefly algorithm. *Neurocomputing*, *150*, 490–500.

Kumar, K. V., & Phani, G. L. (2011). Combined economic emission dispatch-pareto optimal front approach. *International Journal of Computer Applications*, *30*(12), 16–21.

Kundu, D., Suresh, K., Ghosh, S., Das, S., Panigrahi, B. K., & Das, S. (2011). Multi-objective optimization with artificial weed colonies. *Information Sciences*, *181*(12), 2441–2454.

Kursawe, F. (1990). A variant of evolution strategies for vector optimization. *Parallel Problem Solving from Nature*, (pp. 193–197), Springer.

Lee, K. S., & Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer Methods in Applied Mechanics and Engineering*, *194*(36), 3902–3933.

Li, H., & Ye, C. (2012). Firefly algorithm on multi-objective optimization of production scheduling system. *Advances in Mechanical Engineering and its Applications*, *3*(1), 258–262.

Li, X., & Yin, M. (2014). Self-adaptive constrained artificial bee colony for constrained numerical optimization. *Neural Computing and Applications*, *24*(3-4), 723–734.

Li, Y., Liu, Y., & Huang, P. (2015). An modified hybrid invasive weed optimization for array antenna beamforming. *2015 IEEE International Conference on Mechatronics and Automation (ICMA), Beijing, China, August 2015*, (pp. 2358–2362), IEEE.

Liang, J., Qu, B., & Suganthan, P. (2013, 11). *Problem definitions and evaluation criteria for the cec 2014 special session and competition on single objective real-parameter numerical optimization* (Tech. Rep. No. Technical Report 201311). Computational Intelligence Laboratory, Zhengzhou University, China and Nanyang Technological University, Singapore.

Liang, J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P., Coello, C. C., & Deb, K. (2006). Problem definitions and evaluation criteria for the cec 2006 special session on constrained real-parameter optimization. *Journal of Applied Mechanics*, *41*, 8.

Liang, R.-H., Wang, J.-C., Chen, Y.-T., & Tseng, W.-T. (2015). An enhanced firefly algorithm to multi-objective optimal reactive power dispatch with uncertainties consideration. *International Journal of Electrical Power & Energy Systems*, *64*, 1088–1097.

Lin, C.-H., Lien, W.-M., Wang, W.-W., Chen, S.-H., Lo, C.-H., Lin, S.-Y., & Lai, J.-S. (2014). Ntuh-ii robot arm with dynamic torque gain adjustment method for frozen shoulder rehabilitation. *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2014), Chicago IL, September 2014*, (pp. 3555–3560), IEEE.

Lin, M.-H., Tsai, J.-F., Hu, N.-Z., & Chang, S.-C. (2013). Design optimization of a speed reducer using deterministic techniques. *Mathematical Problems in Engineering*, *2013*.

Liu, C., Gao, Z., & Zhao, W. (2012). A new path planning method based on firefly algorithm. *2012 Fifth International Joint Conference on Computational Sciences and Optimization (CSO), Harbin, June 2012*, (pp. 775–778), IEEE.

Liu, F., Cheng, W. M., & Zhao, N. (2012, 9). Optimal design of lower extremity for portable human exoskeletons using improved particle swarm optimization. *Materials Processing Technology II*, *538*, (pp. 3215–3221), Trans Tech Publications. doi: 10.4028/www.scientific.net/AMR.538-541.3215

Liu, H., Cai, Z., & Wang, Y. (2010). Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. *Applied Soft Computing*, *10*(2), 629–640.

Liu, J., Teo, K., Wang, X., & Wu, C. (2015). An exact penalty function-based differential search algorithm for constrained global optimization. *Soft Computing*, *20*(4), 1305–1313.

Liu, R., Wang, X., Li, Y., & Zhang, X. (2012). Multi-objective invasive weed optimization algortihm for clustering. *2012 IEEE Congress on Evolutionary Computation (CEC), Brisbane, QLD, June 2012*, (pp. 1–8), IEEE.

Ljung, L. (1987). System identification: theoryfor the user. *Englewood Cliffs*.

Long, Y., Du, Z.-j., Wang, W.-d., & Dong, W. (2016). Robust sliding mode control based on ga optimization and cmac compensation for lower limb exoskeleton. *Applied Bionics and Biomechanics*, *2016*.

Low, K. (2011). Robot-assisted gait rehabilitation: From exoskeletons to gait systems. *2011 Defense Science Research Conference and Expo (DSR), August 2011, Singapore*, (pp. 1–10), IEEE.

Łukasik, S., & Żak, S. (2009). Firefly algorithm for continuous constrained optimization tasks. *Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, (pp. 97–106), Springer.

Ma, Y., Zhao, Y., Wu, L., He, Y., & Yang, X.-S. (2015). Navigability analysis of magnetic map with projecting pursuit-based selection method by using firefly algorithm. *Neurocomputing*, *159*, 288–297.

Maghsoudlou, H., Afshar-Nadjafi, B., & Niaki, S. T. A. (2016). A multi-objective invasive weeds optimization algorithm for solving multi-skill multi-mode resource constrained project scheduling problem. *Computers & Chemical Engineering*, *88*, 157–169.

Mahapatra, S., Panda, S., & Swain, S. C. (2014). A hybrid firefly algorithm and pattern search technique for sssc based power oscillation damping controller design. *Ain Shams Engineering Journal*, *5*(4), 1177–1188.

Mahdavi, M., Fesanghary, M., & Damangir, E. (2007). An improved harmony search algorithm for solving optimization problems. *Applied Mathematics and Computation*, *188*(2), 1567–1579.

Mahto, S. K., & Choubey, A. (2016). A novel hybrid iwo/wdo algorithm for nulling pattern synthesis of uniformly spaced linear and non-uniform circular array antenna. *AEU-International Journal of Electronics and Communications*, *70*(6), 750–756.

Marichelvam, M. K., Prabaharan, T., & Yang, X.-S. (2014). A discrete firefly algorithm for the multi-objective hybrid flowshop scheduling problems. *IEEE Transactions on Evolutionary Computation*, *18*(2), 301–305.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*(6), 369–395.

Marler, R. T., & Arora, J. S. (2010). The weighted sum method for multi-objective optimization: new insights. *Structural and Multidisciplinary Optimization*, *41*(6), 853–862.

Mazhoud, I., Hadj-Hamou, K., Bigeon, J., & Joyeux, P. (2013). Particle swarm optimization for solving engineering problems: a new constraint-handling mechanism. *Engineering Applications of Artificial Intelligence*, *26*(4), 1263–1273.

Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. *Ecological Informatics*, *1*(4), 355–366.

Mehrabian, A. R., & Yousefi-Koma, A. (2007). Optimal positioning of piezoelectric actuators on a smart fin using bio-inspired algorithms. *Aerospace Science and Technology*, *11*(2), 174–182.

Mezura-Montes, E., & Coello, C. A. C. (2005). Useful infeasible solutions in engineering optimization with evolutionary algorithms. *MICAI 2005: Advances in Artificial Intelligence*, (pp. 652–662), Springer.

Mezura Montes, E., Coello Coello, C. A., & Landa-Becerra, R. (2003). Engineering optimization using simple evolutionary algorithm. *15th IEEE International Conference on Tools with Artificial Intelligence*, (pp. 149–156), IEEE.

Mezura-Montes, E., Miranda-Varela, M. E., & del Carmen Gómez-Ramón, R. (2010). Differential evolution in constrained numerical optimization: an empirical study. *Information Sciences*, *180*(22), 4223–4262.

Miranda-Linares, D., Alrezage, G., & Tokhi, M. O. (2015). Control of lower limb exoskeleton for elderly assistance on basic mobility tasks. *19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, October 2015*, (pp. 441–446), IEEE.

Mirjalili, S., & Lewis, A. (2015). Novel performance metrics for robust multi-objective optimization algorithms. *Swarm and Evolutionary Computation*, *21*, 1–23.

Mishra, A., Agarwal, C., Sharma, A., & Bedi, P. (2014). Optimized gray-scale image watermarking using dwt–svd and firefly algorithm. *Expert Systems with Applications*, *41*(17), 7858–7867.

Mohammadi, S., Mozafari, B., Solimani, S., & Niknam, T. (2013). An adaptive modified firefly optimisation algorithm based on hong's point estimate method to optimal operation management in a microgrid with consideration of uncertainties. *Energy*, *51*, 339–348.

Moubarak, S., Pham, M. T., Pajdla, T., & Redarce, T. (2009). Design and modeling of an upper extremity exoskeleton. *World Congress on Medical Physics and Biomedical Engineering, Munich, Germany, September 7-12, 2009*, (pp. 476–479), Springer.

Mun, S., & Cho, Y.-H. (2012). Modified harmony search optimization for constrained design problems. *Expert Systems with Applications*, *39*(1), 419–423.

Naidu, K., Mokhlis, H., Bakar, A., Terzija, V., & Illias, H. A. (2014). Application of firefly algorithm with online wavelet filter in automatic generation control of an interconnected reheat thermal power system. *International Journal of Electrical Power & Energy Systems*, *63*, 401–413.

Naidu, Y. R., & Ojha, A. K. (2014). Solving nonlinear constrained optimization problems by invasive weed optimization using penalty function. *2014 IEEE International Advance Computing Conference (IACC), Gurgaon, February 2014*, (pp. 1326–1330), IEEE.

Naidu, Y. R., & Ojha, A. K. (2015a). A hybrid version of invasive weed optimization with quadratic approximation. *Soft Computing*, *19*(12), 3581–3598.

Naidu, Y. R., & Ojha, A. K. (2015b). Solving nonlinear constrained optimization problems using invasive weed optimization. *Intelligent Computing, Communication and Devices*, (pp. 127–133), Springer.

Nasir, A., & Tokhi, M. O. (2014). A novel hybrid bacteria-chemotaxis spiral-dynamic algorithm with application to modelling of flexible systems. *Engineering Applications of Artificial Intelligence*, *33*, 31–46.

Nasir, A., & Tokhi, M. O. (2015). Novel metaheuristic hybrid spiral-dynamic bacteria-chemotaxis algorithms for global optimisation. *Applied Soft Computing*, *27*, 357–375.

Nasir, A. N. K., Ismail, R. R., & Tokhi, M. O. (2016). Adaptive spiral dynamics metaheuristic algorithm for global optimization with application to modelling of a flexible system. *Applied Mathematical Modelling*, *40*(9–10), 5442–5461.

Nasir, A. N. K., Tokhi, M. O., Ghani, N. M. A., & Ahmad, M. A. (2012). A novel hybrid spiral dynamics bacterial chemotaxis algorithm for global optimization with application to controller design. *2012 UKACC International Conference on Control (CONTROL), Cardiff, Wales, September 2012*, (pp. 753–758), IEEE.

Nayak, M. R., Krishnanand, K. R., & Rout, P. K. (2011). Optimal reactive power dispatch based on adaptive invasive weed optimization algorithm. *2011 International Conference on Energy, Automation, and Signal (ICEAS), December 2011, Bhubaneswar, Odisha*, (pp. 1–7), IEEE.

Nef, T., Guidali, M., & Riener, R. (2009). Armin iii–arm therapy exoskeleton with an ergonomic shoulder actuation. *Applied Bionics and Biomechanics*, *6*(2), 127–142.

Nekouie, N., & Yaghoobi, M. (2016). A new method in multimodal optimization based on firefly algorithm. *Artificial Intelligence Review*, 1–21.

Niknam, T., Azizipanah-Abarghooee, R., & Roosta, A. (2012). Reserve constrained dynamic economic dispatch: a new fast self-adaptive modified firefly algorithm. *Systems Journal, IEEE*, *6*(4), 635–646.

Niknam, T., Azizipanah-Abarghooee, R., Roosta, A., & Amiri, B. (2012). A new multiobjective reserve constrained combined heat and power dynamic economic emission dispatch. *Energy*, *42*(1), 530–545.

Nikoofard, A. H., Hajimirsadeghi, H., Rahimi-Kian, A., & Lucas, C. (2012). Multiobjective invasive weed optimization: Application to analysis of pareto improvement models in electricity markets. *Applied Soft Computing*, *12*(1), 100–112.

Niu, B., Wang, J., & Wang, H. (2015). Bacterial-inspired algorithms for solving constrained optimization problems. *Neurocomputing*, *148*, 54–62.

Nunes de Castro, L. (2012). *Nature-inspired computing design, development, and applications*. IGI Global.

Ojha, A. K., & Naidu, Y. R. (2015). A hybrid cat swarm optimization with invasive weed optimization. *2015 International Conference on Electrical, Electronics, Signals, Communication and Optimization (EESCO), Visakhapatnam, January 2015*, (pp. 1–3), IEEE.

Olamaei, J., Moradi, M., & Kaboodi, T. (2013). A new adaptive modified firefly algorithm to solve optimal capacitor placement problem. *2013 18th Conference on Electrical Power Distribution Networks (EPDC), Kermanshah, May 2013*, (pp. 1–6), IEEE.

Opara, K., & Arabas, J. (2011). Benchmarking procedures for continuous optimization algorithms. *Journal of Telecommunications and Information Technology*, 73–80.

Ouyang, A., Peng, X., Wang, Q., Wang, Y., & Truong, T. K. (2016). A parallel improved iwo algorithm on gpu for solving large scale global optimization problems. *Journal of Intelligent & Fuzzy Systems*(Preprint), 1–11. doi: 10.3233/JIFS-169033

Ozsoydan, F. B., & Baykasoglu, A. (2015). A multi-population firefly algorithm for dynamic optimization problems. , (pp. 1–7), IEEE.

Pahlavani, P., Delavar, M. R., & Frank, A. U. (2012). Using a modified invasive weed optimization algorithm for a personalized urban multi-criteria path optimization problem. *International Journal of Applied Earth Observation and Geoinformation*, *18*, 313–328.

Pal, S., Basak, A., & Das, S. (2009). Constrained real parameter optimization with an ecologically inspired algorithm. *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, December 2009*, (pp. 1270–1275), IEEE.

Parsopoulos, K. E., & Vrahatis, M. N. (2002). Particle swarm optimization method in multiobjective problems. *Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, Spain, March 2002*, (pp. 603–607), ACM.

Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Systems*, *22*(3), 52–67.

Peng, S., Ouyang, A.-J., & Zhang, J. J. (2015). An adaptive invasive weed optimization algorithm. *International Journal of Pattern Recognition and Artificial Intelligence*, *29*(02), 1559004.

Poerwanto, H. (1998). *Dynamic simulation and control of flexible manipulator systems* (Unpublished doctoral dissertation). Department of Automatic Control and Systems Engineering, The University of Sheffield, UK.

Pourjafari, E., & Mojallali, H. (2012). Solving nonlinear equations systems with a new approach based on invasive weed optimization algorithm and clustering. *Swarm and Evolutionary Computation*, *4*, 33–43.

Pouya, A. R., Solimanpur, M., & Rezaee, M. J. (2016). Solving multi-objective portfolio optimization problem using invasive weed optimization. *Swarm and Evolutionary Computation*, *28*.

Rad, H. S., & Lucas, C. (2007). A recommender system based on invasive weed optimization algorithm. *2007 IEEE Congress on Evolutionary Computation (CEC), Singapore, September 2007*, (pp. 4297–4304), IEEE.

Rahmani, A., & MirHassani, S. A. (2014). A hybrid firefly-genetic algorithm for the capacitated facility location problem. *Information Sciences*, *283*, 70–78.

Rajan, A., & Malakar, T. (2015). Optimal reactive power dispatch using hybrid nelder–mead simplex based firefly algorithm. *International Journal of Electrical Power & Energy Systems*, *66*, 9–24.

Rakshit, P., Das, P., Konar, A., Nasipuri, M., & Janarthanan, R. (2012). A recurrent fuzzy neural model of a gene regulatory network for knowledge extraction using invasive weed and artificial bee colony optimization algorithm. *2012 1st International Conference on Recent Advances in Information Technology (RAIT)*, (pp. 385–391), IEEE.

Rangaiah, G. P. (2009). *Multi-objective optimization: techniques and applications in chemical engineering* (Vol. 1). World Scientific.

Rao, S. S., & Rao, S. S. (2009). *Engineering optimization: theory and practice*. John Wiley & Sons.

Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). Gsa: a gravitational search algorithm. *Information Sciences*, *179*(13), 2232–2248.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*. Frommann-Holzboog, Stuttgart.

Reddy, N. S., Saketh, M. S., Pal, P., & Dey, R. (2016). Optimal pid controller design of an inverted pendulum dynamics: A hybrid pole-placement & firefly algorithm approach. *2016 IEEE First International Conference on Control, Measurement and Instrumentation (CMI)*, (pp. 305–310), IEEE.

Rizk-Allah, R., Zaki, E. M., & El-Sawy, A. A. (2013). Hybridizing ant colony optimization with firefly algorithm for unconstrained optimization problems. *Applied Mathematics and Computation*, *224*, 473–483.

Roeva, O., & Slavov, T. (2012). Firefly algorithm tuning of pid controller for glucose concentration control during e. coli fed-batch cultivation process. *Proceedings of the Federated Conference on Computer Science and Information Systems (FedCSIS), Wroclaw, September 2012*, (pp. 455–462).

Roshanaei, M., Lucas, C., & Mehrabian, A. R. (2009). Adaptive beamforming using a novel numerical optimisation algorithm. *Microwaves, Antennas & Propagation, IET*, *3*(5), 765–773.

Roy, G. G., Chakraborty, P., & Das, S. (2010). Designing fractional-order piλdμ controller using differential harmony search algorithm. *International Journal of Bio-Inspired Computation*, *2*(5), 303–309.

Roy, G. G., Chakraborty, P., Zhao, S.-Z., Das, S., & Suganthan, P. N. (2010). Artificial foraging weeds for global numerical optimization over continuous spaces. *IEEE Congress on Evolutionary Computation (CEC), Barcelona, Spain, July 2010*, (pp. 1–8), IEEE.

Roy, G. G., Das, S., Chakraborty, P., & Suganthan, P. N. (2011). Design of non-uniform circular antenna arrays using a modified invasive weed optimization algorithm. *IEEE Transactions on Antennas and Propagation*, *59*(1), 110–118.

Roy, S., Islam, S. M., Das, S., & Ghosh, S. (2013). Multimodal optimization by artificial weed colonies enhanced with localized group search optimizers. *Applied Soft Computing*, *13*(1), 27–46.

Rubio-Largo, Á., Vega-Rodríguez, M. A., & González-Álvarez, D. L. (2014). An improved multiobjective approach inspired by the flashing behaviour of fireflies for traffic grooming in optical wdm networks. *Applied Soft Computing*, *21*, 617–636.

Rudolph, G. (1998). On a multi-objective evolutionary algorithm and its convergence to the pareto set. *The 1998 IEEE International Conference on Evolutionary Computation Proceedings, 1998. IEEE World Congress on Computational Intelligence., Anchorage, AK, May 1998*, (pp. 511–516), IEEE.

Sadollah, A., Bahreininejad, A., Eskandar, H., & Hamdi, M. (2013). Mine blast algorithm: A new population based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, *13*(5), 2592–2612.

Sahu, R. K., Panda, S., & Pradhan, P. C. (2015). Design and analysis of hybrid firefly algorithm-pattern search based fuzzy pid controller for lfc of multi area power systems. *International Journal of Electrical Power & Energy Systems*, *69*, 200–212.

Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. *Journal of Mechanical Design*, *112*(2), 223–229.

Saravanan, B., Vasudevan, E. R., & Kothari, D. P. (2014). Unit commitment problem solution using invasive weed optimization algorithm. *International Journal of Electrical Power & Energy Systems*, *55*, 21–28.

Sayadi, M., Ramezanian, R., & Ghaffari-Nasab, N. (2010). A discrete firefly meta-heuristic with local search for makespan minimization in permutation flow shop scheduling problems. *International Journal of Industrial Engineering Computations*, *1*(1), 1–10.

Sayadi, M. K., Hafezalkotob, A., & Naini, S. G. J. (2013). Firefly-inspired algorithm for discrete optimization problems: An application to manufacturing cell formation. *Journal of Manufacturing Systems*, *32*(1), 78–84.

Schott, J. R. (1995). *Fault tolerant design using single and multicriteria genetic algorithm optimization.* (Tech. Rep.). DTIC Document.

Sengupta, A., Chakraborti, T., Konar, A., & Nagar, A. (2011). Energy efficient trajectory planning by a robot arm using invasive weed optimization technique. *2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC), October 2011, Salamanca*, (pp. 311–316), IEEE.

Sengupta, A., Chakraborti, T., Konar, A., & Nagar, A. K. (2012). An intelligent invasive weed optimization: a q-learning approach. *Proceedings of the International Conference on Genetic and Evolutionary Methods (GEM)*, (pp. 1), The Steering Committee of The

World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).

Setiadi, H., & Jones, K. O. (2016). Power system design using firefly algorithm for dynamic stability enhancement. *Indonesian Journal of Electrical Engineering and Computer Science*, *1*(3), 446–455.

Shafaati, M., & Mojallali, H. (2012). Modified firefly optimization for iir system identification. *Journal of Control Engineering and Applied Informatics*, *14*(4), 59–69.

Shah-Hosseini, H. (2011). Principal components analysis by the galaxy-based search algorithm: a novel metaheuristic for continuous optimisation. *International Journal of Computational Science and Engineering*, *6*(1-2), 132–140.

Shareef, H., Ibrahim, A. A., Salman, N., Mohamed, A., & Ai, W. L. (2014). Power quality and reliability enhancement in distribution systems via optimum network reconfiguration by using quantum firefly algorithm. *International Journal of Electrical Power & Energy Systems*, *58*, 160–169.

Shi, Y., Hou, L., & Zheng, X. (2015). Solving workshop layout by hybridizing invasive weed optimization with simulated annealing. *2015 IEEE 19th International Conference on Computer Supported Cooperative Work in Design (CSCWD), Calabria, May 2015*, (pp. 484–488), IEEE.

Siddique, M. N. H., & Tokhi, M. O. (2002). Ga-based neuro-fuzzy controller for flexible-link manipulator. *Proceedings of the 2002 International Conference on Control Applications*, *1*, (pp. 471–476), IEEE.

Simon, D. (2013). *Evolutionary optimization algorithms*. John Wiley & Sons.

Singh, D., & Shukla, R. S. (2016). Optimisation of electrochemical micromachining and electrochemical discharge machining process parameters using firefly algorithm. *International Journal of Mechatronics and Manufacturing Systems*, *9*(2), 137–159.

Su, K., Ma, L., Guo, X., & Sun, Y. (2014). An efficient discrete invasive weed optimization algorithm for web services selection. *Journal of Software*, *9*(3), 709–715.

Sundaram, K. M., Kumar, R. S., Krishnakumar, C., & Sugavanam, K. R. (2016). Fuzzy logic and firefly algorithm based hybrid system for energy efficient operation of three phase induction motor drives. *Indian Journal of Science and Technology*, *9*(1).

Sundari, M. G., Rajaram, M., & Balaraman, S. (2016). Application of improved firefly algorithm for programmed pwm in multilevel inverter with adjustable dc sources. *Applied Soft Computing*, *41*, 169–179.

Supriyono, H., Tokhi, M. O., & Zain, B. (2010). Adaptive biologically-inspired algorithm-based controller tuning for input tracking control of flexible manipulators. *2010 IEEE 9th International Conference on Cybernetic Intelligent Systems (CIS), Reading UK, September 2010*, (pp. 1–6), IEEE.

Suresh, K., Kundu, D., Ghosh, S., Das, S., & Abraham, A. (2009). Iwo with increased deviation and stochastic selection (iwo-id-ss) for global optimization of noisy fitness functions. *World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), Coimbatore, December 2009*, (pp. 215–220), IEEE.

Surjanovic, S., & Bingham, D. (2013). Virtual library of simulation experiments: test functions and datasets. *Simon Fraser University, in: http://www. sfu. ca/˜ ssurjano/optimization. html.*

Talatahari, S., Gandomi, A. H., & Yun, G. J. (2014). Optimum design of tower structures using firefly algorithm. *The Structural Design of Tall and Special Buildings*, *23*(5), 350–361.

Tamura, K., & Yasuda, K. (2011). Spiral dynamics inspired optimization. *Journal of Advanced Computational Intelligence and Intelligent Informatics, JACIII*, *15*(8), 1116–

1122.
Tan, K. C., Lee, T. H., & Khor, E. F. (2002). Evolutionary algorithms for multi-objective optimization: performance assessments and comparisons. *Artificial intelligence review*, *17*(4), 251–290.
Tang, W. J., & Wu, Q. H. (2009). Biologically inspired optimization: a review. *Transactions of the Institute of Measurement and Control*, *31*(6), 495–515.
Tian, Y., Gao, W., & Yan, S. (2012). An improved inertia weight firefly optimization algorithm and application. *Proceedings of the 2012 International Conference on Control Engineering and Communication Technology, Liaoning, December 2012*, (pp. 64–68), IEEE Computer Society.
Tilahun, S. L., & Ong, H. C. (2012). Modified firefly algorithm. *Journal of Applied Mathematics*, *2012*.
Toha, S. F., Julai, S., & Tokhi, M. O. (2012). Ant colony based model prediction of a twin rotor system. *Procedia Engineering*, *41*, 1135–1144.
Toha, S. F., & Tokhi, M. O. (2009). Dynamic nonlinear inverse-model based control of a twin rotor system using adaptive neuro-fuzzy inference system. *Third UKSim European Symposium on Computer Modeling and Simulation, 2009. EMS'09, Athens, November 2009*, (pp. 107–111), IEEE.
Toha, S. F., & Tokhi, M. O. (2010). Parametric modelling application to a twin rotor system using recursive least squares, genetic, and swarm optimization techniques. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *224*(9), 961–977.
Tuba, M., & Bacanin, N. (2014a). Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio selection problem. *Applied Mathematics & Information Sciences*, *8*(6), 2831–2844.
Tuba, M., & Bacanin, N. (2014b). Improved seeker optimization algorithm hybridized with firefly algorithm for constrained optimization problems. *Neurocomputing*, *143*, 197–207.
Twin rotor mimo system manual [Computer software manual]. (1996). Sussex, UK.
Veenhuis, C. (2010). Binary invasive weed optimization. *2010 Second World Congress on Nature and Biologically Inspired Computing (NaBIC), December 2010, Fukuoka*, (pp. 449–454), IEEE.
Virk, G. S., Haider, U., Indrawibawa, I. N., Thekkeparampumadom, R. K., & Masud, N. (2014). Exo-legs for elderly persons. *17th International Conference on Climbing and Walking Robots (CLAWAR), Poznan, POLAND, 21-23 July, 2014*, (pp. 85–92), World Scientific.
Vishwakarma, A. Y. B., & Yerpude, A. (2014). A meta-heuristic approach for image segmentation using firefly algorithm. *International Journal Of Computer Trends And Technology (IJCTT)*, *11*(2), 69–73.
Wagan, A. I., Shaikh, M. M., Abro, R., et al. (2015). Wind turbine micrositing by using the firefly algorithm. *Applied Soft Computing*, *27*, 450–456.
Wang, H., Cui, Z., Sun, H., Rahnamayan, S., & Yang, X.-S. (2016). Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism. *Soft Computing*, 1–15.
Wang, H., Wang, W., Sun, H., & Rahnamayan, S. (2016). Firefly algorithm with random attraction. *International Journal of Bio-Inspired Computation*, *8*(1), 33–41.
Wang, H., Zhou, X., Sun, H., Yu, X., Zhao, J., Zhang, H., & Cui, L. (2016). Firefly algorithm with adaptive control parameters. *Soft Computing*, 1–12.
Wang, S.-L., Song, B.-W., & Duan, G.-L. (2015). Multi-objective nondominated sorting in-

vasive weed optimization algorithm for the permanent magnet brushless direct current motor design. In *Genetic and evolutionary computing* (pp. 79–87). Springer.

Wang, S.-L. S. (2001). Motion simulation with working model 2d and msc. visualnastran 4d. *Journal of Computing and Information Science Engineering*, *1*(2), 193–196.

Winter, D. A. (2009). *Biomechanics and motor control of human movement*. John Wiley & Sons.

Woźniak, M., Gabryel, M., Nowicki, R. K., & Nowak, B. A. (2016). An application of firefly algorithm to position traffic in nosql database systems. *Knowledge, Information and Creativity Support Systems*, (pp. 259–272), Springer.

Wu, J., Wang, Y., Huang, J., & Xing, K. (2012). Artificial bee colony algorithm based auto-disturbance rejection control for rehabilitation robotic arm driven by pm-ts actuator. *2012 Proceedings of International Conference on Modelling, Identification & Control (ICMIC), Wuhan, China, June 2012*, (pp. 802–807), IEEE.

Xiong, T., Bao, Y., & Hu, Z. (2014). Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, *55*, 87–100.

Yan, X., Zhu, Y., Wu, J., & Chen, H. (2012). An improved firefly algorithm with adaptive strategies. *Advanced Science Letters*, *16*(1), 249–254.

Yang, X.-S. (2009). Firefly algorithms for multimodal optimization. *Stochastic Algorithms: Foundations and Applications*, (pp. 169–178), Springer.

Yang, X.-S. (2010a). *Engineering optimization: an introduction with metaheuristic applications*. John Wiley & Sons.

Yang, X.-S. (2010b). Firefly algorithm, levy flights and global optimization. *Research and Development in Intelligent Systems XXVI*, (pp. 209–218), Springer.

Yang, X.-S. (2010c). Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, *2*(2), 78–84.

Yang, X.-S. (2010d). *Nature-inspired metaheuristic algorithms*. Luniver Press.

Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, *3*(5), 267–274.

Yang, X.-S. (2013). Multiobjective firefly algorithm for continuous optimization. *Engineering with Computers*, *29*(2), 175–184.

Yang, X.-S., & Deb, S. (2014). Cuckoo search: recent advances and applications. *Neural Computing and Applications*, *9*, 169–174.

Yang, X.-S., & He, X. (2013). Firefly algorithm: recent advances and applications. *International Journal of Swarm Intelligence*, *1*(1), 36–50.

Yang, X.-S., Hosseini, S. S. S., & Gandomi, A. H. (2012). Firefly algorithm for solving non-convex economic dispatch problems with valve loading effect. *Applied Soft Computing*, *12*(3), 1180–1186.

Yang, X.-S., Karamanoglu, M., & He, X. (2013). Multi-objective flower algorithm for optimization. *Procedia Computer Science*, *18*, 861–868.

Yatim, H. M., & Mat Darus, I. Z. (2014). Self-tuning active vibration controller using particle swarm optimization for flexible manipulator system. *WSEAS Transactions on Systems and Control*, *39*(1), 55–66.

Yılmaz, S., & Küçüksille, E. U. (2015). A new modification approach on bat algorithm for solving optimization problems. *Applied Soft Computing*, *28*, 259–275.

Yin, Z., Wen, M., & Ye, C. (2012). Improved invasive weed optimization based on hybrid genetic algorithm. *Journal of Computational Information Systems*, *8*(8), 3437–3444.

Younes, M., Khodja, F., & Kherfane, R. L. (2014). Multi-objective economic emission dispatch solution using hybrid ffa (firefly algorithm) and considering wind power penetration. *Energy*, *67*, 595–606.

Yu, K., Wang, X., & Wang, Z. (2014). An improved teaching-learning-based optimization algorithm for numerical and engineering optimization problems. *Journal of Intelligent Manufacturing*, 1–13.

Yu, S., Su, S., Lu, Q., & Huang, L. (2014). A novel wise step strategy for firefly algorithm. *International Journal of Computer Mathematics*, *91*(12), 2507–2513.

Zadeh, L. (1963). Optimality and non-scalar-valued performance criteria. *IEEE Transactions on Automatic Control*, *8*(1), 59–60.

Zaharis, Z. D., Skeberis, C., Xenos, T. D., Lazaridis, P. I., & Cosmas, J. (2013). Design of a novel antenna array beamformer using neural networks trained by modified adaptive dispersion invasive weed optimization based data. *IEEE Transactions on Broadcasting*, *59*(3), 455–460.

Zhang, W., Yen, G. G., & He, Z. (2014). Constrained optimization via artificial immune system. *IEEE Transactions on Cybernetics*, *44*(2), 185–198.

Zhang, X., Wang, Y., Cui, G., Niu, Y., & Xu, J. (2009). Application of a novel iwo to the design of encoding sequences for dna computing. *Computers & Mathematics with Applications*, *57*(11), 2001–2008.

Zhang, X., Xu, J., Cui, G., Wang, Y., & Niu, Y. (2008). Research on invasive weed optimization based on the cultural framework. *3rd International Conference on Bio-Inspired Computing: Theories and Applications (BICTA 2008), Adelaide, SA, October 2008*, (pp. 129–134), IEEE.

Zhao, H., Wang, P.-h., Peng, X., Qian, J., & Wang, Q. (2009). Constrained optimization of combustion at a coal-fired utility boiler using hybrid particle swarm optimization with invasive weed. *International Conference on Energy and Environment Technology (ICEET'09), Guilin, Guangxi, October 2009*, *1*, (pp. 564–567), IEEE.

Zhou, Y., Chen, H., & Zhou, G. (2014). Invasive weed optimization algorithm for optimization no-idle flow shop scheduling problem. *Neurocomputing*, *137*, 285–292.

Zhou, Y., Luo, Q., & Chen, H. (2013). A novel differential evolution invasive weed optimization algorithm for solving nonlinear equations systems. *Journal of Applied Mathematics*, *2013, Article ID 757391*, 18 pages. doi: 10.1155/2013/757391

Zhou, Y., Luo, q., Chen, H., He, A., & Wu, J. (2014). A discrete invasive weed optimization algorithm for solving traveling salesman problem. *Neurocomputing*, *151*(3), 1227–1236.

Zhou, Y., Zhou, G., & Zhang, J. (2013). A hybrid glowworm swarm optimization algorithm for constrained engineering design problems. *International Journal of Applied Mathematics & Information Sciences*, *7*(1), 379–388.

Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications* (Vol. 63). Citeseer.

Zitzler, E., Deb, K., & Thiele, L. (2000). Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation*, *8*(2), 173–195.

Appendix A

Flow-charts of The Algorithms

A1. Firefly Optimisation Algorithm

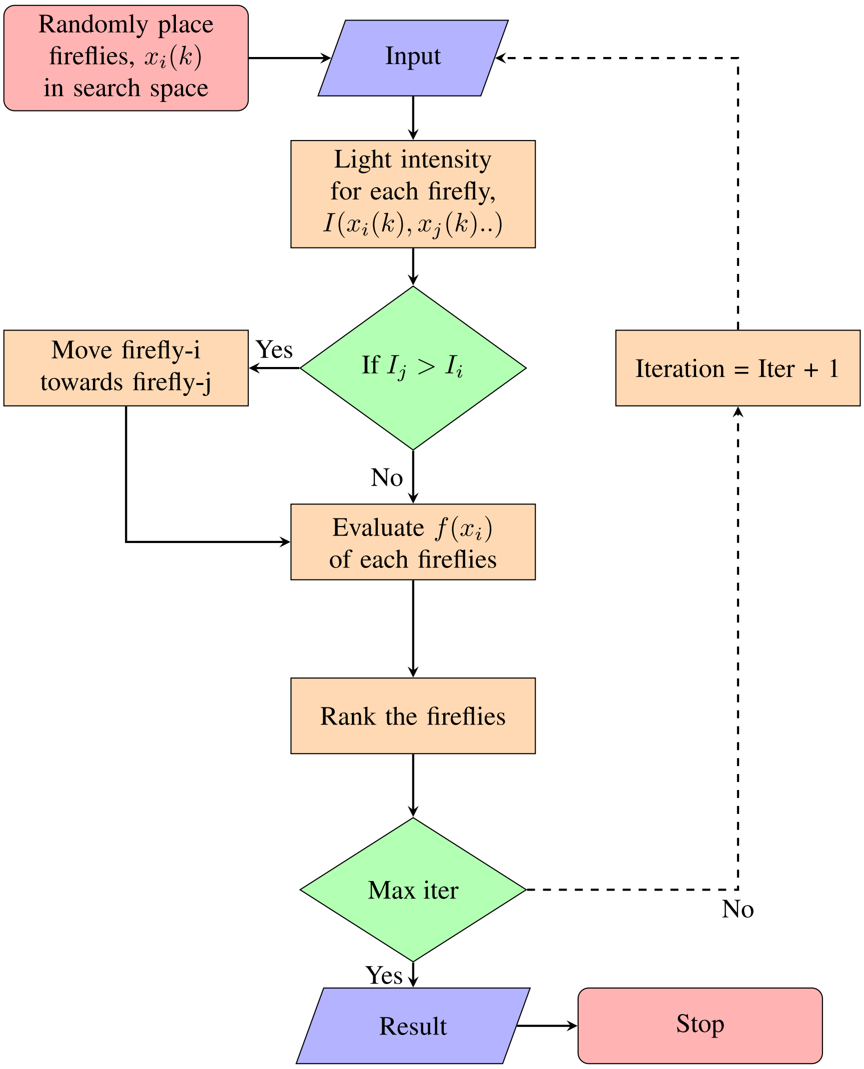


Figure A.1: Flow-chart of firefly algorithm

A2. Invasive Weed Optimisation Algorithm

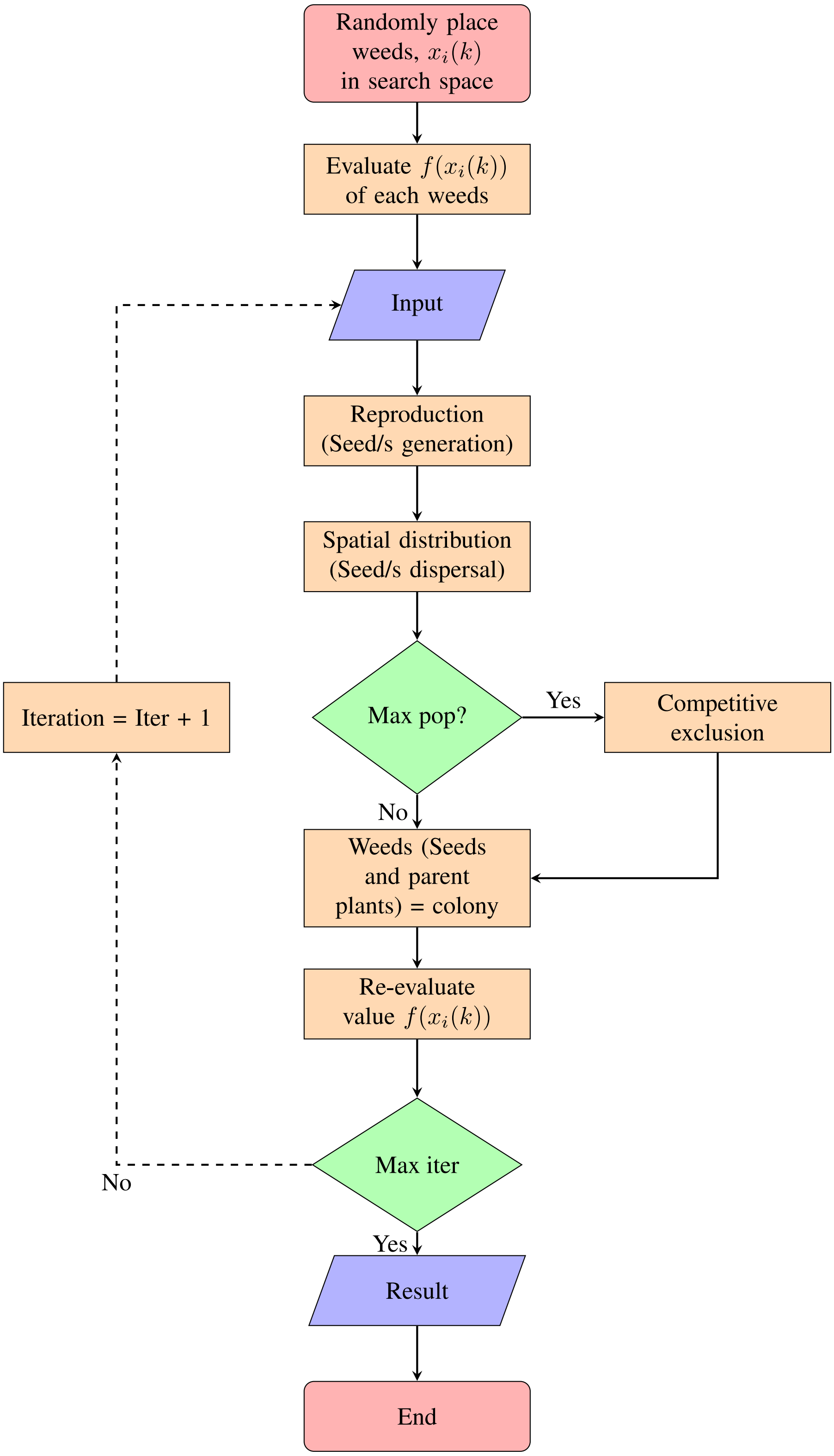


Figure A.2: Flow-chart of invasive weed optimisation algorithm

Appendix B

Benchmark Functions

Details of the benchmark problems utilized in the thesis are as follows:

B1 CEC2014 Benchmark Functions

All test functions are minimisation problems and each function has a shift data and is scalable. For convenience, the search ranges are defined for all test functions as:

$$Search\ range : [-100, 100]^D$$

Problem 1 - High conditioned elliptic function

$$f_{11}(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2 \tag{B.1}$$

Problem 2 - Bent cigar function

$$f_{12}(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2 \tag{B.2}$$

Problem 3 - Discus function

$$f_{13}(x) = 10^6 x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2 \tag{B.3}$$

Problem 4 - Rosenbrock's function

$$f_{14}(x) = \sum_{i=1}^{D-1} [100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2] \tag{B.4}$$

Problem 5 - Ackley's function

$$f_{15}(x) = -20\ exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=i}^{D} {x_i}^2}\right) - exp\left(\frac{1}{D}\sum_{i=1}^{D} \cos 2\pi x_i\right) + 20 + e \tag{B.5}$$

Problem 6 - Weierstrass function

$$f_{16}(x) = \sum_{i=1}^{D} \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k (x_i + 0.5))] - [a^k \cos(2\pi b^k (x_i \cdot 0.5))] \tag{B.6}$$

$$a = 0.5, b = 3.0, kmax = 20.0$$

Problem 7 - Griewank's function

$$f_{17}(x) = \frac{1}{4000} \sum_{i=1}^{D} {x_i}^2 - \prod_{i=1}^{D} \cos(\frac{x_i}{\sqrt{i}}) + 1 \tag{B.7}$$

Problem 8 - Rastrigin's function

$$f_{18}(x) = \sum_{i=1}^{D} [{x_i}^2 - 10\cos(2\pi x_i) + 10] \tag{B.8}$$

Problem 9 - Modified schwefel's function

$$f_{19}(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i) \tag{B.9}$$

$$z_i = x_i + 4.209687462275036\ e + 002$$

Problem 10 - Katsuura function

$$f_{20}(x) = \frac{10}{D^2} \prod_{i=1}^{D} (1 + i \sum_{j=1}^{3} 2 \frac{|2^j x_i - round(2^j x_i)|}{2^j})^{\frac{10}{D^{1.2}}} - \frac{10}{D^2} \tag{B.10}$$

Problem 11 - HappyCat function

$$f_{21}(x) = \left| \sum_{i=1}^{D} x_i^2 - D \right|^{1/4} + (0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/D + 0.5 \tag{B.11}$$

Problem 12 - HGBat function

$$f_{22}(x) = \left| (\sum_{i=1}^{D} x_i^2)^2 - (\sum_{i=1}^{D} x_i)^2 \right|^{1/2} + (0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)/D + 0.5 \tag{B.12}$$

Problem 13 - Expanded griewank's plus rosenbrock's function

$$f_{13}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + ... + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1)) \tag{B.13}$$

Problem 14 - Expanded scaffer's F6 function

$$Scaffer's\ F6\ function:\ g(x, y) = 0.5 + \frac{(\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1 + 0.001(x^2 + y^2))^2}$$

$$f_{14}(x) = g(x_1, x_2) + g(x_2, x_3) + ... + g(x_{D-1}, x_D) + g(x_D, x_1) \tag{B.14}$$

B2 Constrained Optimization Problems

The set of well known benchmark functions for global constrained optimization problems utilised are described below.

Problem g01: Minimize

$$f_{g01}(x) = (x_1 - 2)^2 - (x_2 - 1)^2 \tag{B.15}$$

Subject to:

$$h\ (x) = x_1 - 2x_2 + 1 = 0$$
$$g\ (x) = \frac{x_1^2}{4} + x_2^2 - 1 \leq 0$$

where $-10 \leq x_1 \leq 10$ and $-10 \leq x_2 \leq 10$.

Problem g02: Minimize

$$f_{g02}(x) = 5.35785474x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \tag{B.16}$$

Subject to:

$$g_1\ (x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \leq 0$$
$$g_2\ (x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \leq 0$$
$$g_3\ (x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813x_3^2 - 110 \leq 0$$
$$g_4\ (x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813x_3^2 + 90 \leq 0$$
$$g_5\ (x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \leq 0$$
$$g_6\ (x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \leq 0$$

where $78 \leq x_1 \leq 102$, $33 \leq x_2 \leq 45$, $27 \leq x_3 \leq 45$, $(i = 3, 4, 5)$.

Problem g03: Minimize

$$f_{g03}(x) = (x_1 - 10)^3 - (x_2 - 20)^3 \tag{B.17}$$

Subject to:

$$g_1\ (x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \leq 0$$
$$g_2\ (x) = (x_1 - 6)^2 - (x_2 - 5)^2 - 82.81 \leq 0$$

where $13 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 100$.

Problem g04: Minimize

$$f_{g04}(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + (x_5 - 3)^3$$
$$2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_10 - 7)^2 + 45 \tag{B.18}$$

Subject to:

$$
\begin{aligned}
g_1(x) &= -105 + 4x_1 + 5x_2 - 3x_2 - 3x_7 + 9x_8 \leq 0 \\
g_2(x) &= 10x_1 - 8x_2 - 17x_7 + 2x_8 \leq 0 \\
g_3(x) &= -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \leq 0 \\
g_4(x) &= 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x_3^2 - 7x_4 - 120 \leq 0 \\
g_5(x) &= 5x_1^2 + 8x_2(x_3 - 6)^2 - 2x_4 - 40 \leq 0 \\
g_6(x) &= x_1^2 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \leq 0 \\
g_7(x) &= 0.5(x_1 - 8)^2 + 2(x_2 - 4^2 + 3x_5^2 - x_6 - 30 \leq 0 \\
g_8(x) &= -3x_1 + x_2 + 12(x_9 - 8)^2 - 7x_{10} \leq 0
\end{aligned}
$$

where $-10 \leq x_1 \leq 10$, $(i = 1, ..., 10)$.

Problem g05: Minimize

$$
f(x) = -\frac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)} \tag{B.19}
$$

Subject to:

$$
\begin{aligned}
g_1(x) &= x_1^2 - x_2 + 1 \leq 0 \\
g_2(x) &= 1 - x_1 + (x_2 - 4)^2 \leq 0
\end{aligned}
$$

where $0 \leq x_1 \leq 100$ and $0 \leq x_2 \leq 10$.

Problem g06: Minimize

$$
\begin{aligned}
f_{g06}(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 \\
10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7
\end{aligned} \tag{B.20}
$$

Subject to:

$$
\begin{aligned}
g_1(x) &= -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \leq 0 \\
g_2(x) &= -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \leq 0 \\
g_3(x) &= -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \leq 0 \\
g_4(x) &= 4x_1 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \leq 0
\end{aligned}
$$

where $-10 \leq x_1 \leq 10$, $(i = 1, ..., 7)$.

Problem g07: Minimize

$$
f_{g07}(x) = x_1^2 + (x_2 - 1)^2 \tag{B.21}
$$

Subject to:

$$
h(x) = x_2 - x_1^2 = 0
$$

where $-1 \leq x_1 \leq 1$ and $-1 \leq x_2 \leq 1$.

Problem g08: Minimize

$$f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3 \tag{B.22}$$

Subject to:

$$h_1(x) = x_1^2 + x_2^2 + x_3^2 - 25 = 0$$
$$h_2(x) = 8x_1 + 14x_2 + 7x_3 - 56 = 0$$

where $0 \leq x_1 \leq 10$, $(i = 1, 2, 3)$.

Problem g09: Minimize

$$f_{g09}(x) = 0.000117y_{14} + 0.1365 + 0.00002358y_{13} + 0.0000001502y_{16} + 0.00321y_{12}$$
$$0.004324y_5 + 0.0001\frac{c_{15}}{c_{16}} + 37.48\frac{y_2}{c_{12}} - 0.0000005843y_{17} \tag{B.23}$$

Subject to:

$$g_1(x) = \frac{0.28}{0.72}y_5 - y_4 \leq 0$$
$$g_2(x) = x_3 - 1.5x_2 \leq 0$$
$$g_3(x) = 3496\frac{y_2}{c_{12}} - 21 \leq 0$$
$$g_4(x) = 110.6 + y_1 - \frac{62212}{c_{17}} \leq 0$$
$$g_5(x) = 213.1 - y_1 \leq 0$$
$$g_6(x) = y_1 - 405.23 \leq 0$$
$$g_7(x) = 17.505 - y_2 \leq 0$$
$$g_8(x) = y_2 - 1053.6667 \leq 0$$
$$g_9(x) = 11.275 - y_3 \leq 0$$
$$g_{10}(x) = y_3 - 35.03 \leq 0$$
$$g_{11}(x) = 214.228 - y_4 \leq 0$$
$$g_{12}(x) = y_4 - 665.585 \leq 0$$
$$g_{13}(x) = 7.458 - y_5 \leq 0$$
$$g_{14}(x) = y_5 - 584.463 \leq 0$$
$$g_{15}(x) = 0.961 - y_6 \leq 0$$
$$g_{16}(x) = y_6 - 265.916 \leq 0$$
$$g_{17}(x) = 1.612 - y_7 \leq 0$$
$$g_{18}(x) = y_7 - 7.046 \leq 0$$
$$g_{19}(x) = 0.146 - y_8 \leq 0$$
$$g_{20}(x) = y_8 - 0.222 \leq 0$$

$$g_{21}(x) = 107.99 - y_9 \leq 0$$
$$g_{22}(x) = y_9 - 273.366 \leq 0$$
$$g_{23}(x) = 922.693 - y_{10} \leq 0$$
$$g_{24}(x) = y_{10} - 1286.105 \leq 0$$
$$g_{25}(x) = 926.832 - y_{11} \leq 0$$
$$g_{26}(x) = y_{11} - 1444.046 \leq 0$$
$$g_{27}(x) = 18.766 - y_{16} \leq 0$$
$$g_{28}(x) = y_{12} - 537.141 \leq 0$$
$$g_{29}(x) = 1072.163 - y_{13} \leq 0$$
$$g_{30}(x) = y_{13} - 3247.039 \leq 0$$
$$g_{31}(x) = 8961.448 - y_{14} \leq 0$$
$$g_{32}(x) = y_{14} - 26844.086 \leq 0$$
$$g_{33}(x) = 0.063 - y_{15} \leq 0$$
$$g_{34}(x) = y_{15} - 0.386 \leq 0$$
$$g_{35}(x) = 71084.33 - y_{16} \leq 0$$
$$g_{36}(x) = -140000 + y_{16} \leq 0$$
$$g_{37}(x) = 2802713 - y_{17} \leq 0$$
$$g_{38}(x) = y_{17} - 12146108 \leq 0$$

where:

$$y_1 = x_2 + x_3 + 41.6$$
$$c_1 = 0.024x_4 - 4.62$$
$$y_2 = \frac{12.5}{c_1} + 12$$
$$c_2 = 0.0003535x_1^2 + 0.5311x_1 + 0.08705y_5x_1$$
$$c_3 = 0.052x_1 + 78 + 0.002377y_2x_1$$
$$y_3 = \frac{c_2}{c_3}$$
$$y_4 = 19y_3$$
$$c_4 = 0.04782(x_1 - y_3) + \frac{0.1956x_1 - y_3{}^2}{x_2} + 0.6376y_4 + 1.594y_3$$
$$c_5 = 100x_2$$
$$c_6 = x_1 - y_3 - y_4$$
$$c_7 = 0.950 - \frac{c_4}{c_5}$$
$$y_5 = c_6c_7$$
$$y_6 = x_1 - y_5 - y_4 - y_3$$
$$c_8 = (y_5 + y_4)0.995$$
$$y_7 = \frac{c_8}{y_1}$$

$$
\begin{aligned}
y_8 &= \frac{c_8}{3798} \\
c_9 &= y_7 - \frac{0.0663y_7}{y_8} - 0.3153 \\
y_9 &= \frac{96.82}{c_9} + 0.321y_1 \\
y_{10} &= 1.29Y_5 + 1.258y_4 + 2.29y_3 + 1.71y_6 \\
y_{11} &= 1.71x_1 - 0.452y_4 + 0.580y_3 \\
c_{10} &= \frac{12.3}{752.3} \\
c_{11} &= (1.75y_2)(0.995x_1) \\
c_{12} &= 0.995y_{10} + 1998 \\
y_{12} &= c_{10}x_1 + \frac{c_{11}}{C_{12}} \\
y_{13} &= c_{12} - 1.75y_2 \\
y_{14} &= 3623 + 64.4x_2 + 58.4x_3 + \frac{146312}{y_9 + x_5} \\
c_{13} &= 0.995y_{10} + 60.8x_2 + 48x_4 - 0.1121y_{14} - 5095 \\
y_{15} &= \frac{y_{13}}{c_{13}} \\
y_{16} &= 148000 - 3310000y_{15} + 40y_{13} - 61y_{15}y_{13} \\
c_{14} &= 2324y_{10} - 28740000y_2 \\
y_{17} &= 14130000 - 1328y_{10} - 531y_{11} + \frac{c_{14}}{c_{12}} \\
c_{15} &= \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52} \\
c_{16} &= 1.104 - 0.72y_{15} \\
c_{17} &= y_9 + x_5
\end{aligned}
$$

bounded with $704.4148 \leq x_1 \leq 906.3855$, $68.6 \leq x_2 \leq 288.88$, $0 \leq x_3 \leq 134.75$, $193 \leq x_4 \leq 287.0966$ and $25 \leq x_5 \leq 84.1988$.

Problem g10: Minimize

$$f_{g10}(x) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7) \tag{B.24}$$

Subject to:

$$
\begin{aligned}
g_1\,(x) &= x_3^2 + x_4^2 - 1 \leq 0 \\
g_2\,(x) &= x_9^2 - 1 \leq 0 \\
g_3\,(x) &= x_5^2 + x_6^2 - 1 \leq 0 \\
g_4\,(x) &= x_1^2 + (x_2 - x_9)^2 - 1 \leq 0 \\
g_5\,(x) &= (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \leq 0 \\
g_6\,(x) &= (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \leq 0
\end{aligned}
$$

$$g_7(x) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \leq 0$$
$$g_8(x) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \leq 0$$
$$g_9(x) = x_7^2 + (x_8 - x_9)^2 - 1 \leq 0$$
$$g_{10}(x) = x_2 x_3 - x_1 x_4 \leq 0$$
$$g_{11}(x) = -x_3 x_9 \leq 0$$
$$g_{12}(x) = x_5 x_9 \leq 0$$
$$g_{13}(x) = x_6 x_7 - x_5 x_8 \leq 0$$

where $-10 \leq x_1 \leq 10$, $(i = 1, ..., 8)$ and $1 \leq x_9 \leq 20$.

B3 Engineering Optimization Problems

The set of well known engineering problems comprising constrained optimization problems utilised in the research are described below,

Problem e01: Design of a welded beam problem

$$f_{e01}(x) = 1.10471{x_1}^2 x_2 + 0.04811 x_3 x_4 (14.0 + x_2) \tag{B.25}$$

Subject to:

$$g_1(x) = \tau(x) - \tau_{max} \leq 0$$
$$g_2(x) = \sigma(x) - \sigma_{max} \leq 0$$
$$g_3(x) = x_1 - x_4 \leq 0$$
$$g_4(x) = 0.10471{x_1}^2 + 0.04811 x_3 x_4 (14.0 + x_2) - 5.0 \leq 0$$
$$g_5(x) = 0.125 - x_1 \leq 0$$
$$g_6(x) = \delta(x) - \delta_{max} \leq 0$$
$$g_7(x) = P - P_c(x) \leq 0$$

where:

$$\tau(x) = \sqrt{(\tau)^2 + 2\tau\tau\frac{x_2}{2R} + (\tau)^2},\ \tau' = \frac{P}{\sqrt{2}x_1 x_2},\ \tau'' = \frac{MR}{J}$$
$$M = P\left(L + \frac{x_2}{2}\right),\ R = \sqrt{{x_2}^2}$$
$$J = 2\left\{\sqrt{2}x_1 x_2\left[\frac{{x_2}^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\}$$
$$\sigma(x) = \frac{6PL}{x_4 {x_3}^2},\ \delta(x) = \frac{4PL^2}{E{x_3}^3 x_4}$$
$$P_c(x) = \frac{4.013E\sqrt{\frac{{x_3}^2{x_4}^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right)$$
$$P = 6000\ lb, L = 14\ in, E = 30 \times 10^6\ psi, G = 12 \times 10^6\ psi$$

Bounded with $0.1 \leq x_1, x_4 \leq 2$, $0.1 \leq x_2, x_3 \leq 10$

Problem e02: Design of a pressure vessel problem

$$f_{e02}(x) = 0.6224x_1x_2 + 1.7781x_2{x_3}^2 + 3.1661{x_1}^2x_4 + 19.84{x_1}^2x_3 \tag{B.26}$$

Subject to:

$$\begin{aligned}
g_1(x) &= -x_1 + 0.0193x_3 \leq 0 \\
g_2(x) &= -x_2 + 0.00954x_3 \leq 0 \\
g_3(x) &= -\pi {x_3}^2 x_4 - \frac{4}{3}\pi {x_3}^3 + 1296000 \leq 0 \\
g_4(x) &= x_4 - 240 \leq 0
\end{aligned}$$

Bounded with $0 \leq x_1, x_2 \leq 99$, $10 \leq x_3, x_4 \leq 200$.

Problem e03: Design of a tension / compression spring problem

$$f_{e03}(x) = (x_3 + 2)x_2{x_1}^2 \tag{B.27}$$

Subject to:

$$\begin{aligned}
g_1(x) &= 1 - \frac{{x_2}^3x_3}{71785{x_1}^4} \leq 0 \\
g_2(x) &= \frac{4{x_2}^2 - x_1x_2}{12566(x_2{x_1}^3 - {x_1}^4)} + \frac{1}{5108{x_1}^2} - 1 \leq 0 \\
g_3(x) &= 1 - \frac{140.45x_1}{{x_2}^2x_3} \leq 0 \\
g_4(x) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0
\end{aligned}$$

Bounded with $0.05 \leq x_1 \leq 2$, $0.25 \leq x_2 \leq 1.3$, $2 \leq x_3 \leq 15$.

Problem e04: Design of a speed reducer problem

$$\begin{aligned}
f_{e04}(x) = {} & 0.7854x_1{x_2}^2(3.3333{x_3}^2 + 14.9334x_3 - 43.0934) \\
& - 1.508x_1({x_6}^2 + {x_7}^2) + 7.4777({x_6}^3 + {x_7}^3) + 0.7854(x_4{x_6}^2 + x_5{x_7}^2)
\end{aligned} \tag{B.28}$$

Subject to:

$$\begin{aligned}
g_1(x) &= \frac{27}{x_1{x_2}^2x_3} - 1 \leq 0 \\
g_2(x) &= \frac{397.5}{x_1{x_2}^2{x_3}^2} - 1 \leq 0 \\
g_3(x) &= \frac{1.93{x_4}^3}{x_2{x_6}^4x_3} - 1 \leq 0 \\
g_4(x) &= \frac{1.93{x_5}^3}{x_2{x_7}^4x_3} - 1 \leq 0
\end{aligned}$$

$$g_5(x) = \frac{\sqrt{745\left(\frac{x_4}{x_2x_3}\right)^2 + (16.9\times 10^6)}}{110x_6^3} - 1 \le 0$$
$$g_6(x) = \frac{\sqrt{745\left(\frac{x_5}{x_2x_3}\right)^2 + (157.5\times 10^6)}}{85x_7^3} - 1 \le 0$$
$$g_7(x) = \frac{x_2x_3}{40} - 1 \le 0$$
$$g_8(x) = \frac{5x_2}{x_1} - 1 \le 0$$
$$g_9(x) = \frac{x_1}{12x_2} - 1 \le 0$$
$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0$$
$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \le 0$$

Bounded with $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.3 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5.5 \le x_7 \le 5.5$.

Problem e05: Design of a gear train problem

$$f_{e05}(x) = ((1/6.931) - (x_3x_2/x_1x_4))^2 \tag{B.29}$$

Subject to: $12 \le x_i \le 60$, $(i = 1, 2, 3, 4)$.

B4 Multi-objective Benchmark Functions

Multi-objective optimisation problems comprising unconstrained, constrained and engineering design problems considered are described below.

B4.1 Multi-objective Functions for Unconstrained Benchmark Problems

Problem MO1: Schaffer function 1 (SCH1):
Minimize

$$f_1(x) = x^2 \\ f_2(x) = (x-2)^2 \tag{B.30}$$

where $-10 \le x_1 \le 10$ and $1 \le i \le 20$.

Problem MO2 Zitzler-Deb-Thiele's function (ZDT1):
Minimize

$$f_1(x) = x_1 \\ f_2(x) = \left(1 + \frac{9}{(n-1)} \sum_{i=2}^{n} x_1\right)(1 - \sqrt{f_1/g}) \tag{B.31}$$

where $0 \le x_1 \le 1$ and $1 \le i \le 20$.

Problem MO3 Kursawe function:
Minimize

$$f_1(x) = \sum_{i=2}^{2} \left(-10e^{-0.2\sqrt{x_i^2 + x_{i+1}^2}}\right) \tag{B.32}$$

$$f_2(x) = \sum_{i=2}^{2} \left(|x_i|^{0.8} + 5\sin x_i^3\right)$$

where $-5 \le x_1 \le 5$ and $1 \le i \le 3$.

B4.2 Multi-objective Functions for Constrained Benchmark Problems

Problem MO4: CTP 1 function:
Minimize

$$f_1(x_1, x_2) = x_1 \\ f_2(x_1, x_2) = (1 + x_2)e^{\left(\frac{x_1}{1+x_2}\right)} \tag{B.33}$$

subject to:

$$g_1(x_1, x_2) = \frac{f_2(x_1, x_2)}{0.858e^{(-0.541 f_1(x_1, x_2))}} \ge 1$$

$$g_2(x_1, x_2) = \frac{f_2(x_1, x_2)}{0.728e^{(-0.295 f_1(x_1, x_2))}} \ge 1$$

where $-5 \leq x_1, x_2 \leq 1$.

Problem MO5: Constr-Ex function:
Minimize

$$f_1(x_1, x_2) = x_1 \qquad \text{(B.34)}$$
$$f_2(x_1, x_2) = \frac{1 + x_2}{x_1}$$

subject to:

$$g_1(x_1, x_2) = x_2 + 9x_1 \geq 6$$
$$g_1(x_1, x_2) = -x_2 + 9x_1 \geq 1$$

where $0.1 \leq x_1 \leq 1$ and $0 \leq x_2 \leq 5$.

Problem MO6: Binh and Korn function:
Minimize

$$f_1(x_1, x_2) = 4x_1^2 + 4x_2^2 \qquad \text{(B.35)}$$
$$f_2(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 5)^2$$

subject to:

$$g_1(x_1, x_2) = (x_1 - 5)^2 + x_2^2 \leq 25$$
$$g_2(x_1, x_2) = (x_1 - 8)^2 + (x_2 + 3)^2 \geq 7.7$$

where $0 \leq x_1 \leq 5$ and $0 \leq x_2 \leq 3$.

Problem MO7: Changkong and Haimes function:
Minimize

$$f_1(x_1, x_2) = 2 + (x_1 - 2)^2 + (x_2 - 1)^2 \qquad \text{(B.36)}$$
$$f_2(x_1, x_2) = 9x_1 - (x_2 - 1)^2$$

subject to:

$$g_1(x_1, x_2) = (x_1)^2 + x_2^2 \leq 225$$
$$g_2(x_1, x_2) = x_1 - 3x_2 + 10 \leq 0$$

where $20 \leq x_1, x_2 \leq 20$.

B4.3 Multi-objective Functions for Engineering Design Problems

Problem MO8: Four bar plane truss problem:
Minimize

$$f_1(x) = L(2x_1 + \sqrt{2}x_2 + \sqrt{x_3} + x_4) \qquad \text{(B.37)}$$
$$f_2(x) = \frac{FL}{E}\left(\frac{2}{x_1} + \frac{2\sqrt{2}}{x_2} - \frac{2\sqrt{2}}{x_3} + \frac{2}{x_4}\right)$$

subject to:

$$
\begin{aligned}
(F/\sigma) &\leq x_1 \leq 3(F/\sigma) \\
\sqrt{2}(F/\sigma) &\leq x_2 \leq 3(F/\sigma) \\
\sqrt{2}(F/\sigma) &\leq x_3 \leq 3(F/\sigma) \\
(F/\sigma) &\leq x_4 \leq 3(F/\sigma)
\end{aligned}
$$

where $F = 10kN$, $E = 2 \times \frac{10^5 kN}{cm^2}$, $L = 200cm$, $\sigma = \frac{10kN}{cm^2}$.