VARIANTS OF GREY WOLF OPTIMIZER AND SINE COSINE ALGORITHM FOR GLOBAL OPTIMIZATION AND THEIR APPLICATIONS

Ph. D. THESIS

by

SHUBHAM GUPTA

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JULY, 2019

VARIANTS OF GREY WOLF OPTIMIZER AND SINE COSINE ALGORITHM FOR GLOBAL OPTIMIZATION AND THEIR APPLICATIONS

A THESIS

Submitted in partial fulfilment of the requirements for the award of the degree

of

DOCTOR OF PHILOSOPHY

in

MATHEMATICS

by

SHUBHAM GUPTA

DEPARTMENT OF MATHEMATICS
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)
JULY, 2019

INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled "**VARIANTS OF GREY WOLF OPTIMIZER AND SINE COSINE ALGORITHM FOR GLOBAL OPTIMIZATION AND THEIR APPLICATIONS**" in partial fulfilment of the requirements for the award of the Degree of Doctor of Philosophy and submitted in the Department of Mathematics of the Indian Institute of Technology Roorkee, Roorkee is an authentic record of my own work carried out during a period from December, 2015 to July, 2019 under the supervision of Dr. Kusum Deep, Professor, Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other Institution.

(SHUBHAM GUPTA)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Kusum Deep)
Supervisor

Date: July , 2019

Abstract

Grey Wolf Optimizer (GWO) and Sine Cosine Algorithm (SCA) are recently developed population based metaheuristic algorithms to solve global optimization problems. The GWO is inspired by the social and leadership behaviour of grey wolves, and the SCA is designed from the inspiration of sine and cosine trigonometric functions. Although these algorithms are recently developed, their effectiveness and advantages are demonstrated in various real world applications like feature selection, thresholding, multi objective optimization, load dispatch problem in electrical engineering, clustering and training of neural network etc.

The aim of this PhD Thesis is to propose some modified variants of the classical GWO and classical SCA which are more effective and reliable in terms of search strategy and solution accuracy of the optimization problems. To achieve these objectives, in the Thesis, First a modified variant of classical GWO called RW-GWO is introduced which improves the exploration as well as exploitation ability of the wolves in a grey wolf pack by introducing two different strategies. In the first strategy, a new search equation based on random walk search mechanism is introduced for the leading hunters, and in second, a greedy selection is applied at the end of each iteration corresponding to each wolf between its current and previous state. The random walk search strategy focuses on enhancing the exploration and exploitation ability of leading guidance and greedy selection preserves the discovered promising areas of the search space. The performance of the RW-GWO algorithm is analyzed and compared with classical GWO on IEEE CEC 2014 benchmark set of unconstrained optimization problems. The numerical results of these test problems demonstrate the superior search ability of proposed algorithm as compared to classical GWO.

Next, another variant of the classical GWO called Memory-based Grey Wolf Optimizer (mGWO) is introduced. The mGWO algorithm utilizes the personal best history of individual wolves to enhance the collaborative strength of grey wolf pack through modified encircling and hunting mechanism. The mGWO also integrates the personal best guidance during the search to share the available best knowledge regarding the search space among the individual search agents. Hence, the leading and personal best guidance together perform the search process in the mGWO. The evaluation of the proposed mGWO is performed on IEEE CEC 2014 benchmark set of unconstrained problems. The numerical results of these test problems demonstrate the better search ability of proposed algorithm as compare to the classical GWO in all the category of optimization problems such as unimodal, multimodal, composite and hybrid functions. The comparison

between the RW-GWO and mGWO concludes that RW-GWO can be preferred for the unimodal and composite problems and for the multimodal and hybrid problems mGWO can be preferred.

To improve the search accuracy of candidate solutions, a new variant of classical SCA called m-SCA is proposed in the Thesis which is based on opposition-based learning and modified position update mechanism. The opposition-based learning is used to generate the opposite candidate solutions so that the stagnation at local optima can be avoided. The jumping rate which allows the algorithm to perform the opposition-based learning phase in the algorithm is fixed to a low value to keep the balance between exploration and exploitation. The search equation of classical SCA is modified based on the cognitive component to reduce the inefficient diversity of search agents and to maintain the balance between exploration and exploitation during the search. The performance of the m-SCA is analyzed and compared with classical SCA on unconstrained benchmark problems given in IEEE CEC 2014. The analysis of the results demonstrates the superior search ability of the m-SCA as compared to classical SCA on all category of problems such as unimodal, multimodal, composite and hybrid benchmark problems.

Next, another modified variant of classical SCA called ISCA is introduced which enhances the performance of the classical SCA based on the personal best history of candidate solutions, crossover operator and modified position update mechanism. In the ISCA, the greedy selection is also employed for each candidate solution between its current and previous state to avoid its divergence from discovered promising search areas. The performance evaluation of the proposed algorithm is performed on IEEE CEC 2014 benchmark suite of unconstrained optimization problems. The numerical results of these test problems demonstrate the superior search ability of proposed algorithm as compared to the classical SCA in all category of benchmark optimization problems. The comparison between the m-SCA and ISCA concludes that ISCA can be preferred for the unimodal, multimodal and hybrid problems and for the composite problems both the algorithms are very competitive to each other.

Further, the performance of classical versions of GWO and SCA, and their proposed variants called RW-GWO, mGWO, m-SCA and ISCA is evaluated on constrained optimization problems. The constrained versions of these algorithms are designed by introducing a simple constraint handling mechanism based on the constraint violation. The constrained benchmark problems given in IEEE CEC 2006 are used for experimentation. The analysis on these problems demonstrate the better search ability of the mGWO algorithm than the classical GWO and RW-GWO algorithms. Similarly, the proposed ISCA algorithm shows their better search ability to solve constrained optimization problems as compared to the classical SCA and m-SCA.

In order to analyze the applicability of the classical GWO, classical SCA and their proposed variants, an unconstrained and nonlinear optimization problem which arises in the field of image processing is selected. The problem is defined to determine the optimal thresholds for image segmentation in grey images. To find the optimum thresholds for an image, Otsu's between-class variance criterion is employed as the fitness function. Nine benchmark images are used for experimentation and several statistical measures are used for comparison. The analysis of results ensure that the proposed improved variant RW-GWO and mGWO perform better than classical GWO, classical SCA, m-SCA and ISCA algorithms.

Next, the classical GWO, classical SCA and their proposed variants called RW-GWO, mGWO, m-SCA and ISCA are implemented on another real-life application which is unconstrained in nature and arises in the field of electrical engineering. The objective of this problem is to determine the optimal setting for the proper coordination of overcurrent relays. The IEEE 3, 4, 6, and 14-bus systems are used for experimentation and validation. The comparison of results demonstrate the better search efficiency and solution accuracy of the proposed RW-GWO algorithm than all other variants of GWO and SCA and their classical versions in finding the optimal setting for overcurrent relays.

Finally, the Thesis is concluded with the limitations and scope of the proposed algorithms. Later it suggests future scope and some new directions of research in this area.

List of Publications

1. Gupta, S., & Deep, K. (2019). A novel random walk grey wolf optimizer. Swarm and Evolutionary Computation, 44, 101-112. Elsevier.
2. Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Systems with Applications, 119, 210-230. Elsevier.
3. Gupta, S., & Deep, K. (2019). Improved sine cosine algorithm with crossover scheme for global optimization. Knowledge-Based Systems, 165, 374-406. Elsevier.
4. Gupta, S., & Deep, K. (2018). Random walk grey wolf optimizer for constrained engineering optimization problems. Computational Intelligence, 34(4), 1025-1045. Wiley.
5. Gupta, S., & Deep, K. (2019). Optimal coordination of overcurrent relays using improved leadership based Grey Wolf Optimizer. Arabian Journal for Science and Engineering. Springer. (Accepted). DOI: 10.1007/s13369-019-04025-z.
6. Gupta, S., & Deep, K. (2019). A Memory-based Grey Wolf Optimizer for global optimization and image segmentation. Expert Systems with Applications. Elsevier. (Revision submitted).

Acknowledgements

I must express my heartiest thanks to the God Almighty, for His blessings to write and complete this Thesis. It gives me immense pleasure to express my deep sense of gratitude and sincere thanks to my supervisor Dr. Kusum Deep, Professor, Department of Mathematics, Indian Institute of Technology Roorkee, for her valuable guidance, constructive comments and encouragement throughout the whole Thesis work. I offer my deep sense of reverence and profund indebtedness to her for sincere concern towards my research work with truly scientific intuition and ideas. Her unique ways of tackling problems from strong fundamentals and clear ideas about the subject helped me in smooth progress of this research work.

I am indebted to the Department of Mathematics, Indian Institute of Technology Roorkee, to all its faculty and staff for their academic support and encouragement. I am grateful to Dr. N. Sukavanam, Head of the Department, Dr. V.K. Katiyar, former Head of the Department, Dr. Kusum Deep, DRC Chairperson for their valuable suggestions and constant moral support during my research work. I am thankful to Dr. Dhish Saxena and Dr. Madhu Jain for their support and encouragement. I am gratefully acknowledge for the financial support by research grant number MHR-02-41-113-429 from MHRD, India.

Collective and individual acknowledgements are also owed to my colleagues at IIT Roorkee. The inspiration, support and cooperation that I have received from them are beyond the scope of my acknowledgement. Many thanks go, in particular, to Dr. J. C. Bansal, Dr. M. Thakur, Dr. Anupam Yadav, Dr. Assif Assad, Dr. Teekam Singh, Dr. Shail Dinkar, Dr. Vishnu Singh, Dr. Kiran Kumar Behera, Dr. Sumit Malik, Dr. Abhishek Banerjee, Dr. Vivek Kumar, Amit Kumar, Jagannath Bhanja, and Sudeep Singh Sanga for their cooperation and support for making my stay at IIT Roorkee a memorable one. I am also thankful to my colleagues Vishesh, Abhishek, Pankaj, Lateef, Anshul, Karuna, Preeti, Ashish, Prince and Manish for providing healthy and progressive research environment.

I make an effort to record my affectionate thanks to my parents Mr. Vijay Gupta and Mrs. Suman Gupta for having faith in me, their unconditional support and love during all my studies. Their encouragement has always given me strength to move ahead in my life. Without their care and support, this Thesis work would not have been possible.

At last but not the least, I wish my sincere gratitude to my family members and thanks to my elder sister and brother Anjali and Aditya for being my best friends and always providing me positive environment and strength in my life.

Finally, I would like to thank everybody who was important to successful realization of Thesis, as well as expressing my apology that I could not mention one by one.

Roorkee (Shubham Gupta)

July , 2019

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

This introductory Chapter states the definitions and underlines the objectives and motivation behind this Thesis. It also reviews the available literature. The chapter closes with a brief summary of the work presented in this Thesis as well as future research directions.

1.1. Optimization

Optimization is the methodology of choosing "the best" alternative(s) among a specified set of available options. This approach of determining "the largest" / "the smallest" possible value, that a given mathematical expression can attain in its specified domain of definition, is called optimization. The mathematical expression that has to be optimized can be linear, nonlinear, integer, geometric or fractional. In some situations, explicit mathematical formulation of the function is not readily defined or may not be available. Many times the mathematical function which needs to be optimized has restrictions in the form of inequality or equality constraints. Therefore, the process of optimization can be considered as a problem of finding those values of the independent variables which do not violate the inequality and equality constraints in such a way to provide an optimal value of the mathematical function being optimized. In other words, the mathematical techniques for determining the optimal value(s) ("the greatest possible value" or "the least possible value") of a mathematical function are called 'Optimization Techniques'. Determining the solution of most realistic problems may not be possible in the absence of robust optimization techniques. In literature, numerous books are available based on mathematical concepts of optimization and some of references are [1-9].

Optimization problems arise in various fields of science, engineering, software industry, economics, manufacturing system, physical science and transportation etc. In view of their applicability, it is necessary to design and develop efficient and reliable computational algorithms.

1.2. Definition of an Optimization Problem

Mathematically speaking, the most general formulation of single objective optimization problem is:

$$Max/Min \quad F(X), \quad X = (x_1, x_2, \dots, x_D) \in R^D \tag{1.1}$$

$$s.t. \quad g_j(X) \leq 0 \quad j = 1,2, \dots, J \tag{1.2}$$

$$h_k(X) = 0 \quad k = 1,2, \dots, K \tag{1.3}$$

$$l_i \leq x_i \leq u_i \quad i = 1,2, \dots, D \tag{1.4}$$

where $F, g_1, g_2, \dots, g_J, h_1, h_2, \dots h_K$ are real valued functions.

Function $F(X)$ that is to be optimized (maximized or minimized) is called the 'objective function'. Inequalities $g_j(X) \leq 0$ for $j = 1,2, \dots, J$ are known as the inequality constraints and equalities $h_k(X) = 0$ for $k = 1,2, \dots, K$ are called equality constraints. It is desired to determine those values of the independent variables $x_1, x_2, \dots, x_D$ which optimize the objective function without violating any of the restriction, imposed in equation (1.2), (1.3) and (1.4). The variables x_i's are known as 'decision variables'. l_i's are the lower bounds and u_i's are the upper bounds of the decision variables x_i. A decision vector $X = (x_1, x_2, \dots, x_D) \in R^D$ which satisfies all the constraints is called a 'feasible solution'. A feasible solution which optimizes the objective function is called a feasible optimal solution.

On the basis of presence of constraints, there are two types of optimization problems named unconstrained optimization problems and constrained optimization problems. Unconstrained optimization problems involve an objective function given by equation (1.1) or lower or upper bounds on variables given by equation (1.4). Constrained optimization problems involve an objective function given in equation (1.1), the box constraints given by equation (1.4), inequality constraints given by equation (1.2) and/or linear or/and non-linear, equality constraints given by equation (1.3). Due to presence of inequality and equality constraints, constrained optimization problems are more difficult to solve.

1.3. Local and Global Optimal Solutions

Let S denote the feasible region of the solution vectors that satisfies all the constraints of an optimization problem. Then, in case of a minimization problem, if for $\bar{X} \in S$ there exists a neighbourhood $N_{\in}(\bar{X})$ around $\bar{X}$ such that $F(\bar{X}) \leq F(X)$ for each $X \in S \cap N_{\in}(\bar{X})$, then $\bar{X}$ is known as a 'local minimum solution'. However, if, $\bar{X} \in S$ and $F(\bar{X}) \leq F(X)$ for all $X \in S$ then $\bar{X}$ is known as a 'global minimum solution' of the optimization problem at hand. Fig 1.1 shows local and global optimum solutions of a mathematical function.

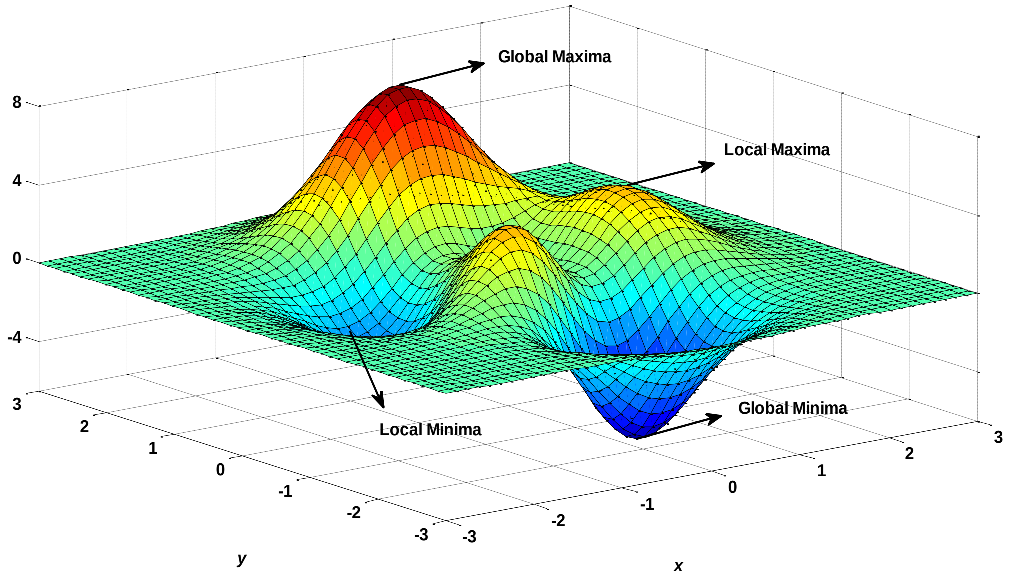


Fig 1.1: Demonstration of local optima and global optima

In general it may happen that there are either no optimal solutions, or a unique optimal solution or several optimal solutions, for a given nonlinear optimization problem. In case if a problem has a single local optimal solution then it is also the global optimal solution. If, however, the optimization problem has several local optimal solutions, then, in general, one or more of them could be the global optimal solutions. In a Linear Programming Problem, it is for sure that, every local optimal solution is the global optimal solution. On the contrary, in case of a Non Linear Optimization Problem, if the objective function is convex (for minimization case) and its feasible domain is also convex, then it is guaranteed that the local optimal solution is also the global optimal solution.

In many nonlinear optimization problems, it is usually desirous to determine a global optimal solution instead of a local optimal solution. But, in general, it is often difficult to obtain the global optimal solution of a nonlinear optimization problem, rather than finding the local optimal solution. However, due to its practical significance, it becomes necessary to determine the global optimal solution.

For a mathematical function which is twice-differentiable, there exist conditions which may be used to determine a local optimal solution. In case the test fails, then due to the property of continuous differentiability of function a solution with a lesser objective function value can be determined in its neighbourhood. Thus, a sequence of solutions can be constructed which converge to the local optimal solution. However, in general, such tests are not sufficient. It may be said that, a global optimization problem is not solvable in a finite number of steps. Therefore any given solution cannot be guaranteed as a solution of global minima without evaluating the objective function at least at one solution of its neighbourhood. But, the neighbourhoods of a solution may be unbounded, therefore, an infinite numbers of steps are required to attain the global minima.

1.4. Methods for Global Optimization

Global optimization focuses on finding the best of the local minima. Designing global optimization techniques is not an easy task since, in general, there is no criterion for deciding whether a global optimal solution has been achieved or not. In view of the practical necessity and with the availability of fast and readily computing machines, many computational techniques are now being reported in literature for solving nonlinear optimization problems. The methods currently available in literature for solving nonlinear global optimization problems may be broadly classified as deterministic methods and probabilistic methods.

The deterministic methods try to guarantee that a neighbourhood of the global optima is attained. Such methods do not use any stochastic techniques, but rely on a thorough search of the feasible domain. However, they are applicable only to a restricted class of functions. On the other hand, probabilistic methods are used to find the near optimal solution. This is achieved by assuming that the good solutions are near to each other in the search space. This assumption is valid for most real life problems [10]. The probabilistic methods uses the probabilistic or stochastic approach to search for the global optimal solutions. Although probabilistic methods do not give an absolute guarantee, these methods are sometimes preferred over the deterministic methods because they are applicable to a wider class of problems. Several other methods can also be used as optimization task [11-13].

1.5. Nature Inspired Optimization Algorithms

One of the most striking trend that emerged in the optimization field is the simulation of natural processes as efficient global search methods. The natural processes or phenomena are firstly analysed mathematically and then coded as computer programs for solving complex nonlinear real world problems. The resulting methods are called ‘Nature Inspired Algorithms (NIA)’ that can often outperform classic methods. The advantages of these methods are their ability to solve various standard or application based problems successfully without any prior knowledge of the problem space. Moreover, these algorithms are more likely to obtain the global optima of a given problem. They do not require any continuity and differentiability of the objective functions and / or constraints. Also, they work on a randomly generated population of solutions instead of one solution. They are easy to programme and can be easily implemented on a computer.

The most primitive subfield of nature inspired optimization techniques is the evolutionary algorithms which mimics the concepts of evolution in nature. Genetic Algorithms [14], Genetic Programming [15] and Differential Evolution [16] are some famous evolutionary algorithms. Genetic Algorithm is based on the Darwin's Theory of Evolution which is based on the property of inheritance and survival of the fittest in living organisms. The decision parameters are encoded into encoded space (Binary / Real / Octal, etc.) and crossover, mutation and elitism is performed over a number of generations until a prespecified stopping criteria is attained. Genetic programming is an extension of genetic algorithms in which the programs are expressed as syntax trees rather than as lines of code. Differential Evolution uses only the mutation operator on a target vector. In [17], Ali and Zhu have extended Differential Evolution for constrained optimization using penalty function. These evolutionary algorithms have been applied to solve various real-world application problems [18-26].

Another important development in the area of nature inspired algorithms is the introduction of Particle Swarm Optimization [27]. It mimics the behavior of a flock of birds or school of fish. All the solutions or particles of the swarm fly through the search space using their personal best position in history as well as the global best position of the entire swarm. In [28, 29], an improved PSO is proposed to obtain faster convergence. Particle Swarm Optimization has been applied to many real world problems [30-37].

Glow Worm Swarm Optimization [38, 39] mimics the behavior of glow worms which emit light in order to attract the others in the group for mating. It is particularly designed to capture multiple local and global optima.

Artificial Bee colony optimization [40] is based on self-organization and division of labour, i.e., it is based on inspecting the behaviour of bees on finding nectar and sharing the information of food sources to the bees in the hive, by the employed bees, onlooker bees and scouts. Artificial Bee colony optimization has been applied to many real world problems including [41, 42]. In [43, 44], several analysis have been conducted to for the stability analysis of Artificial Bee Colony algorithm.

Invasive Weed Optimization (IWO) [45] is inspired by the growth process of weeds in nature. It has been applied to solve various real-world applications [46-50].

Another Swarm Intelligence based algorithm is the Spider Monkey Algorithm [51]. It is based on the foraging behavior and fission-fusion social structures of spider monkeys.

Ant Colony Optimization [52] is proposed wherein the pheromone left behind ants and their ability to change their path when an obstacle is encountered on their path, is mimicked into the design of Ant colony optimization.

The behavior of the growth of bacteria forms a basis of Bacterial Foraging Optimization Algorithm [53].

Some methods draw their inspiration from the physical laws of nature. For example Gravitational Search Algorithm [54] is based on gravitational interaction between masses. It artificially simulates the Newton's Theory, Newtonian laws of gravitation and motion. Similarly, Central Force Optimization [55-57] is based on gravitational kinematics.

Harmony Search Algorithm [58] is another nature inspired optimization which is inspired from music. Harmony Search Algorithm has been applied to solve various real-world applications [59, 60].

These days many new nature inspired optimization techniques are being proposed by researchers. Some of them are: Water drop Algorithm [61], Ant Lion Algorithm [62], Firework Algorithms [63], Teaching Learning Based Optimization [64], Water Weed Optimization [45], Kidney Inspired Optimization [65], and Moth-flame Optimization Algorithm [66].

An excellent review of Nature Inspired Optimization Techniques is presented in [67-72].

The scope of this Thesis is Grey Wolf Optimizer (GWO) and Sine Cosine Algorithm (SCA), a nature inspired optimization techniques for global optimization problems.

1.6. The No Free Lunch Theorem

A major and interesting result in optimization theory was the presentation of the "No Free Lunch (NFL) theorem" given by Wolpert and Macready [73, 74]. This theorem states that "the performance of all optimization (search) algorithms, amortized over the set of all possible functions, is equivalent". The theorem has far reaching implications, because it implies that "no algorithm can be designed so that it will be superior to a linear enumeration of the search space, or even a purely random search". Although, the theorem is defined over finite search spaces only, however, it is not proved if the result is applicable to infinite search spaces, e.g. R^d. All computer implementations of search algorithms will, in general, operate on finite search spaces, therefore the theorem is applicable to all existing algorithms. The NFL Theorem states that all search algorithms perform equally well over all functions, it does not necessarily hold for all subsets of this set. The set of all functions over a finite domain includes the set of all the permutations of this domain.

1.7. Grey Wolf Optimizer (GWO)

Grey Wolf Optimizer (GWO) is one of the efficient and reliable algorithm based on the swarm intelligence of grey wolves. This algorithm was developed in 2014, by Mirjalili et al. [75] by analyzing the social and dominant leadership characteristic in grey wolf pack. Grey wolves always try to find an optimal way to find the prey. In the hunting mechanism by grey wolves, a leadership hierarchy is followed which is shown graphically in Fig 1.2. In a grey wolf pack, wolves are divided into four different groups. These groups are divided according to the intelligence and strength of wolves. In the first group of wolves, the dominant or leading wolf is included, which is known as alpha (α). Alpha wolf is responsible for all the decisions which are very crucial for the pack such as selecting a place for staying, how to attack on prey. In the second group of wolves, subordinate wolf to the alpha named as a beta(β) is included. Beta wolf transfers all the essential information provided by alpha to the other wolves of the pack and serves as a main leading wolf for the pack in the absence of alpha. In the third group, sentinels, caretakers, and hunters of the pack are included. These wolves are known as delta (δ). The last group of wolves is known as omega (ω) and they have the assent of eating the meal in the end after all other wolves.

Muro et al. [76] observed that in the hunting process of prey, grey wolves follow the three main steps namely,

i. Tracking and approaching the prey.
ii. Encircling the prey.
iii. Attack towards the prey.

In the algorithm, the assumption has been presumed that beta and delta wolves have sufficient information about prey location. To design a Grey Wolf Optimizer, Mirjalili et al. [75] have modelled the hunting strategies and the dominant leading characteristics of grey wolves in a mathematical manner. The mathematical model of the algorithm is described in next subsection.

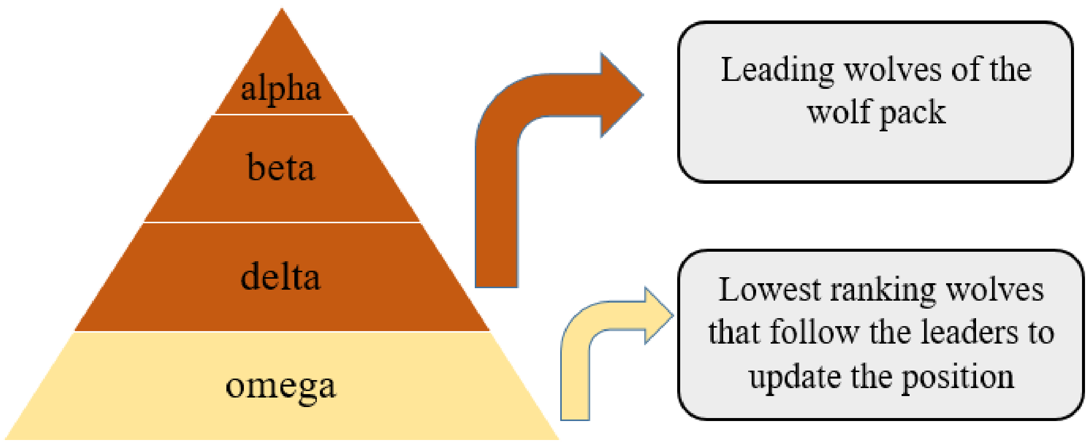


Fig 1.2. Leadership hierarchy of grey wolves

1.7.1. Mathematical Modeling of GWO

This section briefly explains the mathematical modeling of various hunting activities of grey wolves.

1.7.1.1. Leadership Behavior

Since the hunting process of grey wolves starts with searching prey, therefore, the leading wolves, alpha, beta and delta are selected for the hunting process according to the fitness of each individual wolf. In the optimization problem, the fittest solution is called as alpha, second and third best solution are assumed as beta and delta respectively and the remaining solutions of the problem are considered as omega. In GWO algorithm, all the omega wolves iteratively improve their locations with the guidance of leading wolves, alpha, beta and delta.

1.7.1.2. Encircling the Prey

In the classical GWO, it has been discussed that the wolves encircle the prey by the guidance of leading wolves alpha, beta and delta. To accomplish this, each wolf updates its position with the help of leading wolves by using the mathematical equations –

$$X_{t+1} = X_{p,t} - A \cdot D \tag{1.5}$$

where

$$D = |CX_{p,t} - X_t| \tag{1.6}$$

$$A = 2\,ar_1 - a \tag{1.7}$$

$$C = 2\, r_2 \tag{1.8}$$

In the above equations, X_t and X_{t+1} are the states of grey wolf at iteration t and $t + 1$ respectively. $X_{p,t}$ is the location of the prey at iteration t. A and C are coefficient vectors that acts as a exploration and exploitation operators during the search process of prey. a is a scalar quantity for a particular iteration which helps to the coefficient A to control the phase of exploration and exploitation and it decreases linearly from 2 to 0 as the iterations of the algorithm proceeds. r_1 and r_2 are the uniformly distributed random vectors and lie in the interval $(0, 1)$. The vector a can be defined by mathematical equation as follows

$$a = 2 - 2\left(\frac{t}{T}\right) \tag{1.9}$$

where T represents the maximum number of iterations which is fixed as a termination criteria for algorithm.

1.7.1.3. Hunting Behavior

In the classical GWO, it has been assumed that alpha, beta and delta wolves have sufficient information regarding the prey. Therefore, each wolf updates its position with the help of these leading wolves and for a particular iteration, hypothetically the prey position is presumed with the positions of alpha, beta and delta. To attack on prey, the following mathematical equations are proposed by Mirjalili et al. [75]

$$X_1 = X_{\alpha,t} - A_\alpha \cdot D_\alpha \tag{1.10}$$

$$X_2 = X_{\beta,t} - A_\beta \cdot D_\beta \tag{1.11}$$

$$X_3 = X_{\delta,t} - A_\delta \cdot D_\delta \tag{1.12}$$

where

$$D_\alpha = |C_\alpha X_{\alpha,t} - X_t| \tag{1.13}$$

$$D_\beta = |C_\beta X_{\beta,t} - X_t| \tag{1.14}$$

$$D_\delta = |C_\delta X_{\delta,t} - X_t| \tag{1.15}$$

$$X_{t+1} = \frac{X_1 + X_2 + X_3}{3} \tag{1.16}$$

and the values of $A_\alpha, A_\beta, A_\delta$ and $C_\alpha, C_\beta, C_\delta$ can be obtained with the help of equations (1.7) and (1.8).

In this way, a complete cycle of hunting process is performed by mimicking the leadership, encircling and hunting behavior of grey wolves and by repeating this cycle the optima for an optimization problem can be determined. The evolution process of wolf is shown in Fig 1.3.

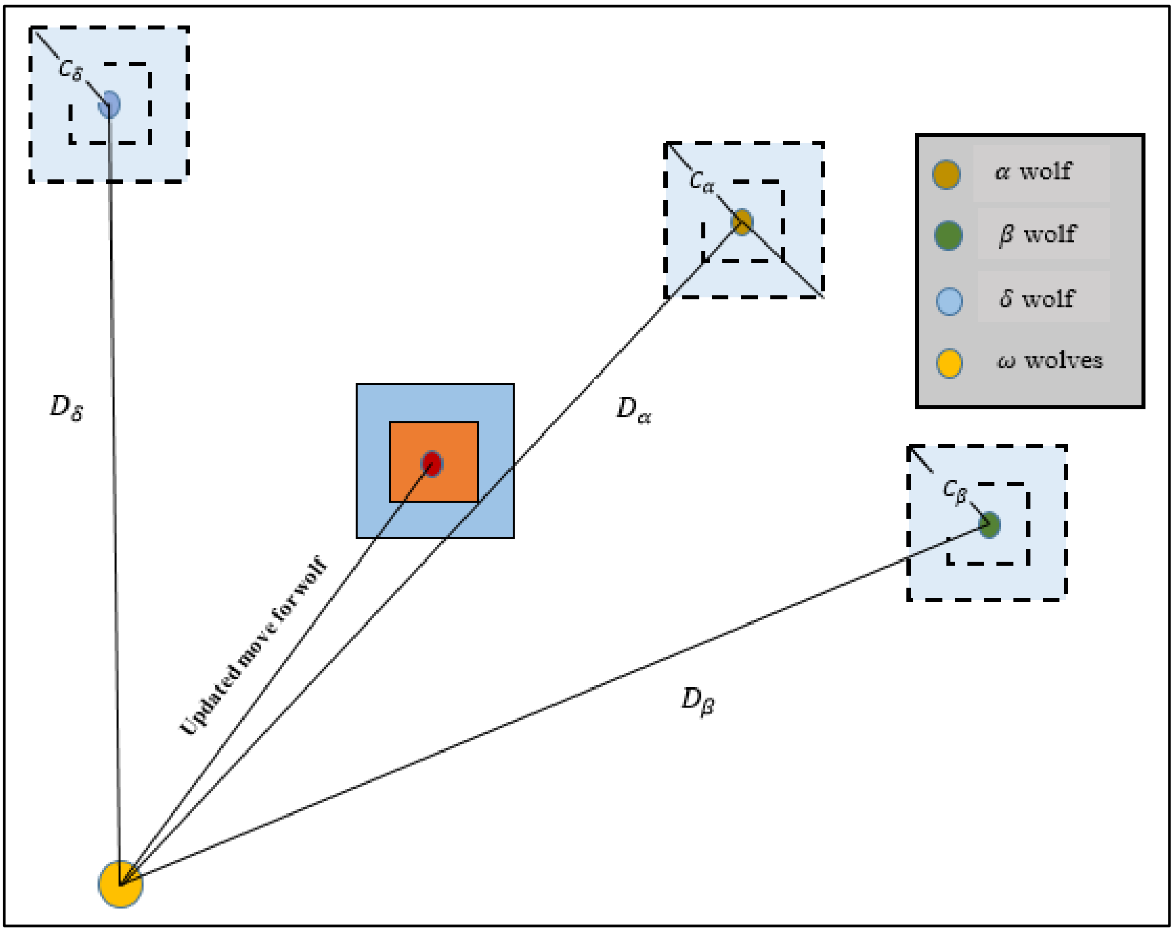


Fig 1.3. Evolution of position in GWO

Various positions in GWO, updated with the help of search equations are presented in Fig 1.4. In this figure (x, y) represents the wolf position and (x^*, y^*) represents the prey position. The step-wise description of the GWO is provided in Algorithm 1.1.

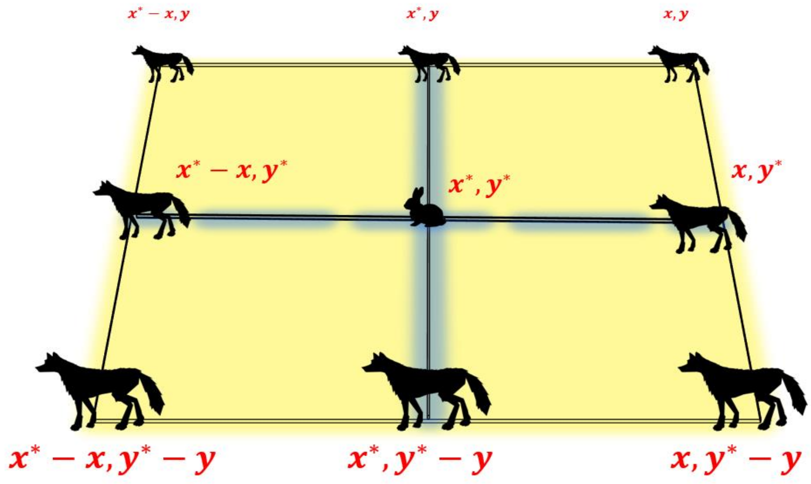


Fig 1.4. The 2-D representation of various possible positions

1.7.1.4. Exploration and Exploitation of Search Space in GWO Algorithm

The exploration and exploitation are two conflicting operators in any metaheuristic optimization algorithm [77]. In the phase of exploration new regions of a search space are discovered and in exploitation phase the potential of previously discovered search regions is analyzed. Therefore, an algorithm should be capable of addressing and balancing these two important operators to estimate the global optima of the problem.

In GWO algorithm the vectors A and C are introduced to address these two operators. When in algorithm $|A| < 1$ or $C < 1$, search regions are exploited and this situation represents the attack on prey. When $|A| > 1$ or $C > 1$, the new search regions are discovered and this situation represents the search behavior of grey wolves to find the prey. As in GWO algorithm, after the half of maximum number of iteration, $|A| < 1$ (as $|a| < 1, and\ A \in (-a, a)$), then in this case the exploration of a search space is performed by the vector C. In the GWO, a balance between exploration and exploitation is maintained with decreasing nature of the variable a. This variable helps to transit from the phase of exploration to exploitation.

Algorithm 1.1. Classical Grey Wolf Optimizer (GWO)

1. $For \quad Min\, F(X) \quad s.t. \quad X_{\min} \leq X \leq X_{\max},\ X = (x_1, x_2, \ldots, x_D) \in R^D$

2. $\boldsymbol{Initialize}$ $the\ grey\ wolf\ population\ \ X_i\ \ (i = 1,2, \ldots, N)$

3. **Evaluate** the fitness of each grey wolf

4. $\boldsymbol{Initialize}$ $the\ iteration\ count\ t = 0$

5. $\boldsymbol{Select}\ \ X_\alpha = fittest\ wolf\ of\ the\ pack$

$X_\beta = second\ best\ wolf$

$X_\delta = third\ best\ wolf$

6. $\boldsymbol{while}\ t < T, maximum\ number\ of\ iterations$

7. $\boldsymbol{for}$ each of the grey wolf

8. Update the state with the help of equation (1.16).

9. $\boldsymbol{end}$

10. Evaluate the fitness of each grey wolf

11. $\boldsymbol{update}$ $the\ leading\ wolves\ X_\alpha, X_\beta\ and\ X_\delta$

12. Update the coefficient a

13. $t = t + 1$

14. end

15. $return\ the\ alpha\ wolf$

1.7.2. Literature Review on GWO

In the literature, several attempts have been done to improve the search ability of the classical GWO so that the optimization problems can be solved more efficiently. These modifications can be categorized into following classes –

1. Improvement by updating the search mechanism of GWO

2. Improvement by introducing the new operators

3. Hybridization with other algorithms

4. Different encoding of individual solutions

1.7.2.1. Improvement by Updating the Search Mechanism of GWO

In the direction of improving the search ability in GWO, search equation and parameters of GWO are modified. For example: Mittal et al. [78] have modified the parameter a in order to maintain an appropriate balance between the operators exploration and exploitation. The proposed modified vector by Mittal et al. [78] can be represented as-

$$a = 2\left[1 - \left(\frac{t}{T}\right)^2\right] \tag{1.17}$$

In [79], the values of vector a is chosen adaptively to maintain an appropriate balance between exploration and exploitation. The adapted values of a can be obtained as follows

$$a = \left[\frac{1 - \left(\frac{t}{T}\right)}{1 - \mu\left(\frac{t}{T}\right)}\right] \tag{1.18}$$

where μ is non-linear modulation index and t represent the current iteration and T stands for maximum number of iterations which is predefined as termination criteria for the algorithm. Malik et al. [80] have proposed a different scheme to approximate the updated position of a current wolf with the help of positions which are obtained from leading wolves of the pack. In this strategy, a weighted average is used instead of taking a simple arithmetic mean. This proposed algorithm performs better as compared to classical GWO on multimodal optimization problems. In [81, 82], Levy-flight search strategy is employed to enhance the search-efficiency of wolf pack. In [83], grouped GWO has been introduced to enhance the global search ability and employed for maximum power point tracking of doubly-fed induction generator. To adopt the parameters of GWO fuzzy logic is utilized in [84, 85]. In [86], improved variant of GWO is proposed called Experienced Grey Wolf Optimization which uses the reinforcement learning for the parameter adaptation and neural network for the adaptation of exploration rate of each wolf. In [87], the parameters of GWO are modified to enhance the search ability of wolves. In [88], the search strategy of GWO is modified by inspiring from PSO and applied the proposed algorithm to solve large-scale optimization problems. Three novel improved variants of GWO based on the concept of astrophysics and prey weight are developed in [89]. In [90], the concept of cellular automata is embedded into GWO to enhance the diversity of wolves in GWO. In [91], the hunting search strategy of GWO is modified to enhance the exploration during the search. The position update equation of GWO is modified in [92] by introducing the contribution of omega wolves.

1.7.2.2. Improvement by Introducing the New Operators

In the direction of improving algorithms, various operators like genetic operators (crossover and mutation) and local search operators (chaotic local search, fuzzy hierarchical) are integrated to enhance the search ability of grey wolves. In [93], different fuzzy hierarchical operators such as centroid and weighted difference are integrated in the search equation of GWO algorithm to force the enhancement of contribution of leading wolves in descending order according to their fitness to propose different versions of classical GWO. In [94], evolutionary population dynamics has been applied to discard the worst fitted wolf from the pack and a new wolf with the help of EPD operator is introduced. In [95, 96], the crossover and mutation operators are introduced in the GWO to enhance its performance. In [97, 98], the concept of opposition-based learning is introduced to avoid the problem of stagnation at local optima. In order to accelerate the convergence rate, chaos theory is integrated in the GWO [99-101]. In [102], the binary crossover and levy-flight distributed random steps are employed to update the wolves in the GWO. In [103], an adaptive bridging mechanism based on β-chaotic sequence is introduced to improve the search strategy of GWO. Various selection methods and their behavior is studied on GWO [104]. In [105], the operator called refraction learning inspired by the principle of light refraction in physics is introduced in the classical GWO to propose a modified variant which can avoid the issue of stagnation at local optima. In [106], a boosted GWO is proposed which utilizes the concept of levy-flight search, opposition-based learning, random spiral-form motions and random leaders to enhance the capability of wolves in terms of exploration and exploitation.

1.7.2.3. Hybridization with Other Algorithms

Generally, hybridization refers to combine two or more search algorithm in order to utilize the impressive characteristics and advantageous of different algorithms. In this direction, in the literature, GWO has been hybridized with various metaheuristics. For example – In [107], Tawhid and Ali have hybridized classical GWO with GA to minimize the potential energy of molecule. This problems consists of many local minima that increases exponentially with the dimension. In [108], the GA and GWO are hybridized to solve the large-scale global optimization problems. In [109-111], a hybridized version of DE and GWO has been proposed to solve the global optimization problem. In [112, 113], GWO and PSO are hybridized to improve the convergence rate. The obtained solutions by this hybridized algorithm are compared with other metaheuristic algorithms. In [114], the GWO is hybridized with SCA to enhance the exploration in GWO. In [115], BBO and GWO are hybridized to enhance the synergy between to different algorithms. In this hybridization, first the BBO is improved by combining the differential mutation and multimigration operators and secondly, GWO is improved by opposition-based learning concept. For the optimal selection of parameters in GWO, Cuckoo Search (CS) algorithm is combined with GWO [116]. In [117], the opposition-based learning and disruption operators are merged in the GWO and the proposed method is hybridized with DE to enhance the global and local search ability of GWO. Gaidhane and Nigam [118] have hybridized the GWO with ABC algorithm to boost up the exploration strength of wolves in the GWO. In [119], GWO is hybridized with Firework algorithm to balance the exploration and exploitation. In [120], the GWO is hybridized with Firefly Algorithm to enhance the diversity in GWO and to avoid the stagnation at local optima.

1.7.2.4. Different Encoding of Individual Solutions

Luo et al. [121] have used the complex valued encoding for the individual wolf with a suggestion that this encoding of wolf can enhance the information strength of the wolves and diversity of the wolf pack. To verify this concept, the comparison is performed with classical GWO, GGSA and ABC on several benchmark test problems.

1.7.2.5. Other Variants of GWO

To solve the problem of multi-criteria optimization multi-objective Grey Wolf Optimizer [122, 123] is designed. In [124], binary version of GWO is proposed using two different approaches.

1.7.3. Applications of GWO

Due to the impressive advantageous of GWO in terms of exploration and exploitation, it has been applied to solve various application problems in different research domains. As the list of the applications of GWO is too large, therefore in this section some of the important and recent applications are listed as: Training of Multilayer perceptron [125], Parameter estimation in surface waves [126], Two-stage assembly flow shop scheduling [127], Optimal control of DC motor [128], Optimal Power flow [129], Training of q-Gaussian radial basis functional-link nets [130], Non-convex economic load dispatch [131], maximum power point tracking of doubly-fed induction generator based wind turbine [83], Placement and sizing of multiple distributed generation [132], Feature selection [133, 134], Unmanned combat aerial vehicle path planning [135], Dynamic scheduling in welding industry [136], Multilevel thresholding for image segmentation [137], Minimization of potential energy [107], Load frequency control of interconnected power system [138], Unit commitment problem [112], Inversion of geoelectrical data [139], Template matching [140], Hyperspectral band selection [141], Short-term unit consignment [142], Optimal reactive power dispatch [143]. Detailed literature on the GWO can be accessed from [144].

1.8. Sine Cosine Algorithm (SCA)

1.8.1. Mathematical Modeling of SCA

The Sine Cosine Algorithm (SCA) is a recently developed metaheuristic algorithm based on the mathematical characteristics of sine and cosine trigonometric functions. This algorithm was designed by Mirjalili in 2015 [145]. Like other population-based metaheuristic optimization algorithms, SCA also starts with a set of randomly distributed solutions, then each candidate solution updates their position with the help of following equations –

$$X_{i,t+1} = X_{i,t} + A\sin(r_1)\,|CX_\alpha - X_{i,t}| \tag{1.19}$$

$$X_{i,t+1} = X_{i,t} + A\cos(r_1)\,|CX_\alpha - X_{i,t}| \tag{1.20}$$

The above two equations are used in SCA in a following manner

$$X_{i,t+1} = \begin{cases} X_{i,t} + A\sin(r_1)\,\left|CX_\alpha - X_{i,t}\right| & if\ r < 0.5 \\ X_{i,t} + A\cos(r_1)\,\left|CX_\alpha - X_{i,t}\right| & otherwise \end{cases} \tag{1.21}$$

where $X_{i,t}$ and $X_{i,t+1}$ represents the i^{th} solution vector at t^{th} and $(t+1)^{th}$ iteration respectively. X_α is the fittest solution in the solution set, r is a uniformly distributed random number in the interval (0, 1) and r_1 is a random number in the interval $(0, 2\pi)$ and decides the direction of moment of current candidate solution which can be either towards the X_α or outside X_α. The vector C provides a weight to X_α which emphasizes on exploration ($C > 1$) and exploitation ($C < 1$). The vector C also helps in avoiding the premature convergence at the end of iterations. The vector r helps in transition from sine to cosine functions and vice versa. The effect of random number A on sine and cosine function is shown in Fig 1.5. The effect of random number r_1 on the position of candidate solutions is shown in Fig 1.6.

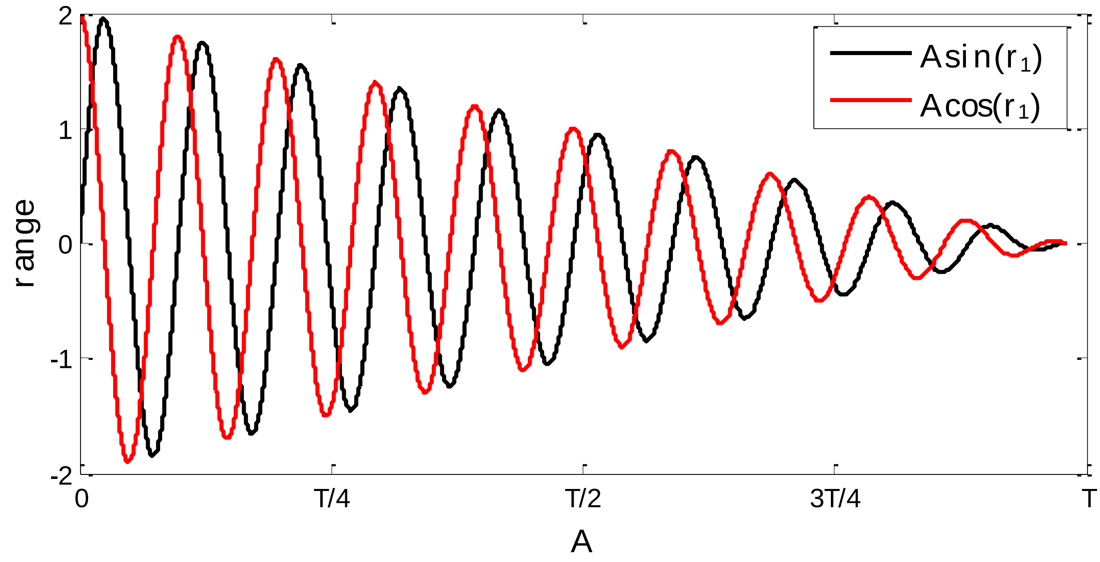


Fig 1.5. The impact of sine and cosine functions with coefficient A

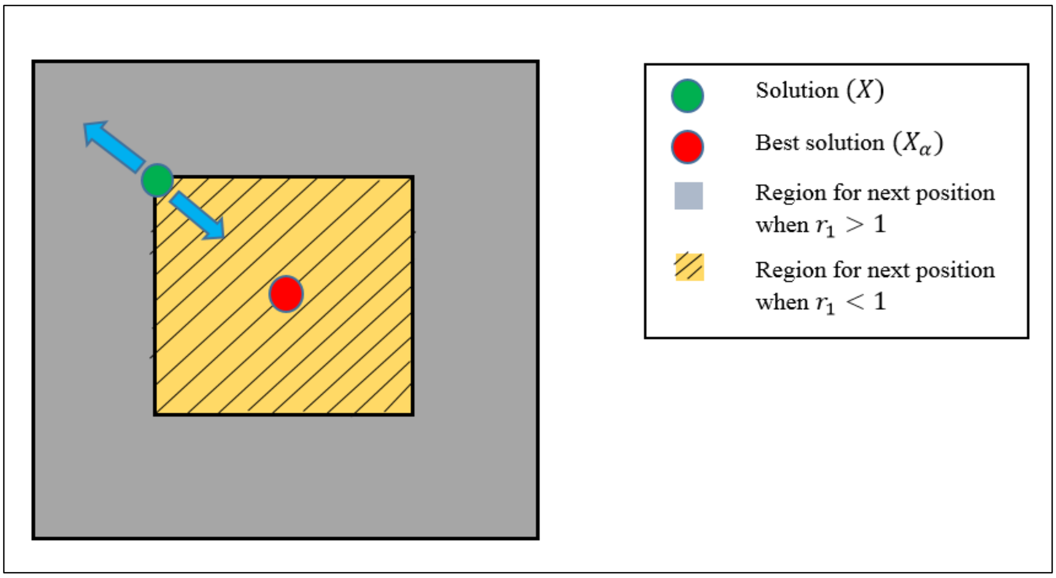


Fig 1.6. Effect of the parameter r_1 in updating the position of candidate solutions

The parameter A is a random number which decides the area of the search space around the current candidate solution. This region of search space may lie inside X_α and $X_{i,t}$ or outside them. The parameter A also helps in exploration and exploitation of a search space as well as in maintaining a suitable balance between them. In the first half of the total number of iterations, coefficient A contributes in the exploration of a search space while in the second half of the total number of iterations, it is devoted to the exploitation of the search space. Mathematically, the parameter A can be defined as follows –

$$A = 2 - 2\left(\frac{t}{T}\right) \tag{1.22}$$

where T represents the maximum number of iterations which is predefined as the termination criteria for SCA. The steps of Sine Cosine Algorithm are presented in Algorithm 1.2.

Algorithm 1.2. Classical Sine Cosine Algorithm (SCA)

---

1. $For \quad Min\, F(X) \quad s.t. \quad X_{\min} \leq X \leq X_{\max},\ X = (x_1, x_2, \dots, x_D) \in R^D$
2. **Initialize** the population of candidate solutions $X_i \quad (i = 1, 2, \dots, N)$
3. **Evaluate** the fitness of each candidate solution
4. **Select** the best solution X_α from the population of candidate solutions
5. Initialize the iteration count $t = 0$
6. **while** $t < T$
7. Update each solution vector with the help of equation (1.21).
8. Compute the fitness of each updated candidate solution
9. Update the best solution X_α
10. Update the coefficient A
11. $t = t + 1$
12. end of while
13. Return the best solution X_α.

Although the sine cosine algorithm is efficient to explore the search space but in many cases, it suffers from some major difficulties like skipping of true solutions and local optima stagnation and therefore, an improvement is required in the search strategy of classical SCA.

1.8.2. Literature Review on SCA

In the literature, several attempts have been done to improve the search ability of classical SCA. These attempts can be categorized into following classes –

1. Improvement by updating the search mechanism of SCA
2. Improvement by introducing the new operators
3. Hybridization with other algorithms
4. Other variants of SCA

1.8.2.1. Improvement by Updating the Search Mechanism of SCA

The search mechanism of classical SCA is updated and elitism strategy is employed during the search into SCA to enhance its performance [146]. In order to improve the search performance of SCA, a novel weighted update position mechanism is introduced in [147]. In [148], a modified position update mechanism with nonlinear decreasing conversion parameter strategy is introduced for SCA to solve large-dimensional optimization problems.

1.8.2.2. Improvement by Introducing the New Operators

In [149], multi-orthogonal search strategy is employed in SCA to enhance the exploration and exploitation strength of candidate solutions. The concept of opposition-based learning is employed in the SCA [150, 151] to prevent the candidate solutions from stagnation at local optima. In [152], the backtracking search strategy is employed in SCA to use its merits in improving the search ability of SCA. Levy-flight search strategy is embedded into SCA to enhance its global and local search efficiency [153]. A modified version of SCA, based on neighbourhood search and greedy levy mutation, is developed in [154] to improve the solution accuracy and convergence rate. In [155], a modified SCA is introduced based on the Riesz fractional derivative mutation. In this algorithms, the population is initialized with the help of quasi-opposition learning strategy to enhance the exploration ability of candidate solutions. In [156], Gaussian local search and random mutation is used to enhance the diversity during the search and convergence rate. In [157], cloud model based Sine Cosine Algorithm is introduced to adjust the control parameter adaptively while keeping SCA algorithm framework unchanged.

1.8.2.3. Hybridization with Other Algorithms

In order to escape from local optima and to improve the convergence rate, Nenavath and Jatoth [158] have hybridized the SCA with DE. To enhance in the exploitation skills of candidate solutions in SCA, it has been hybridized with GWO by Singh and Singh [114]. To overcome the issue of premature convergence, a hybrid version of SCA and PSO is developed in [159, 160]. In [161], hybrid version of SCA and TLBO is developed to propose a better capability of escaping from local optima and to improve the convergence rate of both the algorithms. In [162], SCA and Brain Storm Optimization Algorithm are hybridized to balance the exploration and exploitation in the SCA.

1.8.2.4. Other Variants of SCA

The multi-objective SCA is proposed by Tawhid and Savsani [163] for multi-criteria optimization. In [164], a binary variant of SCA is also developed to solve the binary optimization problems.

1.8.3. Applications of SCA

The SCA has been applied to solve several various application problems in different research domains. Some of the real world applications where SCA is applied are Feature selection [165], wind speed forecasting [166], training feedforward neural networks [167], handwritten arabic manuscript image binarization [168], optimal power flow [169], Thermal and economical optimization of a shell and tube evaporator [152], short-term hydrothermal scheduling [170], Data clustering [171], Context based image segmentation [172], Re-entry trajectory optimization for space shuttle [173], optimal design of hybrid power generation systems [174], economic load dispatch problem [175], reduction of higher order continuous systems [176], Breast Cancer Classification [177], Pairwise Global Sequence Alignment [178], Optimal Camera Placement [179], Peak operation problem of cascade hydropower reservoirs [156], Design of PID controllers [180], Load frequency control of power system [181] and so on. A literature review on SCA is also presented in [182].

1.9. Motivation and Objectives of the Thesis

The efficiency of any metaheuristic algorithm depends on the operators exploration and exploitation and an appropriate balance between them. An ideal metaheuristic algorithm should have efficient ability to explore the search space in the beginning of algorithm and exploitation at the end of generations of algorithm. In order to establish the balance between exploration and exploitation, some attempts have been done in the Thesis by proposing the variants of Grey Wolf Optimizer and Sine Cosine Algorithm.

The Thesis focuses on possible improvement in the search strategy of both the algorithms GWO and SCA. The objective of the Thesis are

1. Enhance the search ability of the classical GWO and classical SCA by developing their efficient and reliable modified variants.
2. Investigate the performance of proposed variants of GWO and SCA on standard unconstrained and constraint benchmark optimization problems.
3. Implementation of proposed variants of GWO and SCA on real-world application problems.

1.10. Organization of the Thesis

The current chapter follows the literature survey of GWO and SCA and their applications. The summary of other chapters is given below:

Chapter 2 introduces the Novel Random Walk Grey Wolf Optimizer (RW-GWO) which is the modified version of GWO by enhancing the search ability of leading hunters in GWO to provide better guidance of search as compared to the classical GWO. The proposed algorithm is analyzed and compared with classical GWO on unconstrained IEEE CEC 2014 benchmark test problems. The analysis of numerical results demonstrate the superiority of the proposed RW-GWO algorithm in terms of accuracy.

Chapter 3 introduces the Memory-based Grey Wolf Optimizer (mGWO) which utilizes the best memory of each wolves to update their states. In this algorithm, the exploitation is also improved around the best wolf. The proposed algorithm is analyzed and compared with classical GWO on unconstrained IEEE CEC 2014 benchmark test problems. The analysis of numerical results demonstrate the superiority of the proposed mGWO algorithm in terms of solution accuracy as compared to the classical GWO.

Chapter 4 presents the improved variant of SCA called m-SCA which utilizes the concept of opposition-based learning to prevent from stagnation. In the m-SCA position update mechanism is also modified to enhance the search-efficiency. The proposed algorithm is tested on unconstrained CEC 2014 benchmark set and experimentation analysis ensure the superiority of the proposed algorithm as compared to classical SCA in terms of solution accuracy.

In **Chapter 5**, the improved version of classical SCA called ISCA is presented which modifies the search strategy of candidate solutions by the crossover operator and personal best state of each candidate solutions. The validation and comparison of the proposed algorithm with classical SCA is performed on CEC 2014 unconstrained benchmark set. The analysis of the results and comparison with classical SCA demonstrate the better search accuracy of the ISCA as compared to classical SCA.

In **Chapter 6**, the performances of classical GWO, classical SCA, and their proposed variants which are presented in chapters 2, 3, 4, and 5 are evaluated on IEEE CEC 2006 constraint benchmark problem set. The analysis is done based on the criteria provided by IEEE CEC 2006. The comparison among the proposed variants, classical GWO and classical SCA shows the better search efficiency of the proposed variants of GWO and SCA.

Chapter 7 applies the classical GWO, SCA, and their proposed variants which are presented in Chapters 2, 3, 4, and 5 on multilevel thresholding problems. The standard benchmark test images are taken for experimentation. The analysis of the results demonstrates the enhanced search-efficiency of the proposed variants of GWO and SCA to determine the optimal thresholds.

In **Chapter 8**, the problem of determining the optimal coordination of directional overcurrent relays is solved using classical GWO, classical SCA, and their proposed variants which are presented in Chapters 2, 3, 4, and 5. In this study, the IEEE 3, 4, 6 and 14-bus systems are used as test models. The numerical results and their analysis verifies the superior performance of proposed RW-GWO algorithm in finding the optimal setting for directional overcurrent relays.

Chapter 9 concludes the Thesis with overall developments in the proposed algorithms with their future scope in other real-life applications and some future recommendations are discussed.

There are appendix in the Thesis, as mentioned below

1. Appendix A Unconstrained Test Problems
2. Appendix B Constrained Test Problems
3. Appendix C Data set corresponding to various bus-systems for the relay coordination problem.

Chapter 2

A Novel Random Walk Grey Wolf Optimizer for Unconstrained Optimization Problems

In this chapter an attempt is made to improve the search efficiency of grey wolves by proposing a new variant of Grey Wolf Optimizer.

2.1. Introduction

The literature review on GWO presented in Chapter 1 shows that in some cases, the classical GWO faces the situation of stagnation at local optima, slow convergence and insufficient balance between exploration and exploitation. In the present chapter, one major drawback of insufficient guidance in a wolf pack is pointed out and a novel variant of GWO called Random Walk Grey Wolf Optimizer (RW-GWO) is introduced. In the RW-GWO, a random walk based search strategy is applied to update the position of leading hunters called alpha, beta, and delta. The performance of the proposed RW-GWO algorithm is tested on an unconstrained benchmark problem set given in IEEE CEC 2014 and the results are analyzed and compared with classical GWO.

The organization of the chapter is as follows: Section 2.2 provides a motivation behind proposing a new variant of classical GWO and detailed description of the proposed RW-GWO. Section 2.3 provides numerical experimentation, analysis and comparison with classical GWO. Finally, the chapter is closed with concluding remarks in Section 2.4.

2.2. Proposed Random Walk Grey Wolf Optimizer (RW-GWO)

2.2.1. Motivation

Since, the classical GWO is based on the leadership behavior of grey wolves, therefore, the leading wolves are responsible and liable agents of the pack to update the state of each wolf and to provide promising directions of search. Therefore, it is very important that in each iteration, these leading wolves should be the best (in terms of fitness) so that each wolf can update their state with better guidance.

The content of this chapter is published in:

Gupta, S., & Deep, K. (2019). A novel random walk grey wolf optimizer. Swarm and Evolutionary Computation, 44, 101-112. Elsevier.

The classical search equation of GWO shows that each wolf updates its state with the help of leading hunters called alpha (α), beta (β) and delta (δ). But, then the natural question arises that how the states of the leading hunters should be updated, because these wolves have been elected as dominant wolves of the pack? The second question is why alpha wolf should take the guidance of low fitted (inferior) wolves beta and delta of the pack to update its position? Similarly, why beta wolf should update its state with the help of low fitted (inferior) wolf delta? This is the main shortcoming that has been observed in the search mechanism of classical GWO and this may be the reason that the wolf pack trapped in local optima. Therefore, the selection of leading hunters during the search process is very crucial task and some improvisation is needed in the search mechanism of classical GWO for the leading hunters. In this direction, to alleviate from all the issues mentioned above, the present chapter proposes a novel Random Walk Grey Wolf Optimizer (RW-GWO) based on the cauchy random walk to update the leading hunters of the pack.

2.2.2. Cauchy Distribution

Cauchy distribution (Lorentz distribution) is a continuous probability distribution [183], with parameters x_0 and γ. The parameter x_0 is a positive real number termed as location parameter and γ stands for scaling parameter which tells about the shape of the distribution. Less value of γ shows the shape of a distribution with parochial width and high peak. In contrast, the higher value of γ shows the shape with a broad width and lower peak. The probability density function of the distribution is given by

$$f(x, x_0, \gamma) = \frac{1}{\pi\gamma\left[1+\left(\frac{x-x_0}{\gamma}\right)^2\right]}, \quad x \in (-\infty, \infty) \tag{2.1}$$

and the cumulative distribution function of cauchy distribution is given as

$$F(x, x_0, \gamma) = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{x - x_0}{\gamma}\right) \tag{2.2}$$

2.2.3. Cauchy Random Walk Based Strategy for the Leading Hunters

In order to accelerate the search process and to provide an explorative guidance for the leading hunters alpha, beta and delta, a random walk search strategy [184] is utilized. The proposed random walk search mechanism is applied for the leading hunters only so that the more promising and explorative guidance can be discovered for the wolf pack. The pattern of the steps in a random walk is chosen based on the cauchy distribution [183]. As an example, the cauchy distributed random numbers over 100 iterations are shown in Fig 2.1. The reason for chosing cauchy distribution is its infinite variance. Because of infinite variantce, it occasionally generates high values. It is hoped that higher values may help the pack to escape from local optima, and the lower values produced by the distribution may help in exploiting the search space.

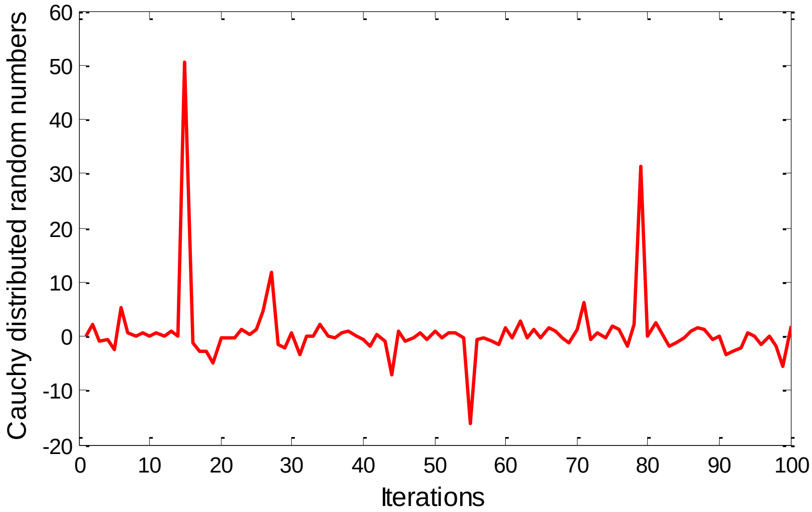


Fig 2.1. Distribution of cauchy random numbers over 100 iterations

The proposed search equation for leading hunters can be expressed as follows:

$$X_{iL,t+1} = X_{iL,t} + \alpha_{L,t} \times S_{L,t} \tag{2.3}$$

where $x_{iL,t}$ represent the position of the i^{th} leading wolf ($i = 1$, represent alpha, $i = 2$ represent beta and $i = 3$ represents delta wolf), S_L is the step size drawn from the cauchy distribution. The variable $\alpha_{L,t}$, which controls the step length $S_{L,t}$ at iteration t, is linearly decreased from the value 2 to 0 over the course of iterations. This selection provides a transition from exploration to the exploitation phase and can be formulated as follows:

$$\alpha_L = 2\left(1 - \frac{t}{T}\right) \tag{2.4}$$

where t is a current iteration, and T is the maximum number of iterations which are predefined for the algorithm.

Higher values of step length is very effective in sudden jump at the time of stagnation at local optima and helps the leading wolves to explore the search space more efficiently. Small values of step length help in exploiting the search regions around the leading wolves. Thus, the cauchy distributed random walk maintains a balance between exploration and exploitation. At the end of each iteration, greedy selection mechanism is employed corresponding to each wolf between its current and previous states. The greedy selection also helps to avoid the divergence of wolves from discovered promising search regions. In the RW-GWO, it can be noticed that any extra function evaluation is not added in the algorithm and therefore, the number of function evaluations remain same in the RW-GWO as in classical GWO.

The pseudo code of the proposed RW-GWO algorithm is described in Algorithm 2.1.

Algorithm 2.1: A Novel Random Walk Grey Wolf Optimizer (RW-GWO)

---

1. $For \quad Min\, F(X) \quad s.t. \quad X_{\min} \leq X \leq X_{\max},\ X = (x_1, x_2, \dots, x_D) \in R^D$

2. $\boldsymbol{Initialize}$ $the\ grey\ wolf\ population\ X_i\ (i = 1,2,\dots,N)$

3. $\boldsymbol{Evaluate}$ $the\ fitness\ of\ each\ wolf$

4. $\boldsymbol{Initialize}$ $the\ iteration\ count\ t = 0$

5. $\boldsymbol{Select}\ \ X_\alpha = fittest\ wolf\ of\ the\ pack$

$X_\beta = second\ best\ wolf$

$X_\delta = third\ best\ wolf$

6. $\boldsymbol{while}\ t < T, maximum\ number\ of\ iterations$

7. $\boldsymbol{for}\ each\ leading\ wolf\ alpha, beta\ and\ delta$

8. $find\ new\ position\ X_{iL}\ for\ the\ leaders\ X_i\ \ using\ eq.(2.3)$

9. $\boldsymbol{end\ for}$

10. $\boldsymbol{for}\ each\ omega\ wolf$

11. $update\ the\ position\ by\ classical\ search\ eq.(1.16)\ of\ GWO$

12. $\boldsymbol{end\ for}$

13. $\boldsymbol{Evaluate}\ the\ fitness\ of\ each\ wolf$

14. $\boldsymbol{for}\ i = 1,2,\dots,N$

15. $\boldsymbol{if}\ F\left(X_{i,t+1}\right) \geq F(X_{i,t})$

16. $X_{i,t+1} = X_{i,t}$

17. $\boldsymbol{end\ if}$

18. $\boldsymbol{end\ for}$

19. $\boldsymbol{update}\ the\ leading\ wolves\ X_\alpha, X_\beta\ and\ X_\delta$

20. $t = t + 1$

21. $\boldsymbol{end\ while}$

22. $return\ the\ alpha\ wolf$

2.2.4. Computational Complexity

Since the computational complexity plays an important role to analyze the complexity of algorithms and the user always prefers less complex algorithm. Therefore, in the present section, the worst time complexity of both the algorithms classical GWO and proposed RW-GWO is calculated in terms of big$-O$ notation using their pseudo codes. The step-wise description of the obtained complexities of classical GWO and the RW-GWO is as follows:

For classical GWO:

1. The classical GWO initializes the wolf pack in $O(N \times D)$ time, where N is the size of pack and D represent the dimension of the problem.
2. Fitness evaluation of wolves requires $O(N)$ time.
3. Selection of leading hunters in classical GWO requires $O(N)$ time.
4. Position update process in the classical GWO requires the $O(N \times D)$ time.

In summary, the total computational time for the classical GWO is equal to $O(N \times D \times T)$ for maximum number of iterations T.

For RW-GWO:

1. The RW-GWO initializes the wolf pack in $O(N \times D)$ time, where N is the size of pack and D represent the dimension of the problem.
2. Fitness evaluation of wolves requires $O(N)$ time.
3. Selection of leading hunters in RW-GWO requires $O(N)$ time.
4. Position update process for leading hunters using random walk search mechanism consumes $O(N \times D)$ time.
5. Position update process for omega wolves using classical search equation of GWO requires $O(N \times D)$ time.

In summary, the total computational time for the RW-GWO is equal to $O(N \times D \times T)$ for maximum number of iterations T. Hence, by comparing the complexities of the classical GWO and RW-GWO, it can be concluded that in terms of computational complexity both the algorithms are same.

2.3. Numerical Experiments and Analysis of Results

2.3.1. Benchmark Functions and Parameter Setting

In this section, the performance of classical GWO and the proposed RW-GWO algorithm is studied on the basis of benchmark problems given in IEEE CEC 2014 [185]. The CEC 2014 benchmark set consists of 30 unconstrained optimization problems from F1 to F30 in which the problems from F1 to F3 are categorized as unimodal, F4 to F16 are multimodal, F17 to F22 are hybrid and F23 to F30 the problems are composite. As per the guidelines provided by IEEE CEC 2014, 51 independent runs are to be performed for each test problem to observe the performance of both the algorithms. The search space for each variable is fixed to [-100, 100], and the termination criteria is set to $(10^D \times D)$ function evaluations where D represent the dimension of the problem. These problems are presented in Appendix A of this Thesis and all the algorithms proposed in this Thesis are investigated on this problem set.

2.3.2. Analysis of the Results

In this section, the numerical results obtained by implementing classical GWO and RW-GWO on IEEE CEC 2014 [185] benchmark problems are provided. The results are presented in the form of absolute error in objective function value. The better results are highlighted in bold face. For a feasible solution X and optima X^* to the problem F, the absolute error is calculated by $|F(X) - F(X^*)|$. The experiments are performed on 10 and 30-dimensional problems and various criteria, such as minimum, median, mean, maximum, standard deviation (STD) of the absolute errors in objective function values of test problems are presented. The performance of the RW-GWO on different categories of benchmarks corresponding to 10 and 30-dimensonal problems is analyzed as follows:

The results for 10 dimension

The results for 10-dimension problems are given in Table 2.1. From this table, it can be observed that:

In all of the 10-dimensional unimodal problems from F1 to F3, the RW-GWO algorithm outperforms classical GWO except for the problem F2. The RW-GWO provides a better minimum, maximum, mean, median and standard deviation value of error in objective functions in problems F1 and F3. In problem F2, except for the median value, the RW-GWO outperforms GWO in all other criteria. Thus, the experimental results demonstrate the better search ability of RW-GWO as compared to classical GWO in terms of solution accuracy.

In 10-dimensional multimodal problems, the RW-GWO outperforms classical GWO in all criteria for the problems F4, F8-F10 and F13-F15. In the problems F5 and F6, the RW-GWO performs better than classical GWO in terms of all criteria except standard deviation. In the problem F7, the RW-GWO provides better results than classical GWO in terms of all the criteria except median value. In F11, the classical GWO is better in providing the mean and median error, and except for these criteria, the RW-GWO is better than classical GWO. In the problem F12, the RW-GWO is better than classical GWO except for minimum error. In F16, except for maximum and standard deviation, the RW-GWO is better than classical GWO.

In 10-dimensional hybrid problems F17, F19-F21, the RW-GWO provides better results in all the criteria as compared to the classical GWO. In problem F18, except for minimum error and in F22, except for standard deviation, the RW-GWO provides better results as compared to classical GWO.

For 10-dimensional composite problems F23, F24, F29 and F30, the RW-GWO provides better results as compared to classical GWO in all the criteria. In problems F25 and F27, the RW-GWO is better than the classical GWO except for maximum and standard deviation value of errors. In F26, both the algorithms classical GWO and RW-GWO provide same results in terms of all the criteria except for the standard deviation. The standard deviation is better in the RW-GWO than the classical GWO for this problem. In F28, except for the minimum error the RW-GWO provides better results as compared to classical GWO.

Thus, it is concluded that the RW-GWO is better than the classical GWO for 10-dimensional problems.

The results for 30 dimension

The results for 30-dimension problems are given in Table 2.2. From this table, it can be observed that:

In all of the 30-dimensional unimodal problems from F1 to F3, the RW-GWO algorithm outperforms classical GWO. The RW-GWO provides a better minimum, maximum, mean, median and standard deviation value of error in objective functions in all these problems as compared to the classical GWO.

In 30-dimensional multimodal problems, the RW-GWO outperforms classical GWO in all criteria for F4, F7-F10, F12 and F13. In the problems F5, F6 and F16, the RW-GWO performs better than classical GWO in terms of all criteria except standard deviation. In problems F11, F14 and F15 except for minimum error, the RW-GWO provides better results as compared to classical GWO. In

In 30-dimensional hybrid problems F18-F20, the RW-GWO provides better results in all the criteria as compared to the classical GWO. In problem F17, except for minimum and median error the RW-GWO is better than classical GWO. In F21, except for median error, the RW-GWO is better than classical GWO. In F22, except for minimum error and standard deviation value of errors, the RW-GWO provides better results as compared to classical GWO.

For 30-dimensional composite problems (F23-F30), the RW-GWO provides better results as compared to classical GWO except for standard deviation value in F24.

Thus, it is concluded that the RW-GWO is better than the classical GWO for 30-dimensional problems.

The unimodal problems are used for the evaluation of the exploitation strength of any search algorithm. Therefore, the RW-GWO algorithm is better in terms of exploiting the search space as it outperformed the classical GWO for unimodal problems. By analyzing the performance of the RW-GWO on multimodal problems, it is found that in terms of exploration ability, the RW-GWO demonstrates its superior search-ability as compared to classical GWO. The RW-GWO algorithm outperformed classical GWO for hybrid and composite problems, and this verifies its superior ability of balancing the exploration and exploitation as compared to the classical GWO.

2.3.3. Statistical Analysis

Although, the analysis of results presented in Section 2.3.2 demonstrate the better performance of the RW-GWO as compared to classical GWO, but in order to make concrete conclusions about the significance of differences in the performance of algorithms, Wilcoxon signed rank test [186] has been performed. The statistical results also demonstrate that the better results obtained by the RW-GWO as compared to classical GWO are not just by chance. The test has been conducted with 5% level of significance and the statistical results are presented in Tables 2.3 and 2.4 for 10 and 30-dimensional CEC 2014 benchmark problems respectively. In these tables '+' indicates the statistically better performance of RW-GWO as compared to classical GWO, '-' indicates that classical GWO performs better than RW-GWO and '=' is used to represent the equivalent performance of both the algorithms. From Table 2.3, it can be seen that out of 30 problems, RW-GWO is better than classical GWO in 22 problems, RW-GWO is inferior to classical GWO in 1 problem, and both are equal in 7 problems. Similarly, from Table 2.4, it can be seen that out of 30 problems RW-GWO is better than classical GWO in 26 problems, RW-GWO is inferior to classical GWO in 1 problem, and both are equal in 3 problems. Thus, the statistical comparison demonstrate the superior performance of the RW-GWO as compared to classical GWO.

2.3.4. Convergence Behavior

Since the fittest solution in each iteration of an algorithm is represented by alpha, therefore, in order to analyze the convergence behavior, median values of alpha solutions are plotted for 30-dimensional problems of IEEE CEC 2014. The convergence curves are shown in Figs 2.2 to 2.5 for 30-dimensional problems. In these figures, the horizontal axis indicates the number of iterations and the vertical axis represents the objective function values (or fitness function value) of the test functions. From the convergence curves, it can be observed that in terms of convergence rate, the proposed RW-GWO algorithm is better than the classical GWO.

2.4. Concluding Remarks

In this chapter, a novel variant of GWO called Random Walk Grey Wolf Optimizer (RW-GWO) is introduced to solve unconstrained global optimization problems. In the RW-GWO, a random walk strategy is applied to update the leading hunters of the pack. The step length of the random walk is drawn from the cauchy distribution which helps to escape from local optima by producing large steps for the random walk. The applied greedy selection mechanism between two consecutive iterations of algorithm maintains the balance between exploration and exploitation and avoids the divergence of wolves from promising areas of the search space. The experimental results on unconstrained problems of IEEE CEC 2014 and their analysis through various metrics such as statistical analysis, convergence analysis demonstrate the superior search ability of the RW-GWO algorithm in terms of exploring as well as exploiting the search space as compared to the classical GWO.

Table 2.1. Error values in objective function obtained by classical GWO and RW-GWO for 10-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal functions	**F1**	**GWO**	1.59E+05	1.42E+06	1.81E+06	9.37E+06	1.80E+06
		RW-GWO	**9.98E+02**	**1.72E+05**	**1.52E+05**	**2.71E+05**	**6.45E+04**
	F2	**GWO**	1.91E+02	**1.13E+03**	1.03E+07	2.65E+08	5.17E+07
		RW-GWO	**7.83E+01**	1.24E+03	**2.19E+03**	**9.28E+03**	**2.36E+03**
	F3	**GWO**	9.91E+01	4.41E+03	4.55E+03	1.30E+04	3.50E+03
		RW-GWO	**1.60E-01**	**8.17E+00**	**2.64E+01**	**4.35E+02**	**6.46E+01**
Multimodal functions	**F4**	**GWO**	6.05E+00	3.52E+01	3.27E+01	3.80E+01	8.88E+00
		RW-GWO	**2.55E+00**	**6.77E+00**	**6.55E+00**	**8.89E+00**	**1.21E+00**
	F5	**GWO**	1.87E+01	2.04E+01	2.03E+01	2.05E+01	**2.41E-01**
		RW-GWO	**5.00E-02**	**2.00E+01**	**1.96E+01**	**2.00E+01**	2.80E+00
	F6	**GWO**	2.14E-01	1.79E+00	1.92E+00	5.54E+00	**1.10E+00**
		RW-GWO	**9.34E-02**	**1.71E+00**	**1.49E+00**	**3.99E+00**	1.11E+00
	F7	**GWO**	**6.78E-02**	8.41E-01	1.11E+00	3.66E+00	8.29E-01
		RW-GWO	8.06E-02	**1.61E-01**	**1.61E-01**	**2.97E-01**	**5.25E-02**
	F8	**GWO**	3.98E+00	8.96E+00	1.05E+01	2.59E+01	4.69E+00
		RW-GWO	**1.99E+00**	**3.98E+00**	**4.51E+00**	**8.96E+00**	**1.47E+00**
	F9	**GWO**	6.16E+00	1.32E+01	1.47E+01	3.40E+01	7.13E+00
		RW-GWO	**3.00E+00**	**9.95E+00**	**1.07E+01**	**2.49E+01**	**4.70E+00**
	F10	**GWO**	1.42E+02	3.78E+02	3.95E+02	1.00E+03	2.11E+02
		RW-GWO	**1.53E+01**	**1.49E+02**	**1.34E+02**	**2.69E+02**	**6.98E+01**
	F11	**GWO**	1.31E+02	**4.37E+02**	**4.70E+02**	1.16E+03	2.18E+02
		RW-GWO	**1.25E+02**	5.59E+02	5.55E+02	**1.10E+03**	**1.95E+02**
	F12	**GWO**	**1.28E-02**	4.58E-01	6.08E-01	1.58E+00	5.22E-01
		RW-GWO	2.35E-02	**7.64E-02**	**8.93E-02**	**1.84E-01**	**3.92E-02**
	F13	**GWO**	7.77E-02	1.71E-01	1.69E-01	3.16E-01	5.81E-02
		RW-GWO	**7.47E-02**	**1.24E-01**	**1.23E-01**	**1.82E-01**	**3.10E-02**
	F14	**GWO**	5.15E-02	2.18E-01	3.37E-01	7.09E-01	2.18E-01
		RW-GWO	**2.91E-02**	**1.26E-01**	**1.37E-01**	**5.79E-01**	**9.51E-02**
	F15	**GWO**	4.47E-01	1.54E+00	1.57E+00	3.90E+00	7.89E-01
		RW-GWO	**2.96E-01**	**6.98E-01**	**7.41E-01**	**1.24E+00**	**1.98E-01**
	F16	**GWO**	9.82E-01	2.33E+00	2.31E+00	**3.40E+00**	**5.51E-01**
		RW-GWO	**5.13E-01**	**2.13E+00**	**2.09E+00**	3.51E+00	5.67E-01
Hybrid functions	**F17**	**GWO**	8.23E+02	2.57E+03	3.85E+03	1.69E+04	3.20E+03
		RW-GWO	**1.19E+02**	**1.43E+03**	**1.97E+03**	**7.78E+03**	**1.86E+03**
	F18	**GWO**	1.01E+02	**1.31E+03**	**3.59E+03**	**1.52E+04**	**4.02E+03**
		RW-GWO	**1.39E+01**	6.67E+03	7.09E+03	2.24E+04	6.35E+03
	F19	**GWO**	1.34E+00	2.03E+00	2.37E+00	5.80E+00	9.09E-01
		RW-GWO	**5.55E-01**	**1.62E+00**	**1.75E+00**	**3.39E+00**	**6.15E-01**
	F20	**GWO**	3.83E+01	1.16E+02	1.45E+03	8.18E+03	2.38E+03
		RW-GWO	**2.95E+00**	**1.24E+01**	**1.47E+01**	**5.18E+01**	**9.47E+00**
	F21	**GWO**	1.13E+02	9.38E+02	1.79E+03	6.18E+03	1.69E+03
		RW-GWO	**3.34E+01**	**2.82E+02**	**4.45E+02**	**3.03E+03**	**5.43E+02**
	F22	**GWO**	2.66E+01	1.46E+02	1.16E+02	1.71E+02	**5.70E+01**
		RW-GWO	**1.37E+00**	**3.73E+01**	**6.59E+01**	**1.64E+02**	5.71E+01

Composite functions	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
	F23	**GWO**	**3.29E+02**	3.35E+02	3.35E+02	3.45E+02	4.54E+00
		RW-GWO	**3.29E+02**	**3.29E+02**	**3.29E+02**	**3.29E+02**	**8.39E-05**
	F24	**GWO**	1.11E+02	1.27E+02	1.33E+02	2.03E+02	2.34E+01
		RW-GWO	**1.07E+02**	**1.19E+02**	**1.19E+02**	**1.35E+02**	**5.97E+00**
	F25	**GWO**	1.37E+02	2.00E+02	1.92E+02	**2.02E+02**	**1.83E+01**
		RW-GWO	**1.32E+02**	**1.98E+02**	**1.85E+02**	2.03E+02	2.30E+01
	F26	**GWO**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	4.08E-02
		RW-GWO	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**2.96E-02**
	F27	**GWO**	4.42E+00	3.46E+02	3.35E+02	**4.08E+02**	**7.15E+01**
		RW-GWO	**1.21E+00**	**3.40E+02**	**3.25E+02**	4.23E+02	8.60E+01
	F28	**GWO**	**2.39E+02**	3.71E+02	4.03E+02	6.91E+02	7.01E+01
		RW-GWO	3.06E+02	**3.06E+02**	**3.06E+02**	**3.07E+02**	**9.33E-02**
	F29	**GWO**	3.54E+02	6.36E+02	4.76E+04	2.39E+06	3.35E+05
		RW-GWO	**2.02E+02**	**2.05E+02**	**2.05E+02**	**2.11E+02**	**1.64E+00**
	F30	**GWO**	5.97E+02	8.99E+02	1.01E+03	2.08E+03	3.40E+02
		RW-GWO	**2.24E+02**	**2.82E+02**	**3.15E+02**	**5.90E+02**	**9.30E+01**

Table 2.2. Error values in objective function obtained by classical GWO and RW-GWO for 30-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal functions	**F1**	**GWO**	8.18E+06	3.04E+07	3.32E+07	9.79E+07	2.02E+07
		RW-GWO	**2.22E+06**	**7.66E+06**	**8.02E+06**	**1.94E+07**	**3.31E+06**
	F2	**GWO**	1.42E+06	6.10E+08	1.01E+09	5.72E+09	1.15E+09
		RW-GWO	**2.83E+04**	**9.28E+04**	**2.23E+05**	**3.35E+06**	**5.51E+05**
	F3	**GWO**	1.58E+04	2.81E+04	2.85E+04	4.58E+04	6.80E+03
		RW-GWO	**1.23E+01**	**5.57E+01**	**3.16E+02**	**1.34E+03**	**4.34E+02**
Multimodal functions	**F4**	**GWO**	1.00E+02	1.77E+02	1.94E+02	3.83E+02	5.82E+01
		RW-GWO	**1.87E+01**	**2.81E+01**	**3.41E+01**	**8.29E+01**	**1.80E+01**
	F5	**GWO**	2.08E+01	2.10E+01	2.10E+01	2.10E+01	**4.83E-02**
		RW-GWO	**2.03E+01**	**2.05E+01**	**2.05E+01**	**2.07E+01**	7.46E-02
	F6	**GWO**	6.24E+00	1.17E+01	1.16E+01	1.82E+01	**2.64E+00**
		RW-GWO	**3.21E+00**	**1.03E+01**	**9.84E+00**	**1.80E+01**	3.49E+00
	F7	**GWO**	2.27E+00	5.31E+00	7.67E+00	1.86E+01	4.64E+00
		RW-GWO	**8.68E-02**	**2.21E-01**	**2.53E-01**	**8.85E-01**	**1.43E-01**
	F8	**GWO**	3.42E+01	6.25E+01	6.50E+01	1.22E+02	1.45E+01
		RW-GWO	**2.49E+01**	**4.34E+01**	**4.38E+01**	**6.64E+01**	**8.48E+00**
	F9	**GWO**	3.88E+01	8.05E+01	8.54E+01	2.42E+02	3.30E+01
		RW-GWO	**3.41E+01**	**6.37E+01**	**6.33E+01**	**9.42E+01**	**1.30E+01**
	F10	**GWO**	6.99E+02	1.74E+03	1.80E+03	3.08E+03	4.93E+02
		RW-GWO	**5.23E+02**	**9.47E+02**	**9.61E+02**	**1.60E+03**	**2.72E+02**
	F11	**GWO**	**1.47E+03**	2.81E+03	2.90E+03	6.45E+03	7.24E+02
		RW-GWO	1.79E+03	**2.62E+03**	**2.68E+03**	**3.49E+03**	**3.68E+02**
	F12	**GWO**	8.20E-02	2.44E+00	2.12E+00	3.13E+00	9.58E-01
		RW-GWO	**2.57E-01**	**5.17E-01**	**5.45E-01**	**1.12E+00**	**1.66E-01**
	F13	**GWO**	2.19E-01	3.77E-01	3.74E-01	6.92E-01	8.88E-02
		RW-GWO	**1.85E-01**	**2.66E-01**	**2.80E-01**	**4.60E-01**	**6.30E-02**
	F14	**GWO**	**1.24E-01**	7.08E-01	7.49E-01	1.04E+01	1.40E+00
		RW-GWO	1.85E-01	**3.01E-01**	**4.23E-01**	**7.72E-01**	**2.15E-01**
	F15	**GWO**	**3.96E+00**	1.46E+01	2.06E+01	1.39E+02	2.20E+01
		RW-GWO	5.08E+00	**8.79E+00**	**8.81E+00**	**1.26E+01**	**1.51E+00**
	F16	**GWO**	9.45E+00	1.10E+01	1.09E+01	1.20E+01	**5.80E-01**
		RW-GWO	**8.98E+00**	**1.02E+01**	**1.03E+01**	**1.15E+01**	6.11E-01
Hybrid functions	**F17**	**GWO**	**4.61E+04**	**4.46E+05**	6.28E+05	3.59E+06	6.11E+05
		RW-GWO	5.68E+04	4.52E+05	**5.71E+05**	**2.06E+06**	**4.10E+05**
	F18	**GWO**	2.12E+03	2.11E+04	5.27E+06	6.41E+07	1.34E+07
		RW-GWO	**4.89E+02**	**6.23E+03**	**6.52E+03**	**1.83E+04**	**4.62E+03**
	F19	**GWO**	7.50E+00	2.07E+01	2.56E+01	8.35E+01	1.77E+01
		RW-GWO	**7.40E+00**	**1.11E+01**	**1.14E+01**	**1.61E+01**	**2.03E+00**
	F20	**GWO**	4.00E+03	1.19E+04	1.31E+04	2.90E+04	5.26E+03
		RW-GWO	**1.02E+02**	**2.66E+02**	**6.27E+02**	**6.00E+03**	**1.12E+03**
	F21	**GWO**	6.12E+04	**1.60E+05**	4.97E+05	4.74E+06	1.05E+06
		RW-GWO	**2.60E+04**	2.42E+05	**2.58E+05**	**6.22E+05**	**1.76E+05**
	F22	**GWO**	**5.13E+01**	1.90E+02	2.50E+02	6.32E+02	**1.16E+02**
		RW-GWO	3.32E+01	**1.62E+02**	**2.08E+02**	**5.43E+02**	1.29E+02

Composite functions	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
	F23	**GWO**	3.17E+02	3.27E+02	3.28E+02	3.38E+02	4.16E+00
		RW-GWO	**3.14E+02**	**3.15E+02**	**3.15E+02**	**3.15E+02**	**2.77E-01**
	F24	**GWO**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**7.27E-04**
		RW-GWO	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	3.04E-03
	F25	**GWO**	2.07E+02	2.11E+02	2.11E+02	2.15E+02	2.04E+00
		RW-GWO	**2.02E+02**	**2.05E+02**	**2.04E+02**	**2.07E+02**	**1.18E+00**
	F26	**GWO**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.01E+02**	9.62E-02
		RW-GWO	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**7.36E-02**
	F27	**GWO**	**4.03E+02**	4.30E+02	4.33E+02	4.86E+02	1.82E+01
		RW-GWO	**4.03E+02**	**4.08E+02**	**4.09E+02**	**4.40E+02**	**6.09E+00**
	F28	**GWO**	7.93E+02	9.07E+02	9.14E+02	1.12E+03	6.63E+01
		RW-GWO	**4.16E+02**	**4.35E+02**	**4.34E+02**	**4.53E+02**	**8.45E+00**
	F29	**GWO**	4.98E+03	3.28E+04	2.90E+05	1.12E+07	1.57E+06
		RW-GWO	**2.08E+02**	**2.14E+02**	**2.14E+02**	**2.19E+02**	**2.37E+00**
	F30	**GWO**	8.09E+03	2.71E+04	2.98E+04	6.80E+04	1.57E+04
		RW-GWO	**2.76E+02**	**6.62E+02**	**6.69E+02**	**1.13E+03**	**2.14E+02**

Table 2.3. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 10-dimentional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	1.40E-09	+	**F16**	6.62E-02	=
F2	4.20E-01	=	**F17**	2.47E-04	+
F3	5.15E-10	+	**F18**	4.80E-03	-
F4	7.35E-10	+	**F19**	3.82E-04	+
F5	9.14E-09	+	**F20**	5.15E-10	+
F6	6.76E-02	=	**F21**	8.69E-08	+
F7	1.49E-09	+	**F22**	1.69E-05	+
F8	3.14E-09	+	**F23**	5.15E-10	+
F9	8.90E-03	+	**F24**	3.57E-05	+
F10	4.94E-09	+	**F25**	2.09E-01	=
F11	5.96E-02	=	**F26**	1.20E-01	=
F12	4.81E-07	+	**F27**	2.90E-01	=
F13	1.36E-04	+	**F28**	2.50E-09	+
F14	1.75E-06	+	**F29**	5.15E-10	+
F15	7.74E-09	+	**F30**	5.15E-10	+

Table 2.4. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 30-dimentional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	6.15E-10	+	**F16**	9.24E-05	+
F2	5.15E-10	+	**F17**	7.57E-01	=
F3	5.15E-10	+	**F18**	6.03E-08	+
F4	5.15E-10	+	**F19**	3.73E-09	+
F5	3.60E-10	+	**F20**	5.15E-10	+
F6	1.28E-02	+	**F21**	2.09E-01	=
F7	5.15E-10	+	**F22**	4.05E-02	+
F8	2.36E-09	+	**F23**	4.89E-10	+
F9	2.22E-05	+	**F24**	2.27E-11	-
F10	1.87E-09	+	**F25**	4.76E-10	+
F11	5.46E-02	=	**F26**	5.82E-06	+
F12	6.92E-09	+	**F27**	7.13E-09	+
F13	3.24E-07	+	**F28**	5.14E-10	+
F14	3.70E-03	+	**F29**	5.15E-10	+
F15	8.49E-06	+	**F30**	5.15E-10	+

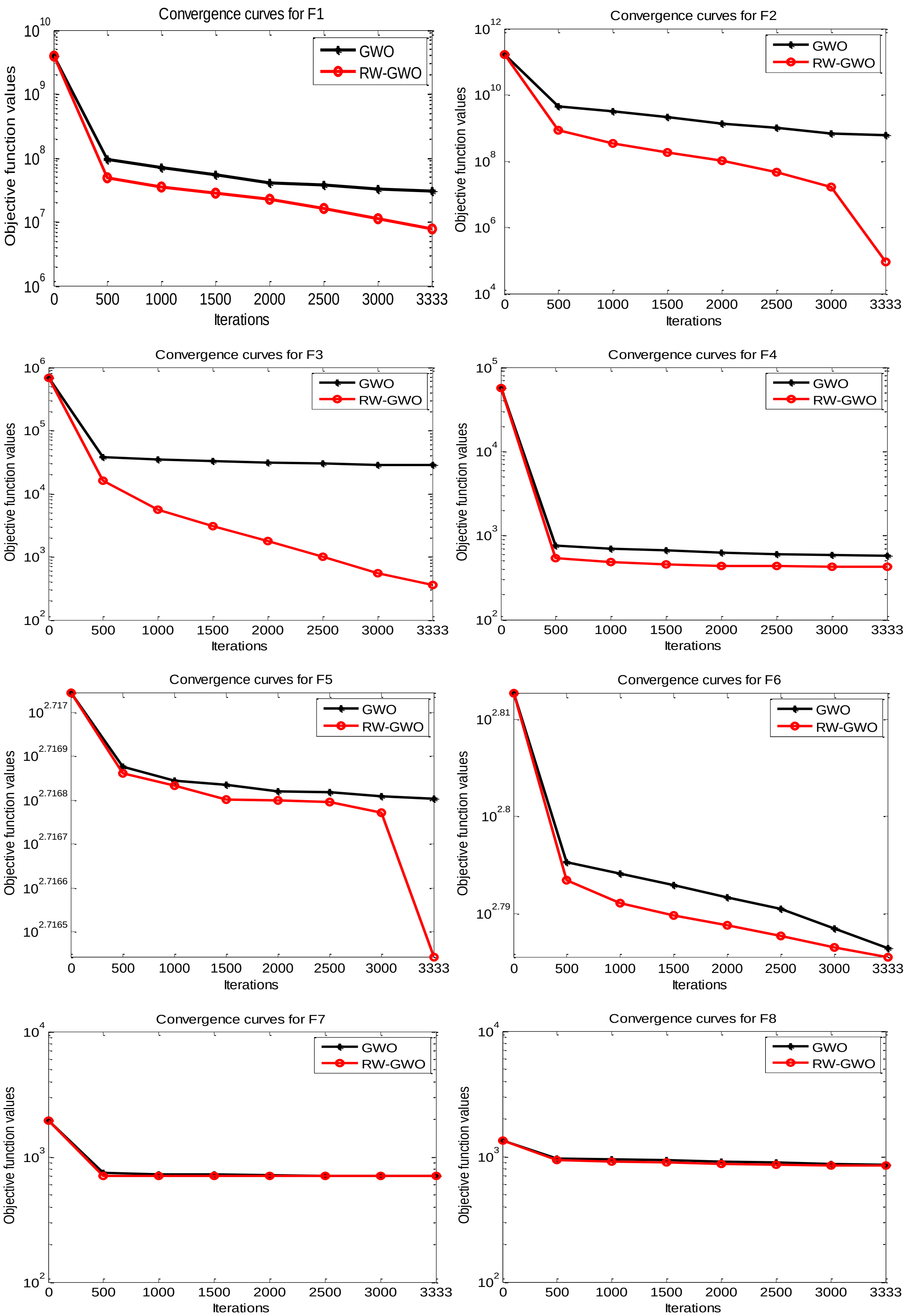


Fig 2.2. Convergence curves for 30-dimensional problems from F1 to F8 corresponding to alpha solution of each iteration

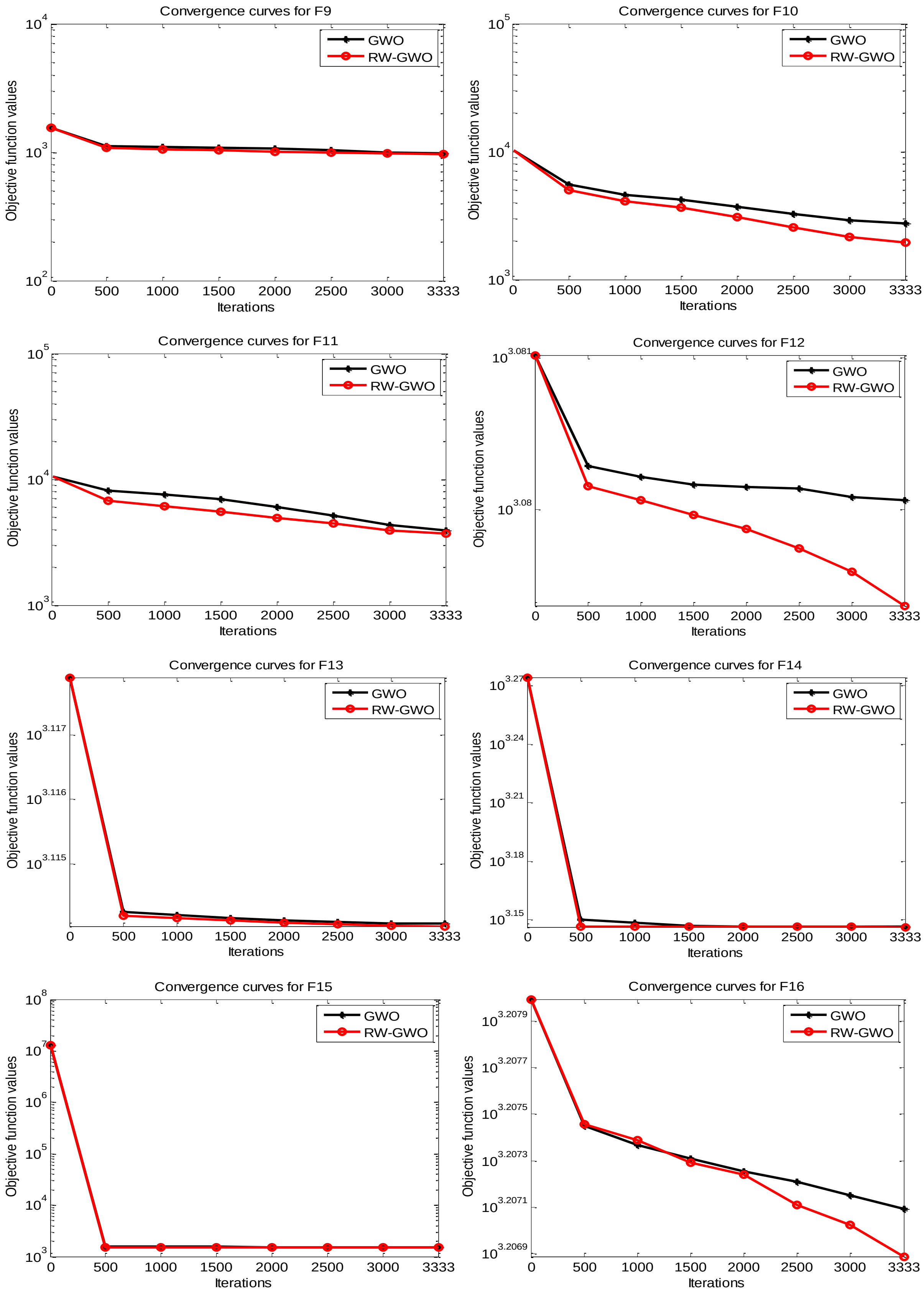


Fig 2.3. Convergence curves for 30-dimensional problems from F9 to F16 corresponding to alpha solution of each iteration

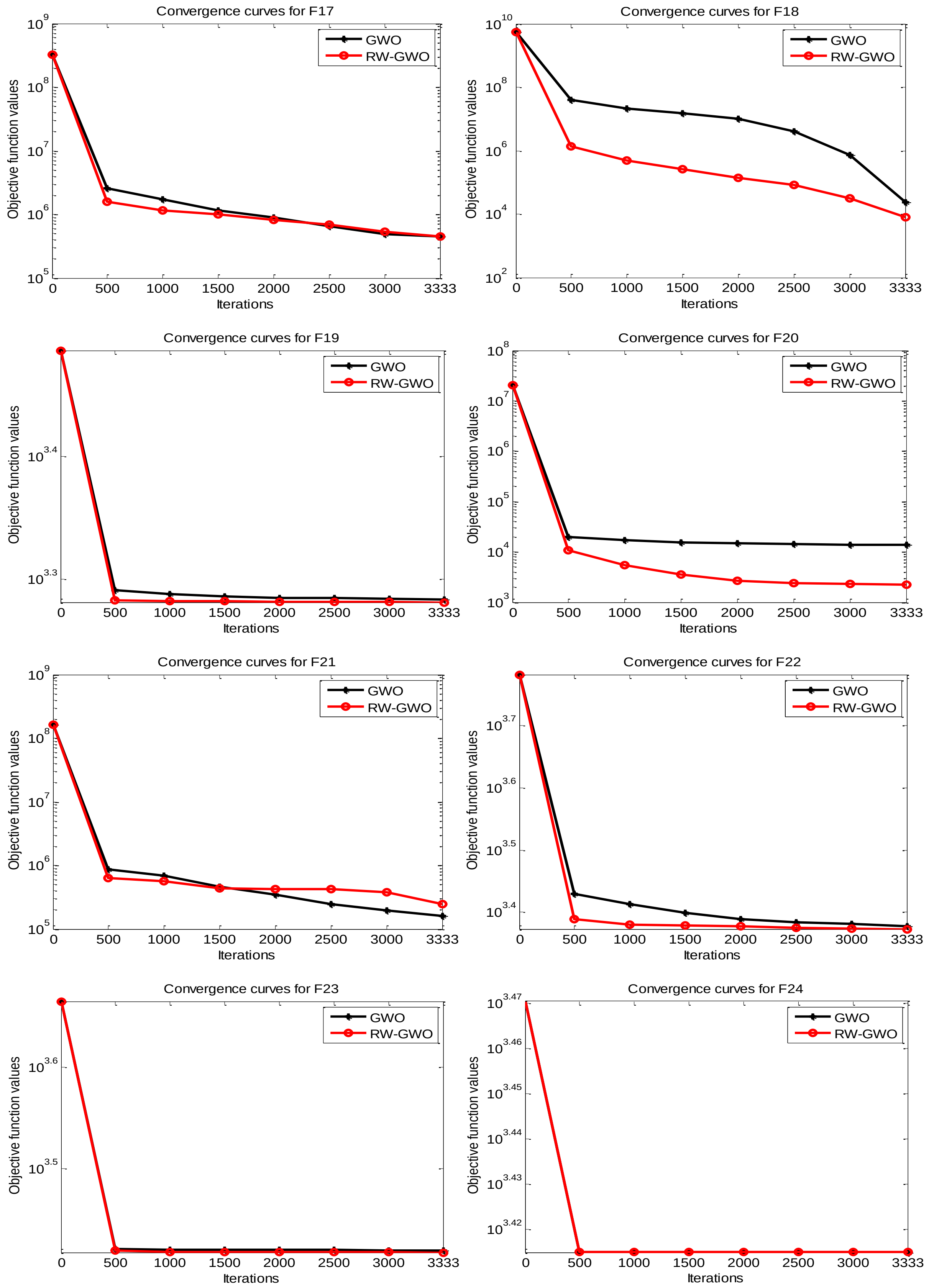


Fig 2.4. Convergence curves for 30-dimensional problems from F17 to F24 corresponding to alpha solution of each iteration

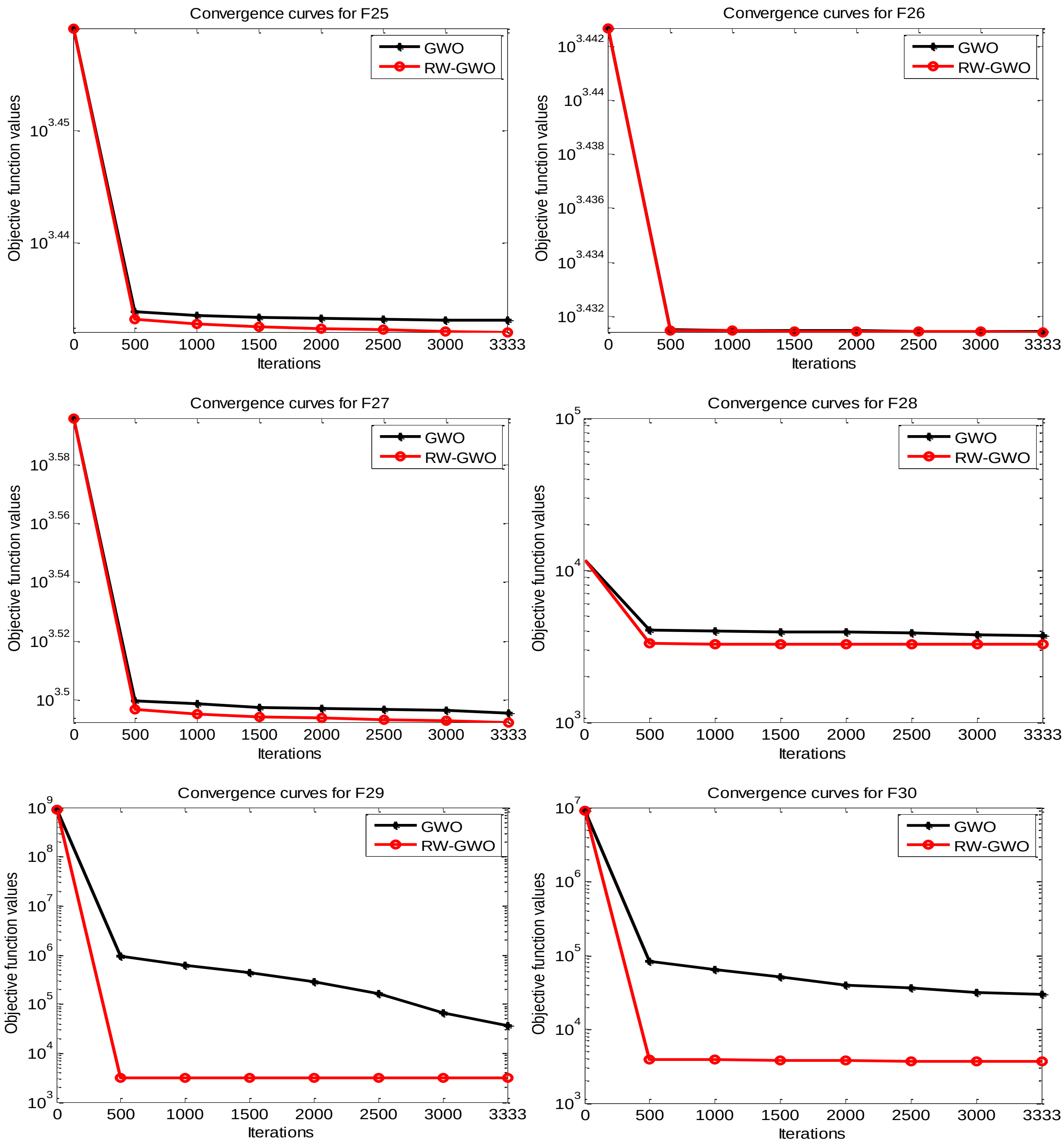


Fig 2.5. Convergence curves for 30-dimensional problems from F25 to F30 corresponding to alpha solution of each iteration

Chapter 3

A Memory-based Grey Wolf Optimizer for Unconstrained Optimization Problems

In this chapter another attempt is made to improve the performance of classical GWO.

3.1. Introduction

Although the classical Grey Wolf Optimizer is capable enough in exploring the promising regions of the search space but in some cases, the algorithm suffers from the issue of getting trapped at local optima and premature convergence. In order to overcome these issues, a novel variant of GWO called Memory-based Grey Wolf Optimizer (mGWO) is proposed in the present chapter. The mGWO utilizes the personal best states of individual wolves which is saved in the memory of wolves to update their positions. The performance of the proposed mGWO algorithm is tested on an unconstrained benchmark problem set given in IEEE CEC 2014 and the results are analyzed and compared with classical GWO. In the chapter, the performance of the mGWO is also evaluated with respect to the RW-GWO which was presented in the previous chapter.

The organization of the chapter is as follows: Section 3.2 provides a motivation behind proposing a new variant of classical GWO and detailed description of the proposed mGWO. Section 3.3 provides numerical experimentation, analysis and comparison of the mGWO with classical GWO. In Section 3.4, a comparison is shown between the mGWO and RW-GWO (introduced in Chapter 2). Finally, the chapter is closed with concluding remarks in Section 3.5.

3.2. Proposed Memory-based Grey Wolf Optimizer (mGWO)

3.2.1. Motivation

The search equation of classical GWO confirms the dependency of the search directions towards the leading hunters (alpha, beta and delta) of the wolf pack. The leading hunters sometimes get stuck in a local optima and fail to provide a promising direction of search for omega wolves.

The content of this chapter is communicated in:

Gupta, S., & Deep, K. (2019). A Memory-based Grey Wolf Optimizer for global optimization and image segmentation. Expert Systems with Applications, Elsevier (Revision submitted).

This situation is usually faced in multimodal functions where large number of local optima are present. In classical GWO, when the leaders get trapped in these local optima then it is difficult to pull out the wolf pack from these optima as the pack is highly dependent on leading hunters. The stagnation at local optima is the cause of premature convergence which is harmful in obtaining the true solution (optima) of the problem. The personal best state of individual wolf which is saved in the memory of wolf can help in such situations and can explore the more promising areas of search space. The individual wolves can share their best knowledge to the leading hunters for better leading search directions. By motivating these collaborative mechanism and information-exchange characteristics, the present chapter utilizes the personal best history of wolves along with the leading guidance to enhance the search efficiency of wolf pack.

3.2.2. Framework of Memory-based Grey Wolf Optimizer

The proposed algorithm called Memory-based Grey Wolf Optimizer (mGWO) integrates the personal best knowledge of the individual wolves in GWO during the search to strengthen the collaboration among the wolves. The strategies introduced in mGWO can be pointed out as follows:

1. The direction of search is now decided from the personal best history towards the leading hunters so that the most promising regions of the search space can be explored.
2. The personal best guidance is added in the search mechanism with the help of new search equation based on random directions.
3. The crossover is performed between the positions obtained through leading guidance and personal best guidance.
4. The greedy selection mechanism is applied to maintain the balance between the exploration and exploitation and to avoid the wolves to diverge from promising regions during the search.

The above strategies are integrated into the classical search mechanism of GWO to enhance the exploration of search space and to alleviate the issues of stagnation at local optima and premature convergence.

The newly proposed mathematical model of encircling mechanism based on the personal best history of wolves is as follows:

$$X_{t+1} = X_p - A_t \times |C_t \times X_p - X_{pbest}| \tag{3.1}$$

where X_{pbest} is the personal best state saved in the memory of wolf X upto iteration t. The other symbols are same as defined in classical GWO. After the encircling mechanism, the hunting is performed based on the leading guidance provided by alpha, beta and delta wolves. The hunting mechanism now utilizes the personal best knowledge of each wolf to approximate the prey location. In the proposed modified hunting mechanism, it has been assumed that the individual wolves may have the information of prey. The modified hunting mechanism based on the personal best state of each wolf can be defined as follows:

$$Z_{i,t+1} = \frac{Y_1 + Y_2 + Y_3}{3} \tag{3.2}$$

where,

$$Y_1 = X_\alpha - A_{\alpha,t} \times |C_{\alpha,t} \times X_\alpha - X_{i_{pbest}}| \tag{3.3}$$

$$Y_2 = X_\beta - A_{\beta,t} \times |C_{\beta,t} \times X_\beta - X_{i_{pbest}}| \tag{3.4}$$

$$Y_3 = X_\delta - A_{\delta,t} \times |C_{\delta,t} \times X_\delta - X_{i_{pbest}}| \tag{3.5}$$

where $Z_{i,t+1}$ is the updated position of the wolf X_i through hunting mechanism, $X_{i_{pbest}}$ is the personal best state saved in the memory of wolf X_i upto iteration t. Rest of the symbols are same as defined in the search equation of classical GWO. In order to explore and retrace the neighbourhood areas around the personal best state of wolves and to mimic the concept that the individual wolf may have the information about the prey, a new search equation is proposed given by

$$\hat{X}_{i,t+1} = X_{i_{pbest}} + k \times (X_{r_1} - X_{r_2}) \tag{3.6}$$

where X_{r_1} and X_{r_2} are the wolves which are randomly selected from the pack. The parameter k is scaling factor which controls the effect of difference vector. The higher value of k leads to the high exploration and the small values favors the exploitation. Therefore, in the present chapter, the value of the parameter k is selected as a variable which decreases linearly from 1 to 0 over the course of iterations. This value of parameter k helps in exploiting the neighbourhood areas around the personal best states of wolves. In order to combine the information about the prey obtained from leading hunters and individual wolves and to obtain a new state of wolf, the crossover is performed between the positions obtained from leading and personal best guidance, which can be defined as follows:

$$X_{i,t+1}^j = \begin{cases} Z_{i,t+1}^j & if\ rand < CR \\ \hat{X}_{i,t+1}^j & otherwise \end{cases} \tag{3.7}$$

where, CR is the crossover probability which is fixed as 0.5 in our study. $Z_{i,t+1}$ and $\hat{X}_{i,t+1}$ are the positions obtained from equations (3.2) and (3.6) respectively. When each wolf of the pack is updated using equation (3.7), a greedy selection is applied corresponding to each wolf between its current and previous states. This greedy selection mechanism avoids the divergence of wolves from discovered promising areas of the search space. The steps of the proposed mGWO are provided in Algorithm 3.1.

Algorithm 3.1: pseudo code of Memory-based Grey Wolf Optimizer (mGWO)

---

1. $For \quad Min\ F(X) \quad s.t. \quad X_{\min} \leq X \leq X_{\max},\ X = (x_1, x_2, \ldots, x_D) \in R^D$

2. **Initialize** the grey wolf population X_i $(i = 1,2,\ldots,N)$

3. **Evaluate** the fitness of each grey wolf

4. **Initialize** the parameters CR and maximum number of iterations T

5. **Initialize** the memory matrix for grey wolf pack as $\left[X_{i_{pbest}}\right]_{i=1}^{N} = [X_i]_{i=1}^{N}$

6. $\boldsymbol{Select}$ $X_\alpha = fittest\ wolf\ of\ the\ pack$

$X_\beta = second\ best\ wolf$

$X_\delta = third\ best\ wolf$

7. Initialize the iteration count $t = 0$

8. $\boldsymbol{while}$ $t < T$

9. $\boldsymbol{for}$ each leader wolf

10. update the position of each wolf using proposed search equation (3.7)

11. $\boldsymbol{end\ for}$

12. **Evaluate** the fitness of each grey wolf

13. $\boldsymbol{for}$ $i = 1,2,\ldots,N$

14. $\boldsymbol{if}$ $F(X_{i,t+1}) \geq F(X_{i,t})$

15. $X_{i,t+1} = X_{i,t}$

16. $\boldsymbol{end\ if}$

17. $\boldsymbol{end\ for}$

18. **Update** the leading wolves X_α, X_β and X_δ

19. $t = t + 1$

20. **end while**

21. $return\ the\ alpha\ wolf$

3.2.3. Computational Complexity

The time complexities of the proposed mGWO is discussed as follows:

1. The mGWO initializes the wolf pack in $O(N \times D)$ time, where N the size of pack and D represent the dimension of the problem.
2. Selection of leading hunters requires $O(N)$ time.
3. Position update in the mGWO requires the $O(N \times D)$ time.
4. The greedy selection requires an additional $O(N)$ time in the proposed mGWO.
5. Fitness evaluation of updated wolves requires $O(N)$ time.

In summary, the total computational time for the proposed mGWO algorithms is equal to $O(N \times D \times T)$ for maximum number of iterations T and this is same as for classical GWO.

3.3. Numerical Experiments and Analysis of Results

3.3.1. Benchmark Functions and Parameter Setting

In the chapter the proposed mGWO is tested on the same set of unconstrained benchmark problems which are given in IEEE CEC 2014 [185] and used in Chapter 2. In our experimentation, the dimension of the problems are fixed as 10 and 30. The population size for the classical GWO and mGWO is fixed as $3 \times D$, and the termination criteria is taken as the maximum number of function evaluations. The termination criteria is selected as per the guidelines of IEEE CEC 2014 which is $10^4 \times D$, where D represents the dimension of the problem.

3.3.2. Analysis of the Results

In this section, the numerical results obtained by implementing classical GWO and mGWO on IEEE CEC 2014 benchmark problems are provided. The results are presented in the form of absolute error in objective function values and the better results are highlighted in bold face. For a feasible solution X and optima X^* to the problem F, the absolute error is calculated by $|F(X) - F(X^*)|$. The experiments are performed on 10 and 30-dimensional problems and various criteria, such as minimum, median, mean, maximum, standard deviation (STD) of the absolute errors in objective function values of test problems are presented. The performance of the mGWO on different categories of benchmarks corresponding to 10 and 30-dimensonal problems is analyzed as follows:

The results for 10-dimensional problems

The results for 10-dimensional problems are given in Table 3.1. On observing the results from the table, it is found that the mGWO performs better as compared to the classical GWO.

In 10-dimensional unimodal problems F1 and F3, the mGWO provides better results in all the criteria as compared to classical GWO. In problem F2, the mGWO provides better mean, maximum and standard deviation value of errors as compared to classical GWO. In terms of minimum and median error, the classical GWO provides better results as compared to the mGWO.

In 10-dimensional multimodal problems F6-F11, F14 and F15, the mGWO provides better results in all the criteria as compared to classical GWO. In problems F4, F5 and F16, the mGWO provides better results in all criteria except standard deviation as compared to classical GWO. In F12, classical GWO is better than mGWO in terms of minimum error while in other criteria mGWO provides better results as compared to classical GWO. In F13, the mGWO is better than classical GWO in all the criteria except maximum error.

In all the 10-dimensional hybrid problems (F17-F22), the mGWO provides better results in terms of all the criteria as compared to classical GWO.

In 10 dimensional composite problems (F23-F25 and F27-F30), the mGWO performs better than classical GWO in all the criteria except for standard deviation in the problems F25, F27 and F28. In F26, both the algorithm are same in terms of all the criteria except for standard deviation which is better in mGWO than the classical GWO.

The results for 30 dimension

The results for 30-dimension problems are given in Table 3.2. On observing the results from the table, it is found that the mGWO performs better for 30-dimsnional problems also as compared to the classical GWO.

In all the 30-dimensional unimodal problems from F1 and F3, the mGWO provides better results in all the criteria as compared to classical GWO.

In 30-dimensional multimodal problems F4, F6-F11, F13 and F15, the mGWO provides better results in all the criteria as compared to classical GWO. In problems F5 and F16, the mGWO provides better results in all criteria except standard deviation as compared to classical GWO. In problem F12 and F14, classical GWO is better than mGWO in terms of minimum error while in other criteria mGWO provides better results as compared to classical GWO.

In all the 30-dimensional hybrid problems (F17-F22), the mGWO provides better results in terms of all the criteria as compared to classical GWO except for minimum error in F17.

In 30 dimensional composite problems (F23-F30), the mGWO performs better than classical GWO in all the criteria except for standard deviation in the problems F24.

Hence, an overall analysis of the proposed mGWO on different category of benchmarks demonstrate the better exploration and exploitation ability of the mGWO as compared to the classical GWO. The experimental results also demonstrate that the proposed strategies in the mGWO establishes a more appropriate balance between exploration and exploitation as compared to the classical GWO.

3.3.3. Statistical Analysis

In order to make concrete conclusions about the significance of differences in the performance of the proposed mGWO and classical GWO, a non-parametric Wilcoxon test [186] is applied. The test has been conducted at 5% level of significance. The statistical results are presented in Tables 3.3 and 3.4 corresponding to 10 and 30-dimensional IEEE CEC 2014 problems. In these tables, $'+/=/-'$ sign are used to indicate that the mGWO is significantly better, equal or worse than the classical GWO. From the tables, it can be observed that out of 30 problems the mGWO is better than classical GWO in 28 problems corresponding to 10-dimensional benchmarks. Similarly for 30-dimensional problems, the mGWO is better than classical GWO in 29 problems. Overall, from the statistical results, it can be concluded that the proposed mGWO algorithm has significantly improved the search efficiency and accuracy of wolves in obtaining the solution.

3.3.4. Convergence Behavior

Although the alpha, beta and delta wolves are considered as the leading wolves for the pack and are responsible for guiding the search, but the elite solution of the pack is alpha. Therefore, the improvement in the solution for any problem can be analyzed through the solution alpha. Hence, the history of the elite solution is plotted in Figs 3.1 to 3.4 for the classical GWO and mGWO in terms of convergence curves corresponding to the 30-dimensional benchmark problems. The growth of iterations is shown on horizontal axis, and the vertical axis represents the objective function values. By inspecting the convergence history of alpha solution in the mGWO, it is empirical to conclude that the proposed mGWO algorithm is successful to provide a significant improvement in the search efficiency of classical GWO in terms of convergence rate.

3.4. Comparison Between RW-GWO and mGWO Algorithms

Although, both the variants RW-GWO (proposed in the Chapter 2) and the mGWO (proposed in the current chapter) provide better results as compared to the classical GWO, but the comparison between these two variants is required on different categories of benchmark problems to elect the best performer. Therefore, in this section, a comparison is performed between the algorithms RW-GWO and mGWO. For the sake of comparison, the results of the RW-GWO are reproduced and presented in Tables 3.1 and 3.2 for 10 and 30-dimensional problems respectively.

The comparison for 10 dimension

The results for 10-dimension problems are given in Table 3.1. The description of the results on different category of benchmarks is as follows:

In 10-dimensional unimodal problem F1, the RW-GWO provides better minimum, maximum and standard deviation value of error while in terms of mean and median value of error, mGWO is better than others. In the problems F2 and F3, the RW-GWO is better than classical GWO and mGWO in terms of all the criteria except for median value in F2. The median value of error is better in classical GWO than others for F2.

In 10-dimensional multimodal problems, F6, F8-F11, mGWO provides better results in all criteria as compared to the classical GWO and RW-GWO. In F4, the mGWO is better than other in terms of minimum and median of error values while in terms of other criteria, RW-GWO provides better results than others. In F5, in terms of minimum and mean value of error, mGWO, in terms of median and maximum value of error, RW-GWO, and in terms of standard deviation value classical GWO is better than other comparative algorithms. In F7, the RW-GWO is better than classical GWO and mGWO for all the criteria except minimum error, for minimum error mGWO is better than other algorithms. In F12, the mGWO is better in terms of median and mean error, in terms of minimum error classical GWO is better and in terms of maximum and standard deviation of error, RW-GWO is better than its comparative algorithms. In F13, minimum error is better in mGWO, median error is better in mGWO as well as RW-GWO, and mean, maximum and standard deviation of errors are better in RW-GWO. In F14, mGWO provides better results in all criteria than others except minimum error. In terms of minimum error, RW-GWO is better than others for F14. In F15, except for the minimum error, RW-GWO is better than others and for minimum error mGWO is better than other algorithms. In F16, mGWO is better in all the criteria except standard deviation and for standard deviation, RW-GWO is better than other algorithms.

In 10-dimensional hybrid problems F17, F19 and F22, the mGWO provides better results in all criteria as compared to the classical GWO and RW-GWO. In F18 and F21, in terms of minimum error, RW-GWO is better than other while in other criteria, mGWO provides better results as compared to classical GWO and RW-GWO. In F20, RW-GWO provides better results than classical GWO and mGWO in terms of all the criteria.

In 10-dimensional composite problems F23, F26, F29 and F30, the RW-GWO is provides either better results or same results as compared to the classical GWO and mGWO. In F24, the mGWO provides better results in all the criteria as compared to the others. In F25, the mGWO is better than others in terms of minimum, mean and maximum error while the RW-GWO is better in terms of median and standard deviation than other algorithms. In F27, the mGWO is better than others in terms of mean, median and maximum error while in terms of minimum and standard deviation, RW-GWO provides better results as compared to the others. In F28, the RW-GWO provides better than other algorithms in all the criteria except minimum error. The minimum error in F28 is better in mGWO than other algorithms.

The comparison for 30 dimension

The results for 30-dimension problems are given in Table 3.2. The description of the results on different category of 30-dimensional benchmarks is as follows:

In 30-dimensional unimodal problem F1, the mGWO provides better results as compared to the classical GWO and RW-GWO in all the criteria except for standard deviation. For standard deviation, RW-GWO is better than the other algorithms. In problems F2 and F3, the RW-GWO provides better results in all the criteria as compared to the classical GWO and mGWO.

In 30-dimensional multimodal problems, F4, F5 and F7, RW-GWO provides better results in all criteria as compared to the classical GWO and mGWO. In F6, F8-F10, mGWO provides better results in all criteria as compared to the classical GWO and RW-GWO. In F11, the standard deviation value of errors is better in RW-GWO while other criteria are better in mGWO. In F12, the minimum and median value of errors is better in mGWO while in other criteria RW-GWO provides better results than others. In F13, minimum, median, and mean values of errors are better in RW-GWO while maximum and standard deviation are better in mGWO as compared to other comparative algorithms. In F14, the classical GWO provides better minimum error while in terms of other criteria, mGWO is better than others. In F15, the mGWO is better than others in providing the minimum, median and mean value of errors while the other criteria are better in RW-GWO. In F16, minimum, median and mean values of errors are better in mGWO, maximum value of errors is better in RW-GWO and standard deviation value is better in classical GWO than other comparative algorithms.

In 30-dimensional hybrid problems F17- F19, F21 and F22, the mGWO provides better results in all criteria as compared to the classical GWO and RW-GWO. In F20, the RW-GWO provides better minimum and median value of errors as compared to classical GWO and mGWO while in remaining criteria, mGWO is better than others.

In 30-dimensional composite problems F23, the minimum and maximum errors are better in RW-GWO, median and mean errors are same in RW-GWO and mGWO but better than classical GWO. The standard deviation is better in mGWO than other algorithms for F23. In F24, all the criteria are same in all the algorithms except for standard deviation. The standard deviation for F24 is better in classical GWO than others. In F25, the minimum, mean and maximum error value of errors are better in the RW-GWO and median value is same for RW-GWO and mGWO but better than classical GWO while the standard deviation is better in mGWO than other algorithms. In F26, classical GWO, RW-GWO and mGWO provides same results in terms of minimum, median, mean and maximum value of errors. The standard deviation values for F26 is better in mGWO as compared to the other algorithms. In F27, mGWO is better than other algorithms for all the criteria. In F28-F30, the RW-GWO provides better results in all criteria as compared to the classical GWO and mGWO.

Although, the numerical results demonstrate the differences in providing the results, but in order to analyze the best performer corresponding to each category of benchmarks and to make concrete conclusions about the significance of differences in the performance of algorithms, statistical comparison between RW-GWO and mGWO is performed through Wilcoxon signed rank test. The comparison is performed using same parameter setting as used in previous chapter. The statistical results are listed in Table 3.5 and the best performer is listed in the same table. The convergence behavior of classical GWO, RW-GWO and mGWO is also compared in the Figs 3.1 to 3.4. In the problems F2-F5, F12, F22, F25, F28-F30, the convergence rate is better in RW-GWO as compared to classical GWO and mGWO. In problems F1, F6, F10, F17 and F21 the convergence is better in RW-GWO at some fixed initial iteration and after that the convergence in mGWO is better as compared to other algorithms. Overall, from all the comparison analysis the following remarks can be made:

1. In terms of worst time complexity calculated through big $-O$ notation, all the algorithms classical GWO, RW-GWO and mGWO are same.

2. On unimodal problems, the RW-GWO performs better as compared to the classical GWO and mGWO.
3. On multimodal and hybrid problems, the mGWO algorithm is more successful as compared to the classical GWO and RW-GWO.
4. On the set of composite problems, the RW-GWO algorithm is more successful as compared to the classical GWO and mGWO.
5. The convergence rate is better in RW-GWO as compared to classical GWO and mGWO.

3.5. Concluding Remarks

In the chapter, a new variant of classical GWO called Memory-based GWO (mGWO) is proposed which enhances the collaborative strength of the wolf pack by utilizing the personal best history of wolves. The incorporation of personal best state of wolves in the search equation helps to explore the neighbourhood regions of available promising areas in the search space. The applied greedy selection mechanism between two consecutive iterations of algorithm maintain the balance between exploration and exploitation and avoids the divergence of wolves from promising areas of the search space. The experimental results, statistical and convergence analysis on IEEE CEC 2014 benchmarks demonstrate the superior search ability of the mGWO as compared to the classical GWO.

Moreover, the performance comparison between the variants RW-GWO (proposed in Chapter 2) and mGWO (proposed in the current chapter) concludes that the RW-GWO outperforms mGWO in unimodal and composite problems while in the multimodal and hybrid problems, the mGWO outperforms RW-GWO. Thus, the exploration strength is better in the mGWO as compared to RW-GWO and the exploitation strength of RW-GWO is better than mGWO. In maintaining the balance between exploration and exploitation, both the algorithms are very competitive. The convergence behaviour of algorithms demonstrate the better convergence rate in RW-GWO as compared to classical GWO and mGWO.

Table 3.1. Error values in objective function obtained by classical GWO, RW-GWO and mGWO for 10-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	minimum	median	mean	maximum	STD
Unimodal problems	**F1**	**GWO**	1.59E+05	1.42E+06	1.81E+06	9.37E+06	1.80E+06
		RW-GWO	**9.98E+02**	1.72E+05	1.52E+05	**2.71E+05**	**6.45E+04**
		mGWO	2.80E+03	**3.00E+04**	**6.37E+04**	5.39E+05	9.68E+04
	F2	**GWO**	1.91E+02	**1.13E+03**	1.03E+07	2.65E+08	5.17E+07
		RW-GWO	**7.83E+01**	1.24E+03	**2.19E+03**	**9.28E+03**	**2.36E+03**
		mGWO	2.31E+02	1.38E+03	2.58E+03	1.03E+04	2.77E+03
	F3	**GWO**	9.91E+01	4.41E+03	4.55E+03	1.30E+04	3.50E+03
		RW-GWO	**1.60E-01**	**8.17E+00**	**2.64E+01**	**4.35E+02**	**6.46E+01**
		mGWO	3.81E+00	3.52E+01	6.08E+01	4.39E+02	8.42E+01
Multimodal problems	**F4**	**GWO**	6.05E+00	3.52E+01	3.27E+01	3.80E+01	8.88E+00
		RW-GWO	2.55E+00	6.77E+00	**6.55E+00**	**8.89E+00**	**1.21E+00**
		mGWO	**1.13E-01**	**4.53E+00**	1.66E+01	3.54E+01	1.63E+01
	F5	**GWO**	1.87E+01	2.04E+01	2.03E+01	2.05E+01	**2.41E-01**
		RW-GWO	5.00E-02	**2.00E+01**	1.96E+01	**2.00E+01**	2.80E+00
		mGWO	**4.05E-02**	2.02E+01	**1.74E+01**	2.04E+01	6.98E+00
	F6	**GWO**	2.14E-01	1.79E+00	1.92E+00	5.54E+00	1.10E+00
		RW-GWO	9.34E-02	1.71E+00	1.49E+00	3.99E+00	1.11E+00
		mGWO	**6.96E-02**	**1.53E-01**	**1.82E-01**	**1.07E+00**	**1.43E-01**
	F7	**GWO**	6.78E-02	8.41E-01	1.11E+00	3.66E+00	8.29E-01
		RW-GWO	8.06E-02	**1.61E-01**	**1.61E-01**	**2.97E-01**	**5.25E-02**
		mGWO	**6.63E-02**	1.88E-01	2.05E-01	4.52E-01	8.28E-02
	F8	**GWO**	3.98E+00	8.96E+00	1.05E+01	2.59E+01	4.69E+00
		RW-GWO	1.99E+00	3.98E+00	4.51E+00	8.96E+00	1.47E+00
		mGWO	**4.70E-04**	**1.99E+00**	**2.01E+00**	**4.98E+00**	**1.13E+00**
	F9	**GWO**	6.16E+00	1.32E+01	1.47E+01	3.40E+01	7.13E+00
		RW-GWO	3.00E+00	9.95E+00	1.07E+01	2.49E+01	4.70E+00
		mGWO	**3.24E-04**	**3.98E+00**	**4.58E+00**	**1.09E+01**	**2.32E+00**
	F10	**GWO**	1.42E+02	3.78E+02	3.95E+02	1.00E+03	2.11E+02
		RW-GWO	1.53E+01	1.49E+02	1.34E+02	2.69E+02	6.98E+01
		mGWO	**3.37E-01**	**6.97E+00**	**9.97E+00**	**1.25E+02**	**1.73E+01**
	F11	**GWO**	1.31E+02	4.37E+02	4.70E+02	1.16E+03	2.18E+02
		RW-GWO	1.25E+02	5.59E+02	5.55E+02	1.10E+03	1.95E+02
		mGWO	**3.48E-01**	**1.22E+02**	**1.00E+02**	**3.88E+02**	**9.14E+01**
	F12	**GWO**	**1.28E-02**	4.58E-01	6.08E-01	1.58E+00	5.22E-01
		RW-GWO	2.35E-02	7.64E-02	8.93E-02	**1.84E-01**	**3.92E-02**
		mGWO	2.06E-02	**6.72E-02**	**8.46E-02**	2.35E-01	5.78E-02
	F13	**GWO**	7.77E-02	1.71E-01	1.69E-01	3.16E-01	5.81E-02
		RW-GWO	7.47E-02	**1.24E-01**	**1.23E-01**	**1.82E-01**	**3.10E-02**
		mGWO	**7.43E-02**	**1.24E-01**	1.32E-01	3.35E-01	4.10E-02
	F14	**GWO**	5.15E-02	2.18E-01	3.37E-01	7.09E-01	2.18E-01
		RW-GWO	**2.91E-02**	1.26E-01	1.37E-01	5.79E-01	9.51E-02
		mGWO	4.04E-02	**1.01E-01**	**1.03E-01**	**1.94E-01**	**3.57E-02**
	F15	**GWO**	4.47E-01	1.54E+00	1.57E+00	3.90E+00	7.89E-01
		RW-GWO	2.96E-01	**6.98E-01**	**7.41E-01**	**1.24E+00**	**1.98E-01**
		mGWO	**2.65E-01**	8.26E-01	8.19E-01	1.37E+00	2.57E-01

	Function	Algorithm	minimum	median	mean	maximum	STD
	F16	**GWO**	9.82E-01	2.33E+00	2.31E+00	3.40E+00	5.51E-01
		RW-GWO	5.13E-01	2.13E+00	2.09E+00	3.51E+00	**5.67E-01**
		mGWO	**2.14E-01**	**1.39E+00**	**1.36E+00**	**2.58E+00**	6.11E-01
Hybrid problems	**F17**	**GWO**	8.23E+02	2.57E+03	3.85E+03	1.69E+04	3.20E+03
		RW-GWO	1.19E+02	1.43E+03	1.97E+03	7.78E+03	1.86E+03
		mGWO	**1.01E+02**	**1.02E+03**	**1.22E+03**	**4.30E+03**	**9.03E+02**
	F18	**GWO**	1.01E+02	1.31E+03	3.59E+03	1.52E+04	4.02E+03
		RW-GWO	**1.39E+01**	6.67E+03	7.09E+03	2.24E+04	6.35E+03
		mGWO	7.86E+01	**3.82E+02**	**1.22E+03**	**7.93E+03**	**1.89E+03**
	F19	**GWO**	1.34E+00	2.03E+00	2.37E+00	5.80E+00	9.09E-01
		RW-GWO	5.55E-01	1.62E+00	1.75E+00	3.39E+00	6.15E-01
		mGWO	**9.25E-02**	**1.26E+00**	**1.17E+00**	**1.62E+00**	**4.38E-01**
	F20	**GWO**	3.83E+01	1.16E+02	1.45E+03	8.18E+03	2.38E+03
		RW-GWO	**2.95E+00**	**1.24E+01**	**1.47E+01**	**5.18E+01**	**9.47E+00**
		mGWO	1.97E+01	5.42E+01	5.31E+01	9.63E+01	2.17E+01
	F21	**GWO**	1.13E+02	9.38E+02	1.79E+03	6.18E+03	1.69E+03
		RW-GWO	**3.34E+01**	2.82E+02	4.45E+02	3.03E+03	5.43E+02
		mGWO	3.36E+01	**1.13E+02**	**1.55E+02**	**4.59E+02**	**1.08E+02**
	F22	**GWO**	2.66E+01	1.46E+02	1.16E+02	1.71E+02	5.70E+01
		RW-GWO	1.37E+00	3.73E+01	6.59E+01	1.64E+02	5.71E+01
		mGWO	**3.29E-01**	**2.08E+01**	**1.82E+01**	**1.20E+02**	**1.77E+01**
Composite problems	**F23**	**GWO**	3.29E+02	3.35E+02	3.35E+02	3.45E+02	4.54E+00
		RW-GWO	**3.29E+02**	**3.29E+02**	**3.29E+02**	**3.29E+02**	**8.39E-05**
		mGWO	**3.29E+02**	**3.29E+02**	**3.29E+02**	**3.29E+02**	1.61E-03
	F24	**GWO**	1.11E+02	1.27E+02	1.33E+02	2.03E+02	2.34E+01
		RW-GWO	1.07E+02	1.19E+02	1.19E+02	1.35E+02	5.97E+00
		mGWO	**1.00E+02**	**1.10E+02**	**1.10E+02**	**1.19E+02**	**3.96E+00**
	F25	**GWO**	1.37E+02	2.00E+02	1.92E+02	2.02E+02	1.83E+01
		RW-GWO	1.32E+02	**1.98E+02**	1.85E+02	2.03E+02	**2.30E+01**
		mGWO	**1.00E+02**	1.99E+02	**1.76E+02**	**2.02E+02**	3.63E+01
	F26	**GWO**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	4.08E-02
		RW-GWO	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**2.96E-02**
		mGWO	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	3.28E-02
	F27	**GWO**	4.42E+00	3.46E+02	3.35E+02	4.08E+02	7.15E+01
		RW-GWO	**1.21E+00**	3.40E+02	3.25E+02	4.23E+02	**8.60E+01**
		mGWO	1.41E+00	**3.05E+02**	**2.17E+02**	**4.00E+02**	1.71E+02
	F28	**GWO**	2.39E+02	3.71E+02	4.03E+02	6.91E+02	7.01E+01
		RW-GWO	3.06E+02	**3.06E+02**	**3.06E+02**	**3.07E+02**	**9.33E-02**
		mGWO	**1.02E+02**	3.57E+02	3.52E+02	5.00E+02	1.01E+02
	F29	**GWO**	3.54E+02	6.36E+02	4.76E+04	2.39E+06	3.35E+05
		RW-GWO	**2.02E+02**	**2.05E+02**	**2.05E+02**	**2.11E+02**	**1.64E+00**
		mGWO	2.60E+02	4.56E+02	4.90E+02	8.77E+02	1.42E+02
	F30	**GWO**	5.97E+02	8.99E+02	1.01E+03	2.08E+03	3.40E+02
		RW-GWO	**2.24E+02**	**2.82E+02**	**3.15E+02**	**5.90E+02**	**9.30E+01**
		mGWO	5.07E+02	6.81E+02	7.59E+02	1.29E+03	1.90E+02

Table 3.2. Error values in objective function obtained by classical GWO, RW-GWO and mGWO for 30-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal problems	F1	GWO	8.18E+06	3.04E+07	3.32E+07	9.79E+07	2.02E+07
		RW-GWO	2.22E+06	7.66E+06	8.02E+06	1.94E+07	**3.31E+06**
		mGWO	**1.97E+06**	**5.62E+06**	**6.57E+06**	**1.81E+07**	3.82E+06
	F2	GWO	1.42E+06	6.10E+08	1.01E+09	5.72E+09	1.15E+09
		RW-GWO	**2.83E+04**	**9.28E+04**	**2.23E+05**	**3.35E+06**	**5.51E+05**
		mGWO	7.33E+04	6.72E+06	1.88E+07	1.01E+08	2.62E+07
	F3	GWO	1.58E+04	2.81E+04	2.85E+04	4.58E+04	6.80E+03
		RW-GWO	**1.23E+01**	**5.57E+01**	**3.16E+02**	**1.34E+03**	**4.34E+02**
		mGWO	1.80E+02	6.36E+02	7.17E+02	2.97E+03	4.69E+02
Multimodal problems	F4	GWO	1.00E+02	1.77E+02	1.94E+02	3.83E+02	5.82E+01
		RW-GWO	**1.87E+01**	**2.81E+01**	**3.41E+01**	**8.29E+01**	**1.80E+01**
		mGWO	6.95E+01	1.16E+02	1.15E+02	1.55E+02	2.74E+01
	F5	GWO	2.08E+01	2.10E+01	2.10E+01	2.10E+01	4.83E-02
		RW-GWO	**2.03E+01**	**2.05E+01**	**2.05E+01**	**2.07E+01**	**7.46E-02**
		mGWO	2.06E+01	2.09E+01	2.09E+01	2.10E+01	7.99E-02
	F6	GWO	6.24E+00	1.17E+01	1.16E+01	1.82E+01	2.64E+00
		RW-GWO	3.21E+00	1.03E+01	9.84E+00	1.80E+01	3.49E+00
		mGWO	**1.20E+00**	**2.99E+00**	**3.17E+00**	**8.77E+00**	**1.46E+00**
	F7	GWO	2.27E+00	5.31E+00	7.67E+00	1.86E+01	4.64E+00
		RW-GWO	**8.68E-02**	**2.21E-01**	**2.53E-01**	**8.85E-01**	**1.43E-01**
		mGWO	3.63E-01	1.09E+00	1.06E+00	1.55E+00	2.49E-01
	F8	GWO	3.42E+01	6.25E+01	6.50E+01	1.22E+02	1.45E+01
		RW-GWO	2.49E+01	4.34E+01	4.38E+01	6.64E+01	8.48E+00
		mGWO	**1.58E+01**	**2.79E+01**	**2.75E+01**	**4.30E+01**	**5.72E+00**
	F9	GWO	3.88E+01	8.05E+01	8.54E+01	2.42E+02	3.30E+01
		RW-GWO	3.41E+01	6.37E+01	6.33E+01	9.42E+01	1.30E+01
		mGWO	**2.33E+01**	**4.32E+01**	**4.35E+01**	**7.69E+01**	**1.11E+01**
	F10	GWO	6.99E+02	1.74E+03	1.80E+03	3.08E+03	4.93E+02
		RW-GWO	5.23E+02	9.47E+02	9.61E+02	1.60E+03	2.72E+02
		mGWO	**2.37E+01**	**4.16E+02**	**4.06E+02**	**8.79E+02**	**2.22E+02**
	F11	GWO	1.47E+03	2.81E+03	2.90E+03	6.45E+03	7.24E+02
		RW-GWO	1.79E+03	2.62E+03	2.68E+03	3.49E+03	**3.68E+02**
		mGWO	**9.28E+02**	**1.94E+03**	**1.96E+03**	**3.21E+03**	5.07E+02
	F12	GWO	8.20E-02	2.44E+00	2.12E+00	3.13E+00	9.58E-01
		RW-GWO	2.57E-01	5.17E-01	**5.45E-01**	**1.12E+00**	**1.66E-01**
		mGWO	**1.19E-01**	**4.81E-01**	6.22E-01	2.36E+00	4.57E-01
	F13	GWO	2.19E-01	3.77E-01	3.74E-01	6.92E-01	8.88E-02
		RW-GWO	**1.85E-01**	**2.66E-01**	**2.80E-01**	4.60E-01	6.30E-02
		mGWO	2.02E-01	3.11E-01	3.12E-01	**4.44E-01**	**4.62E-02**
	F14	GWO	**1.24E-01**	7.08E-01	7.49E-01	1.04E+01	1.40E+00
		RW-GWO	1.85E-01	3.01E-01	4.23E-01	7.72E-01	2.15E-01
		mGWO	1.37E-01	**2.22E-01**	**2.21E-01**	**2.95E-01**	**4.29E-02**
	F15	GWO	3.96E+00	1.46E+01	2.06E+01	1.39E+02	2.20E+01
		RW-GWO	5.08E+00	8.79E+00	8.81E+00	**1.26E+01**	**1.51E+00**
		mGWO	**3.30E+00**	**6.23E+00**	**6.48E+00**	1.31E+01	1.89E+00

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
	F16	**GWO**	9.45E+00	1.10E+01	1.09E+01	1.20E+01	**5.80E-01**
		RW-GWO	8.98E+00	1.02E+01	1.03E+01	**1.15E+01**	6.11E-01
		mGWO	**8.19E+00**	**1.01E+01**	**1.01E+01**	1.17E+01	7.77E-01
Hybrid problems	**F17**	**GWO**	**4.61E+04**	4.46E+05	6.28E+05	3.59E+06	6.11E+05
		RW-GWO	5.68E+04	4.52E+05	5.71E+05	2.06E+06	4.10E+05
		mGWO	5.27E+04	**1.95E+05**	**2.20E+05**	**5.91E+05**	**1.36E+05**
	F18	**GWO**	2.12E+03	2.11E+04	5.27E+06	6.41E+07	1.34E+07
		RW-GWO	4.89E+02	6.23E+03	6.52E+03	1.83E+04	4.62E+03
		mGWO	**4.03E+02**	**9.82E+02**	**1.68E+03**	**1.00E+04**	**1.78E+03**
	F19	**GWO**	7.50E+00	2.07E+01	2.56E+01	8.35E+01	1.77E+01
		RW-GWO	7.40E+00	1.11E+01	1.14E+01	1.61E+01	2.03E+00
		mGWO	**4.95E+00**	**7.17E+00**	**7.07E+00**	**1.05E+01**	**1.20E+00**
	F20	**GWO**	4.00E+03	1.19E+04	1.31E+04	2.90E+04	5.26E+03
		RW-GWO	**1.02E+02**	**2.66E+02**	6.27E+02	6.00E+03	1.12E+03
		mGWO	1.56E+02	2.95E+02	**3.30E+02**	**8.12E+02**	**1.18E+02**
	F21	**GWO**	6.12E+04	1.60E+05	4.97E+05	4.74E+06	1.05E+06
		RW-GWO	2.60E+04	2.42E+05	2.58E+05	6.22E+05	1.76E+05
		mGWO	**1.36E+04**	**7.24E+04**	**8.68E+04**	**2.60E+05**	**5.95E+04**
	F22	**GWO**	5.13E+01	1.90E+02	2.50E+02	6.32E+02	1.16E+02
		RW-GWO	3.32E+01	1.62E+02	2.08E+02	5.43E+02	1.29E+02
		mGWO	**2.86E+01**	**1.60E+02**	**1.62E+02**	**3.06E+02**	**6.69E+01**
Composite problems	**F23**	**GWO**	3.17E+02	3.27E+02	3.28E+02	3.38E+02	4.16E+00
		RW-GWO	**3.14E+02**	**3.15E+02**	**3.15E+02**	**3.15E+02**	2.77E-01
		mGWO	3.15E+02	**3.15E+02**	**3.15E+02**	3.16E+02	**1.87E-01**
	F24	**GWO**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**7.27E-04**
		RW-GWO	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	3.04E-03
		mGWO	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	1.44E-02
	F25	**GWO**	2.07E+02	2.11E+02	2.11E+02	2.15E+02	2.04E+00
		RW-GWO	**2.02E+02**	**2.05E+02**	**2.04E+02**	**2.07E+02**	1.18E+00
		mGWO	2.04E+02	**2.05E+02**	2.05E+02	2.08E+02	**9.30E-01**
	F26	**GWO**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.01E+02**	9.62E-02
		RW-GWO	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	7.36E-02
		mGWO	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**6.59E-02**
	F27	**GWO**	4.03E+02	4.30E+02	4.33E+02	4.86E+02	1.82E+01
		RW-GWO	4.03E+02	4.08E+02	4.09E+02	4.40E+02	6.09E+00
		mGWO	**4.02E+02**	**4.04E+02**	**4.05E+02**	**4.15E+02**	**2.42E+00**
	F28	**GWO**	7.93E+02	9.07E+02	9.14E+02	1.12E+03	6.63E+01
		RW-GWO	**4.16E+02**	**4.35E+02**	**4.34E+02**	**4.53E+02**	**8.45E+00**
		mGWO	6.62E+02	8.78E+02	8.78E+02	9.89E+02	5.37E+01
	F29	**GWO**	4.98E+03	3.28E+04	2.90E+05	1.12E+07	1.57E+06
		RW-GWO	**2.08E+02**	**2.14E+02**	**2.14E+02**	**2.19E+02**	**2.37E+00**
		mGWO	3.38E+03	1.06E+04	1.24E+04	5.64E+04	9.00E+03
	F30	**GWO**	8.09E+03	2.71E+04	2.98E+04	6.80E+04	1.57E+04
		RW-GWO	**2.76E+02**	**6.62E+02**	**6.69E+02**	**1.13E+03**	**2.14E+02**
		mGWO	2.83E+03	5.55E+03	5.84E+03	1.04E+04	1.75E+03

Table 3.3. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 10-dimensional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	5.15E-10	+	**F16**	2.19E-08	+
F2	0	=	**F17**	6.03E-08	+
F3	5.15E-10	+	**F18**	8.47E-04	+
F4	1.12E-07	+	**F19**	6.53E-10	+
F5	1.97E-07	+	**F20**	5.85E-09	+
F6	5.15E-10	+	**F21**	5.15E-10	+
F7	3.33E-09	+	**F22**	5.80E-10	+
F8	5.15E-10	+	**F23**	6.53E-10	+
F9	1.77E-09	+	**F24**	5.15E-10	+
F10	5.15E-10	+	**F25**	4.90E-02	+
F11	2.10E-09	+	**F26**	0	=
F12	1.07E-07	+	**F27**	4.24E-04	+
F13	5.24E-04	+	**F28**	1.52E-02	+
F14	1.32E-09	+	**F29**	8.20E-07	+
F15	3.77E-07	+	**F30**	1.25E-05	+

Table 3.4. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 30-dimensional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	6.15E-10	+	**F16**	9.93E-07	+
F2	5.46E-10	+	**F17**	3.09E-07	+
F3	5.15E-10	+	**F18**	1.87E-09	+
F4	3.52E-09	+	**F19**	5.15E-10	+
F5	1.53E-07	+	**F20**	5.15E-10	+
F6	5.46E-10	+	**F21**	1.20E-08	+
F7	5.15E-10	+	**F22**	1.37E-05	+
F8	5.15E-10	+	**F23**	5.15E-10	+
F9	1.05E-09	+	**F24**	5.15E-10	-
F10	5.15E-10	+	**F25**	5.15E-10	+
F11	1.14E-08	+	**F26**	4.78E-06	+
F12	1.27E-08	+	**F27**	6.15E-10	+
F13	1.64E-04	+	**F28**	8.92E-03	+
F14	8.25E-08	+	**F29**	3.56E-08	+
F15	5.85E-09	+	**F30**	5.15E-10	+

Table 3.5. Comparison between RW-GWO and mGWO algorithms

Function	Dimension = 10		Dimension = 30	
	p-value	winner	p-value	winner
F1	6.24E-06	**mGWO**	1.96E-02	**mGWO**
F2	5.67E-01	**same**	6.93E-10	**RW-GWO**
F3	2.76E-04	**RW-GWO**	2.86E-04	**RW-GWO**
F4	3.18E-02	**mGWO**	5.15E-10	**RW-GWO**
F5	1.74E-03	**RW-GWO**	5.15E-10	**RW-GWO**
F6	6.92E-09	**mGWO**	8.77E-10	**mGWO**
F7	1.74E-03	**RW-GWO**	6.15E-10	**RW-GWO**
F8	1.87E-09	**mGWO**	3.14E-09	**mGWO**
F9	8.65E-09	**mGWO**	8.69E-08	**mGWO**
F10	5.15E-10	**mGWO**	6.53E-10	**mGWO**
F11	5.15E-10	**mGWO**	4.17E-08	**mGWO**
F12	2.90E-01	**same**	8.22E-01	**same**
F13	2.81E-01	**same**	3.77E-03	**RW-GWO**
F14	2.70E-02	**mGWO**	1.97E-07	**mGWO**
F15	1.20E-01	**same**	1.61E-07	**mGWO**
F16	2.53E-06	**mGWO**	9.52E-02	**same**
F17	2.83E-02	**mGWO**	2.66E-07	**mGWO**
F18	3.33E-06	**mGWO**	1.69E-07	**mGWO**
F19	6.52E-06	**mGWO**	5.15E-10	**mGWO**
F20	8.77E-10	**RW-GWO**	2.65E-01	**same**
F21	1.59E-06	**mGWO**	5.31E-07	**mGWO**
F22	1.45E-06	**mGWO**	3.58E-02	**mGWO**
F23	2.83E-02	**same**	5.80E-10	**mGWO**
F24	2.50E-09	**mGWO**	5.15E-10	**RW-GWO**
F25	7.43E-01	**same**	9.27E-06	**RW-GWO**
F26	1.52E-01	**same**	2.57E-01	**same**
F27	1.04E-03	**mGWO**	2.53E-06	**mGWO**
F28	4.89E-04	**RW-GWO**	5.15E-10	**RW-GWO**
F29	5.15E-10	**RW-GWO**	5.15E-10	**RW-GWO**
F30	5.15E-10	**RW-GWO**	5.15E-10	**RW-GWO**

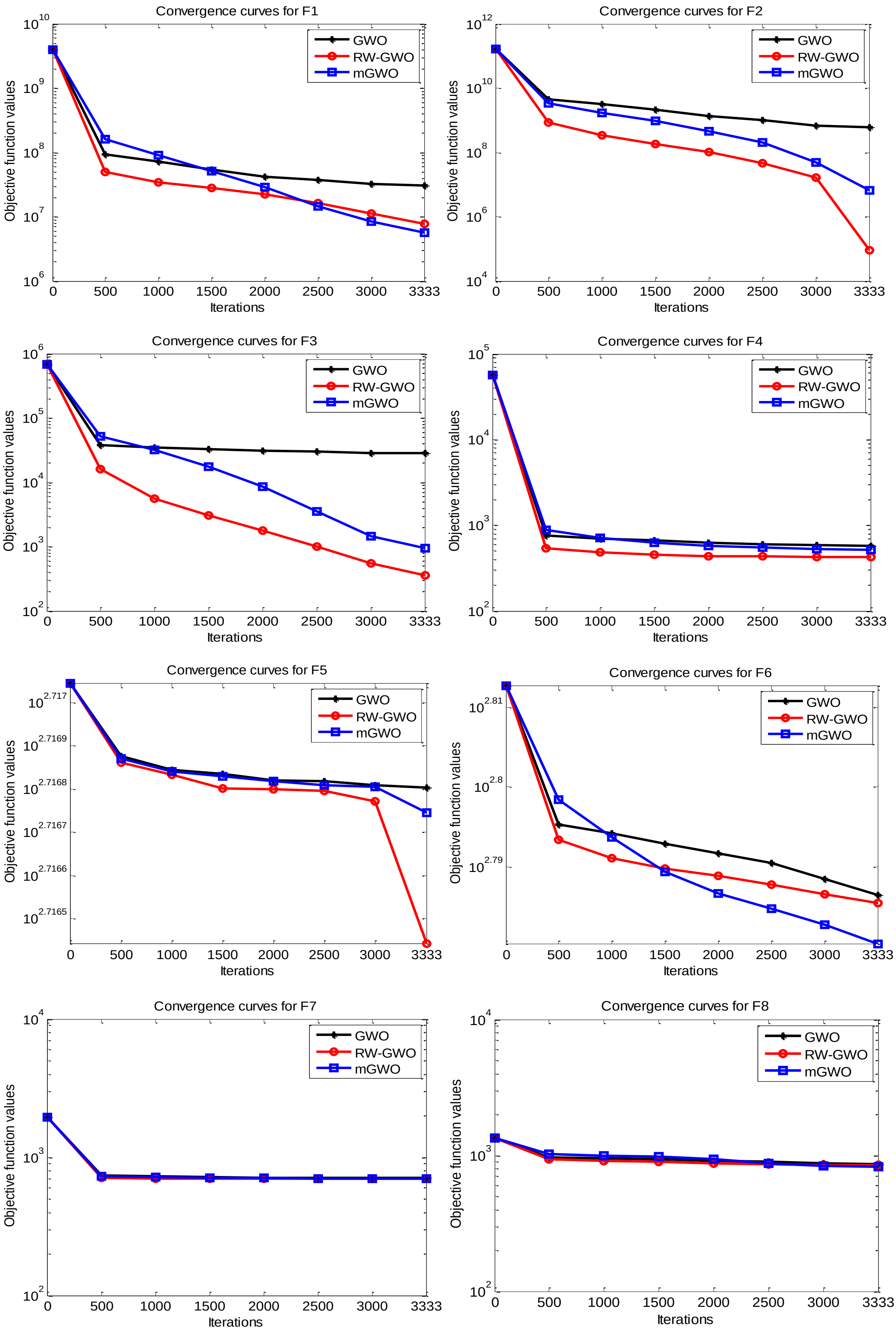


Fig 3.1. Convergence curves for 30-dimensional problems from F1 to F8 corresponding to alpha solution of each iteration

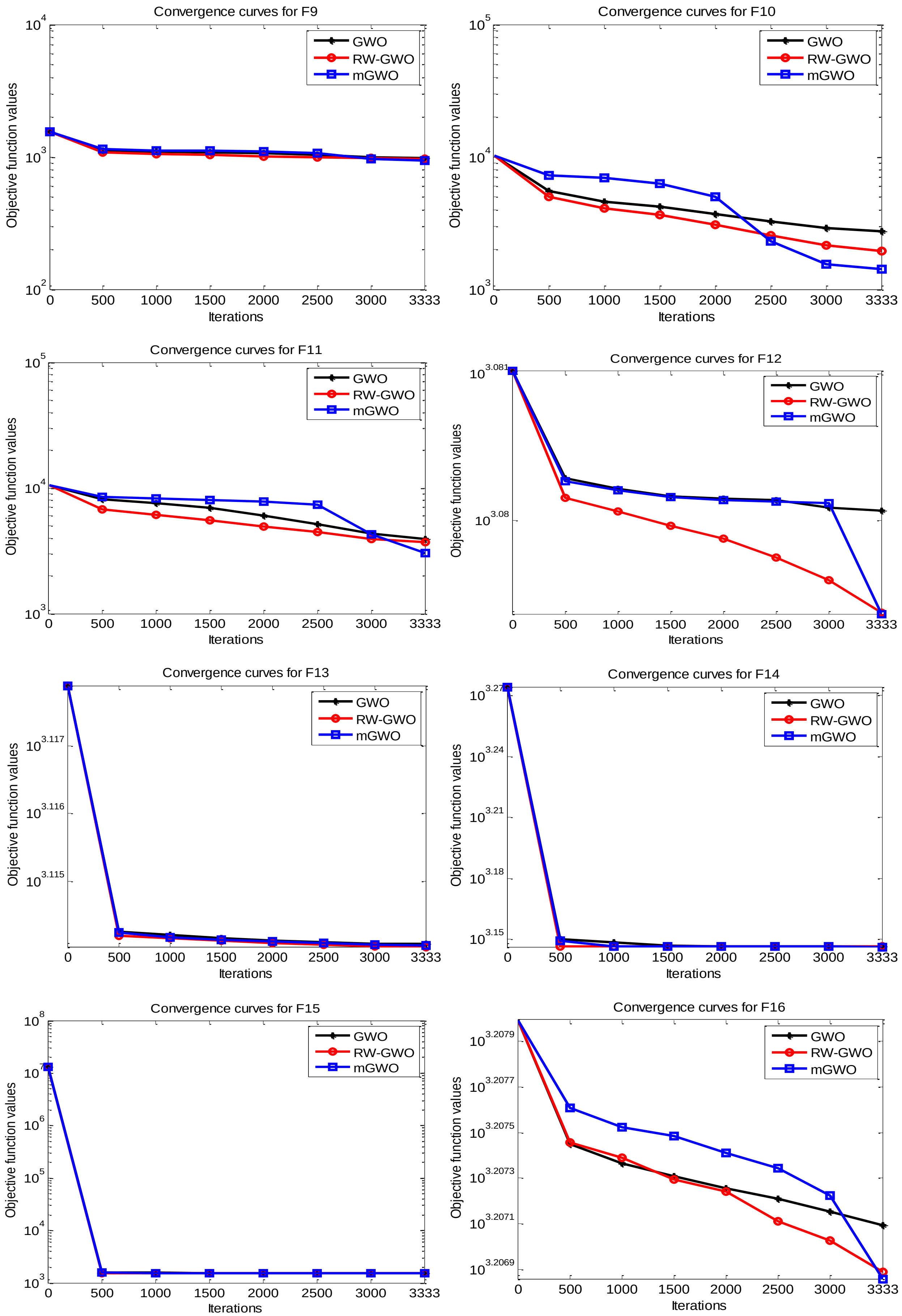


Fig 3.2. Convergence curves for 30-dimensional problems from F9 to F16 corresponding to alpha solution of each iteration

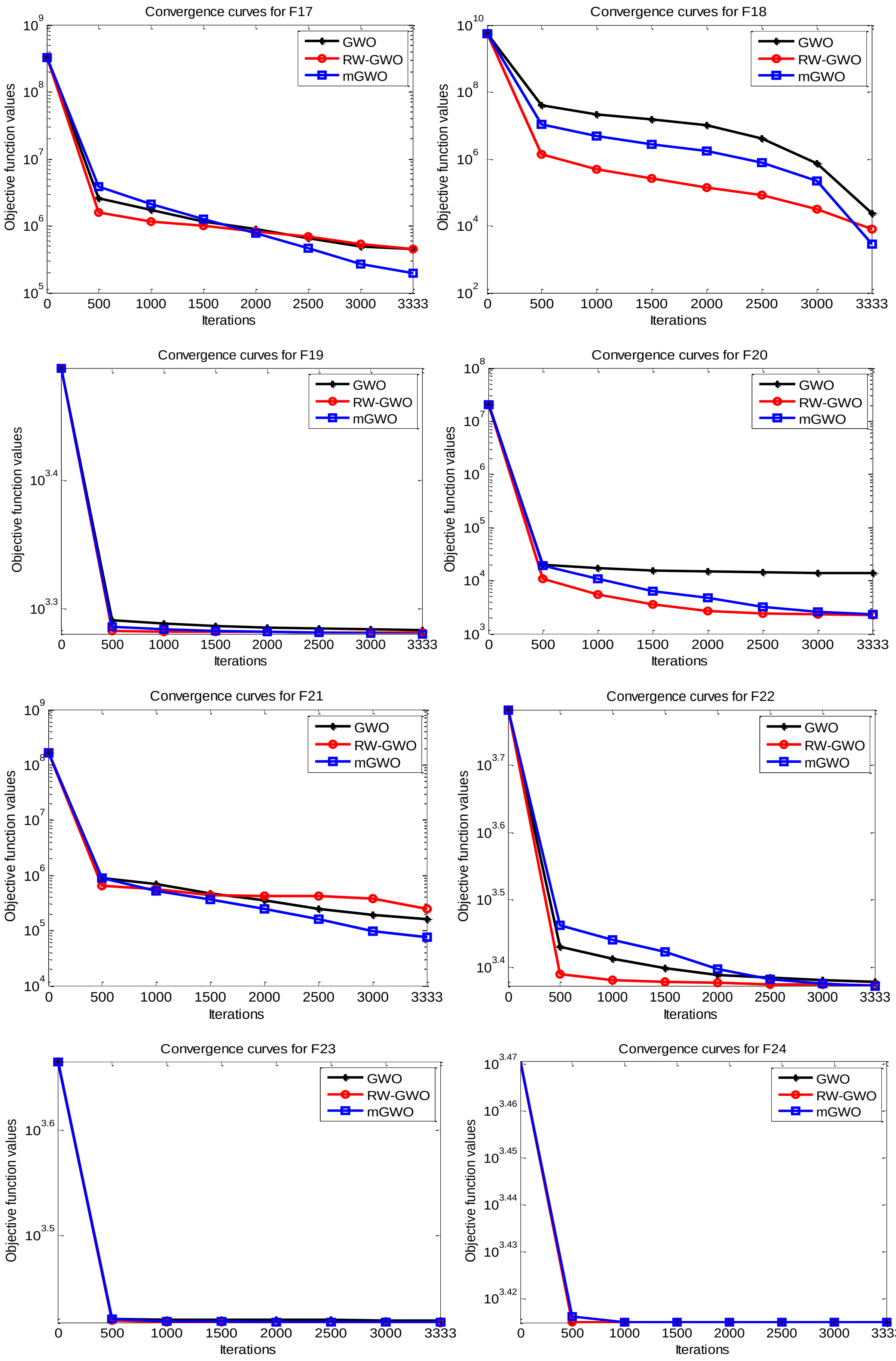


Fig 3.3. Convergence curves for 30-dimensional problems from F17 to F24 corresponding to alpha solution of each iteration

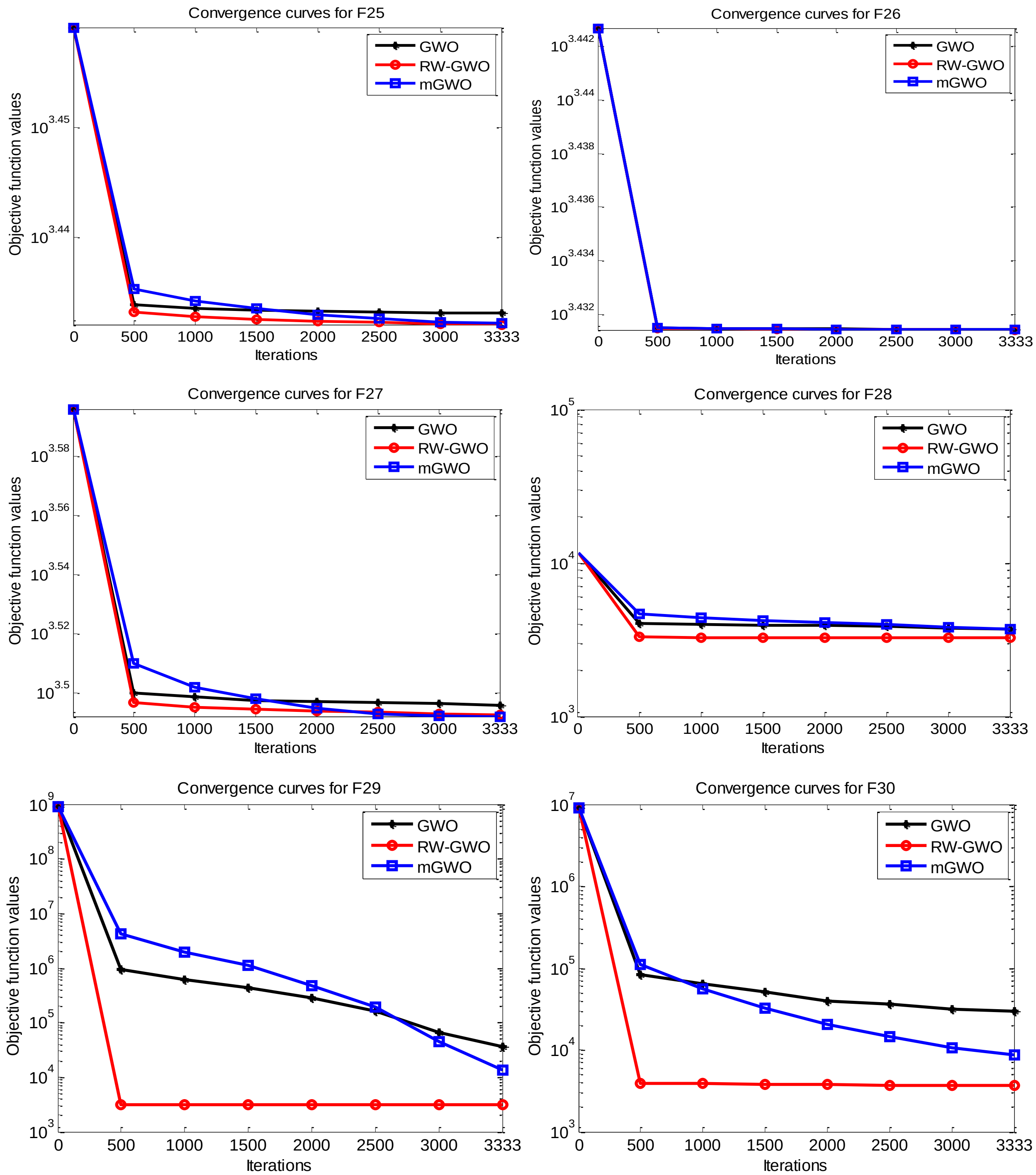


Fig 3.4. Convergence curves for 30-dimensional problems from F25 to F30 corresponding to alpha solution of each iteration

Chapter 4

A Modified Sine Cosine Algorithm with Opposition-based Learning for Unconstrained Optimization Problems

In this chapter an attempt is made to propose a new variant of Sine Cosine Algorithm.

4.1. Introduction

The literature on the SCA shows that in some cases, the algorithm suffers from the situation of skipping true solutions and stagnates at local optima. This happens when the mechanism of exploration and exploitation is faulty. Therefore, in the present chapter, an attempt has been made towards the eradication of these issues from the classical SCA by proposing a modified variant of SCA called Modified Sine Cosine Algorithm (m-SCA). In the m-SCA, two different strategies are employed. First, the opposition-based learning is used to generate the opposite candidate solutions which helps in avoiding the local optima during the search. Second, in order to maintain an appropriate balance between exploration and exploitation, the position update mechanism of the SCA is modified based on the cognitive component. The performance of the proposed m-SCA is tested on an unconstrained benchmark problem set given in IEEE CEC 2014 and the results are analyzed and compared with classical SCA.

The organization of this chapter is as follows: Section 4.2 provides a motivation and detailed description of the proposed m-SCA. Section 4.3 provides numerical experimentation, analysis and comparison of the m-SCA with classical SCA. Finally, the chapter is closed with concluding remarks in Section 4.4.

4.2. Proposed Modified Sine Cosine Algorithm (m-SCA)

4.2.1. Motivation

The efficiency, in terms of search ability, of metaheuristic algorithms depends on how well a

The content of this chapter is published in:

Gupta, S., & Deep, K. (2019). A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Systems with Applications, 119, 210-230. Elsevier.

metaheuristic achieves a balance between exploration and exploitation of the search space. In the classical SCA, each candidate solution uses the information of best candidate solution to update its state and the previous set of candidate solutions is replaced by new updated candidate solutions. In this process, only the best solution is saved for the next iteration. Therefore, there may be a possibility that the set of candidate solutions gets trapped in local optima due to insufficient diversity and improper guidance of search. Therefore, in the present chapter, an opposition-based learning is used to generate an opposite approximate to the current set of candidate solutions, so that the situation of local optima stagnation can be tackled. In order to provide promising direction of search and to enhance the collective strength of SCA, the personal best states of candidate solutions are used to integrate a cognitive component in the search equation of classical SCA. The concept of generating opposite numbers using opposition-based learning is presented as follows:

4.2.2. Opposite Numbers

Consider a point $x \in [x_{min}, x_{max}]$ where $x_{min}, x_{max} \in R$. Then an opposite point [187] of x denoted by x_{op} can be obtained as –

$$x_{op} = x_{min} + x_{max} - x \tag{4.1}$$

The definition of opposite numbers can be extended for higher dimension also [187, 188]. In $D-$ dimensional space the opposite point $X_{op} = \left({x_{op}}^1, {x_{op}}^2, \dots, {x_{op}}^D\right)$ of a point $X = (x^1, x^2, \dots, x^D) \in R^D$ can be calculated as –

$${x_{op}}^j = {x_{min}}^j + {x_{max}}^j - x^j, \qquad j = 1,2, \dots, D \tag{4.2}$$

where $X_{\min} = ({x_{min}}^1, {x_{min}}^2, \dots, {x_{min}}^D)$ and $X_{\max} = ({x_{max}}^1, {x_{max}}^2, \dots, {x_{max}}^D)$ are the lower and upper limits for any point $X \in R^D$.

In [189], it has been proved that finding the unknown optimal solution to the problem with a random direction along with its opposite estimate provides a higher chance as compared to pure random direction. The integration of opposite numbers in search algorithm is fruitful when the current obtained solution is far away from the optima especially when the optima is in opposite direction of a current solution.

The concept of opposite numbers is used by many researchers to enhance the ability of learn, search and optimize the metaheuristic algorithms. The literature on OBL and its benefits in metaheuristics can be found in [190].

4.2.3. Framework of the Modified Sine Cosine Algorithm

The proposed m-SCA utilizes two different strategies to update the solutions in each iteration. In the first strategy, the opposite population $\left(X_{op,i}\right)_{i=1}^{N}$ of candidate solutions $(X_i)_{i=1}^{N}$ is generated according to the perturbation or jumping rate (J_R) by using opposite numbers as described in Section 4.2.2 and then the best candidate solutions equal to the population size are selected from the population $\left(X_{op,i}\right)_{i=1}^{N} \cup (X_i)_{i=1}^{N}$. In the m-SCA, jumping rate is fixed at 0.1 to avoid the overflow of diversity because high jumping rate may skips true solutions during the search and this misleads the search process. Also, in this situation, the role of search equation of SCA will be deficient. This jumping rate helps to move out from local optima and provides a new directions of search. This new direction may have higher chance of locating global optima especially in those cases where the optima are just in opposite direction from current solution.

In second strategy, the personal best state of candidate solution is integrated in the search equation of classical SCA as a cognitive component. The classical search equation of SCA, given by

$$X_{i,t+1} = \begin{cases} X_{i,t} + A\sin(b)\left|CX_\alpha - X_{i,t}\right| & if\ r < 0.5 \\ X_{i,t} + A\cos(b)\left|CX_\alpha - X_{i,t}\right| & otherwise \end{cases} \tag{4.3}$$

is modified based on the personal best history of candidate solutions. The proposed position update search equation is defined as follows:

$$X_{i,t+1} = \begin{cases} X_{i,t} + \overbrace{A\sin(b)\left|CX_\alpha - x_{i,t}\right|}^{\text{Social Component}} + \overbrace{S_R\left(X_{i_{pbest}} - X_{i,t}\right)}^{\text{Cognitive Component}} & if\ r < 0.5 \\ X_{i,t} + \underbrace{A\cos(b)\left|CX_\alpha - x_{i,t}\right|}_{\text{Social Component}} + \underbrace{S_R\left(X_{i_{pbest}} - X_{i,t}\right)}_{\text{Cognitive Component}} & otherwise \end{cases} \tag{4.4}$$

where $X_{i,t+1}$ is the new updated state of a candidate solution X_i, $X_{i_{pbest}}$ is the personal best history of a candidate solution X_i . S_R is the coefficient which controls the effect of difference vector $\left(X_{i_{pbest}} - X_{i,t}\right)$. This parameter has been fixed as random number which is uniformly distributed within the interval $(0,1)$. The cognitive component $S_R\left(X_{i_{pbest}} - X_{i,t}\right)$ helps in local search and utilizes the information of personal best state of candidate solution which is saved in the memory of i^{th} candidate solution. In the proposed search equation (4.4), the cognitive component is used to exploit all the promising regions around the previously obtained best solutions. In equation (4.4) the second term on right hand side refers to the social component as it provides the direction towards the elite solution of population. The term $S_R\left(X_{i_{pbest}} - X_{i,t}\right)$ of the search equation (4.4) is referred to cognitive component as it provides the direction towards the individual's personal best history which is saved in the memory. Thus, the social and cognitive components maintain the balance between exploration and exploitation in the m-SCA.

The concept of generating opposite estimates using opposition-based learning and integration of personal best history in the search equation provide an enhanced global and local search in the m-SCA and help to alleviate from the situation of stagnation at local optima and increases the exploration ability of candidate solutions. The step wise description of the proposed Modified Sine Cosine Algorithm (m-SCA) is presented in Algorithm 4.1.

Algorithm 4.1. Modified Sine Cosine Algorithm (m-SCA)

1. $For \quad Min\ F(X) \quad s.t. \quad X_{\min} \leq X \leq X_{\max},\ X = (x_1, x_2, \ldots, x_D) \in R^D$

2. **Initialize** the population of candidate solutions X_i $(i = 1,2,\ldots,N)$

3. **Evaluate** the fitness of each candidate solution

4. **Select** the best solution X_α from the population of candidate solutions

5. **Initialize** the algorithm parameters:

T $-$ Maximum number of iterations

$J_R = 0.1$, Jumping rate.

6. Initialize the iteration count $t = 0$

7. $\boldsymbol{while}$ $t < T$

8. Generate a uniformly distributed random number k within the interval $(0,1)$

9. $\boldsymbol{if}$ $k < J_R$

10. Calculate the opposite population $(X_{op,i})_{i=1}^{N}$ of current population $(X_i)_{i=1}^{N}$ using eq. (4.2)

11. Evaluate the fitness of each opposite solution

12. Select the N best solutions from the population $(X_{op,i})_{i=1}^{N} \cup (X_i)_{i=1}^{N}$

13. Update the best solution X_α

14. $\boldsymbol{else}$

15. $\boldsymbol{for}$ each individual solution

16. Update the state with the help of equation (4.4)

17. Evaluate the fitness of updated candidate solution

18. Update the best solution X_α

19. $\boldsymbol{end\ for}$

20. $\boldsymbol{end\ if}$

21. $t = t + 1$

22. $\boldsymbol{end\ while}$

23. Return the best solution X_α.

4.2.4. Computational Complexity

Computational complexity of metaheuristic algorithm is very crucial to analyze the efficiency of the algorithm. The complexity of the algorithm primarily depends on the structure of the algorithm, population size, dimension of the decision vector and maximum number of iterations. Thus, the complexity of the proposed algorithm in terms of big-O notation can be calculated from the pseudo code as follows:

For classical SCA:

1. The classical SCA initializes the population of candidate solutions in $O(N \times D)$ time, where N the population size and D represent the dimension of the problem.
2. Fitness evaluation of population requires $O(N)$ time.
3. Selection of best candidate solution from the population requires $O(N)$ time.
4. Position update process in the classical SCA requires $O(N \times D)$ time.

In summary, the total computational time for the classical SCA is equal to $O(N \times D \times T)$ for maximum number of iterations T.

For m-SCA:

1. The m-SCA initializes the population of candidate solutions in $O(N \times D)$ time, where N the population size and D represent the dimension of the problem.
2. Fitness evaluation of population requires $O(N)$ time.
3. Selection of best candidate solution from the population requires $O(N)$ time.
4. Generation of opposite population from the current population requires $O(N \times D)$ time.
5. Fitness evaluation of opposite population requires $O(N)$ time.
6. Position update process through the proposed search equation (4.4) requires $O(N \times D)$ time.

In summary, the total computational time for the m-SCA is equal to $O(N \times D \times T)$ for maximum number of iterations T. Hence, by comparing the complexities of the classical SCA and m-SCA, it can be concluded that in terms of computational complexity both the algorithms are same.

4.3. Experimental Results and Discussion

4.3.1. Benchmark Functions and Parameter Setting

In the present chapter**,** the performance of the proposed m-SCA is evaluated on the same set of benchmark problems as given in IEEE CEC 2014 [185] and used in previous chapters. The dimension of the problems are fixed as 10 and 30 in our study and the termination criteria is adopted same as provided by IEEE CEC 2014. For all the test problems, the population size of solutions is taken as $3 \times D$ where D represents the dimension of the problem.

4.3.2. Analysis of the Results

In this section, the numerical results obtained by implementing classical SCA and m-SCA on IEEE CEC 2014 [185] benchmark problems are provided. The results are presented in the form of absolute error in objective function value. The better results are highlighted in bold face. For a feasible solution X and optima X^* to the problem F, the absolute error is calculated by $|F(X) - F(X^*)|$. The experiments are performed on 10 and 30-dimensional problems and various criteria, such as minimum, median, mean, maximum, standard deviation (STD), of the absolute errors in objective function values of test problems are presented. The performance of the m-SCA on different categories of benchmarks corresponding to 10 and 30-dimensonal problems is analyzed as follows:

The results for 10 dimension

The results for 10-dimensional problems are given in Table 4.1. From the results presented in this table it is observed that:

In all of the 10-dimensional unimodal problems from F1 to F3, the m-SCA outperforms classical SCA. The m-SCA provides a better minimum, maximum, mean, median and standard deviation value of error in objective functions in all the unimodal problems.

In all the 10-dimensional multimodal problems, the m-SCA outperforms classical SCA in all criteria except for standard deviation value in F5.

In all the 10-dimensional hybrid problems (F17-F22), the m-SCA outperforms classical SCA and provides better minimum, median, mean, maximum and standard deviation value of the errors in objective function.

For 10-dimensional composite problems F24, F26-F30, the m-SCA provides better results as compared to classical SCA in all the criteria. In problems F23 and F25, the m-SCA is better than the classical SCA for all the criteria except standard deviation.

Thus, it is concluded that the m-SCA performs better as compared to the classical SCA for 10-dimensional benchmark problems.

The results for 30 dimension

The results for 30-dimensional problems are given in Table 4.2. From the results presented in this table it is observed that:

In all of the 30-dimensional unimodal problems from F1 to F3, the m-SCA outperforms classical SCA. The m-SCA provides a better minimum, maximum, mean, median and standard deviation value of error in objective functions in all the unimodal problems.

In 30-dimensional multimodal problems, the m-SCA provides better results in all the criteria for F4, F7-F9, and F13-F15. In the remaining problems, the m-SCA is better than the classical SCA for all the criteria except standard deviation.

In all the 30-dimensional hybrid problems (F17-F22), the m-SCA outperforms classical SCA and provides better minimum, median, mean, maximum and standard deviation value of the errors in objective function.

For all the 30-dimensional composite problems (F23-F30), the m-SCA provides better results as compared to classical SCA in all the criteria.

Thus, it is concluded that the m-SCA performs better as compared to the classical SCA for 30-dimensional benchmark problems.

4.3.3. Statistical Analysis

To evaluate the improvement in the performance of the m-SCA against the classical SCA, a non-parametric Wilcoxon signed rank test [186] is used at 5% confidence interval. The statistical results are presented in Tables 4.3 and 4.4 corresponding to 10 and 30-dimensional IEEE CEC 2014 problems. In these tables, $'+/=/-'$ sign are used to indicate that the m-SCA is either significantly better, equal or worse than the classical SCA. The statistical results presented in Table 4.3 corresponding to 10-dimensional benchmark problems show that the m-SCA is significantly better than the classical SCA in all the problems. Similarly, Table 4.4 indicates that in all the 30-dimensional problems, the m-SCA is significantly better than the classical SCA. Thus, the statistical comparison demonstrate the superior performance of the m-SCA as compared to classical SCA.

4.3.4. Convergence Behavior

In this section, the convergence behavior of the classical SCA and m-SCA are compared through the convergence curves. These curves are shown in Figs 4.1 to 4.4 corresponding to the 30-dimensional IEEE CEC 2014 problems. The iterations are shown on horizontal axis and the vertical axis represents the objective function value. By inspecting the convergence history of elite candidate solution during the intermediate iterations of algorithms, it is obvious to conclude that the proposed m-SCA provides a better convergence rate as compared to classical SCA.

4.4. Concluding Remarks

In the chapter, an improved version of classical SCA called Modified Sine Cosine Algorithm (m-SCA) is introduced to solve global optimization problems. The m-SCA is proposed by finding out the drawbacks in classical SCA related to the insufficient diversity, skipping true solutions and stagnation to local optima. Therefore, an opposition-based learning is integrated into the search mechanism of classical SCA to provide a better and promising move at the time of stagnation. The search equations of classical SCA is modified by adding the cognitive component so that the promising direction of search can be provided to the candidate solutions. The experimental results on IEEE CEC 2014 benchmark set and their analysis through various metrics such as statistical analysis, convergence analysis demonstrate the superior search ability of the m-SCA as compared to the classical SCA. From the experimental analysis, it is also evident that the proposed m-SCA is able to maintain a more appropriate balance between exploration and exploitation as compared to the classical SCA.

Table 4.1. Error values in objective function obtained by classical SCA and m-SCA for 10-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal functions	**F1**	**SCA**	5.46E+06	2.06E+07	1.54E+07	8.10E+07	1.42E+07
		m-SCA	**3.57E+05**	**1.80E+06**	**1.66E+06**	**4.28E+06**	**1.03E+06**
	F2	**SCA**	3.54E+08	3.01E+09	2.25E+09	8.90E+09	2.15E+09
		m-SCA	**5.63E+04**	**2.17E+05**	**2.04E+05**	**4.23E+05**	**8.91E+04**
	F3	**SCA**	1.97E+03	2.12E+04	1.52E+04	7.26E+04	1.79E+04
		m-SCA	**6.39E+02**	**1.85E+03**	**1.38E+03**	**6.09E+03**	**1.23E+03**
Multimodal functions	**F4**	**SCA**	6.40E+01	3.88E+02	2.42E+02	1.55E+03	3.31E+02
		m-SCA	**1.04E+00**	**2.05E+01**	**1.83E+01**	**3.85E+01**	**1.17E+01**
	F5	**SCA**	2.02E+01	2.04E+01	2.04E+01	2.07E+01	**1.03E-01**
		m-SCA	**4.92E+00**	**1.94E+01**	**2.03E+01**	**2.04E+01**	2.92E+00
	F6	**SCA**	6.05E+00	9.76E+00	1.00E+01	1.15E+01	1.22E+00
		m-SCA	**6.89E-01**	**1.40E+00**	**1.28E+00**	**3.02E+00**	**5.69E-01**
	F7	**SCA**	6.66E+00	6.62E+01	7.58E+01	1.19E+02	3.19E+01
		m-SCA	**4.28E-01**	**6.98E-01**	**7.08E-01**	**9.06E-01**	**9.64E-02**
	F8	**SCA**	4.27E+01	7.41E+01	7.11E+01	1.09E+02	1.58E+01
		m-SCA	**4.22E+00**	**1.02E+01**	**9.68E+00**	**1.76E+01**	**2.67E+00**
	F9	**SCA**	2.60E+01	7.95E+01	8.36E+01	1.18E+02	2.11E+01
		m-SCA	**6.66E+00**	**1.12E+01**	**1.13E+01**	**1.78E+01**	**2.46E+00**
	F10	**SCA**	6.91E+02	1.32E+03	1.33E+03	1.76E+03	2.21E+02
		m-SCA	**9.27E+01**	**3.08E+02**	**2.93E+02**	**6.50E+02**	**1.35E+02**
	F11	**SCA**	1.20E+03	1.64E+03	1.72E+03	2.02E+03	2.52E+02
		m-SCA	**1.37E+02**	**4.32E+02**	**4.41E+02**	**7.62E+02**	**1.45E+02**
	F12	**SCA**	8.00E-01	1.53E+00	1.45E+00	2.65E+00	4.37E-01
		m-SCA	**2.62E-01**	**6.47E-01**	**6.17E-01**	**1.04E+00**	**1.74E-01**
	F13	**SCA**	5.74E-01	2.80E+00	2.85E+00	4.81E+00	1.25E+00
		m-SCA	**1.53E-01**	**2.43E-01**	**2.36E-01**	**4.37E-01**	**5.35E-02**
	F14	**SCA**	7.00E-01	1.45E+01	1.48E+01	2.77E+01	7.30E+00
		m-SCA	**1.19E-01**	**2.37E-01**	**2.35E-01**	**3.63E-01**	**6.22E-02**
	F15	**SCA**	6.49E+00	1.11E+04	7.92E+03	4.67E+04	1.23E+04
		m-SCA	**1.02E+00**	**1.78E+00**	**1.73E+00**	**2.77E+00**	**3.89E-01**
	F16	**SCA**	2.92E+00	3.90E+00	3.98E+00	4.37E+00	3.16E-01
		m-SCA	**1.59E+00**	**2.53E+00**	**2.58E+00**	**2.99E+00**	**2.77E-01**
Hybrid functions	**F17**	**SCA**	3.82E+03	3.44E+05	1.71E+05	1.88E+06	3.97E+05
		m-SCA	**8.84E+02**	**2.15E+03**	**1.82E+03**	**8.18E+03**	**1.17E+03**
	F18	**SCA**	8.46E+03	2.81E+06	5.83E+05	3.87E+07	6.49E+06
		m-SCA	**2.92E+02**	**2.37E+03**	**1.70E+03**	**7.32E+03**	**1.88E+03**
	F19	**SCA**	6.50E+00	1.13E+01	1.11E+01	2.27E+01	3.55E+00
		m-SCA	**1.56E+00**	**2.02E+00**	**2.00E+00**	**2.51E+00**	**1.83E-01**
	F20	**SCA**	3.78E+02	1.61E+05	5.16E+04	1.10E+06	2.42E+05
		m-SCA	**9.71E+01**	**6.54E+02**	**4.10E+02**	**2.72E+03**	**5.57E+02**
	F21	**SCA**	1.89E+03	1.23E+05	5.03E+04	8.77E+05	1.90E+05
		m-SCA	**5.24E+02**	**1.47E+03**	**1.30E+03**	**3.92E+03**	**6.09E+02**
	F22	**SCA**	5.40E+01	2.27E+02	2.32E+02	5.04E+02	1.12E+02
		m-SCA	**2.51E+01**	**3.04E+01**	**2.96E+01**	**4.13E+01**	**3.75E+00**

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Composite functions	**F23**	**SCA**	3.40E+02	4.32E+02	4.30E+02	5.36E+02	**4.62E+01**
		m-SCA	**1.35E+01**	**3.10E+02**	**3.30E+02**	**3.30E+02**	7.00E+01
	F24	**SCA**	1.48E+02	2.02E+02	2.03E+02	2.37E+02	2.43E+01
		m-SCA	**1.08E+02**	**1.16E+02**	**1.15E+02**	**1.23E+02**	**2.97E+00**
	F25	**SCA**	1.70E+02	2.06E+02	2.07E+02	2.22E+02	**7.83E+00**
		m-SCA	**1.27E+02**	**1.51E+02**	**1.52E+02**	**1.83E+02**	1.32E+01
	F26	**SCA**	1.01E+02	1.03E+02	1.02E+02	1.04E+02	8.56E-01
		m-SCA	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**3.34E-02**
	F27	**SCA**	1.23E+02	4.45E+02	4.61E+02	6.11E+02	1.16E+02
		m-SCA	**2.84E+00**	**4.58E+00**	**4.46E+00**	**8.52E+00**	**1.19E+00**
	F28	**SCA**	4.14E+02	5.49E+02	5.49E+02	7.39E+02	7.74E+01
		m-SCA	**2.22E+02**	**4.15E+02**	**4.35E+02**	**5.32E+02**	**7.28E+01**
	F29	**SCA**	5.59E+03	1.52E+05	4.72E+04	1.83E+06	3.14E+05
		m-SCA	**2.77E+02**	**5.04E+02**	**4.60E+02**	**1.03E+03**	**1.61E+02**
	F30	**SCA**	1.29E+03	8.13E+03	4.88E+03	4.71E+04	9.54E+03
		m-SCA	**1.18E+03**	**1.68E+03**	**1.68E+03**	**2.27E+03**	**2.41E+02**

Table 4.2. Error values in objective function obtained by classical SCA and m-SCA for 30-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal functions	**F1**	**SCA**	1.61E+08	3.93E+08	5.14E+08	1.27E+09	2.77E+08
		m-SCA	**9.86E+06**	**2.23E+07**	**2.26E+07**	**4.60E+07**	**6.35E+06**
	F2	**SCA**	1.58E+10	3.08E+10	3.35E+10	6.50E+10	1.28E+10
		m-SCA	**1.82E+07**	**7.00E+07**	**8.42E+07**	**2.16E+08**	**5.50E+07**
	F3	**SCA**	3.01E+04	5.05E+04	5.48E+04	1.20E+05	1.85E+04
		m-SCA	**1.60E+04**	**2.69E+04**	**2.70E+04**	**4.01E+04**	**6.53E+03**
Multimodal functions	**F4**	**SCA**	8.80E+02	2.22E+03	3.44E+03	1.67E+04	2.97E+03
		m-SCA	**1.37E+02**	**1.76E+02**	**1.81E+02**	**2.31E+02**	**2.31E+01**
	F5	**SCA**	2.08E+01	2.10E+01	**2.09E+01**	**2.10E+01**	**4.13E-02**
		m-SCA	**2.07E+01**	**2.09E+01**	**2.09E+01**	**2.10E+01**	6.41E-02
	F6	**SCA**	3.15E+01	3.79E+01	3.77E+01	4.20E+01	**2.85E+00**
		m-SCA	**8.20E+00**	**1.45E+01**	**1.45E+01**	**2.17E+01**	3.08E+00
	F7	**SCA**	1.29E+02	2.91E+02	3.49E+02	8.76E+02	1.68E+02
		m-SCA	**1.26E+00**	**1.86E+00**	**1.95E+00**	**3.19E+00**	**5.00E-01**
	F8	**SCA**	2.25E+02	2.78E+02	2.84E+02	3.87E+02	3.51E+01
		m-SCA	**7.74E+01**	**1.14E+02**	**1.13E+02**	**1.35E+02**	**1.17E+01**
	F9	**SCA**	2.22E+02	3.05E+02	3.13E+02	4.41E+02	4.18E+01
		m-SCA	**1.04E+02**	**1.32E+02**	**1.35E+02**	**1.82E+02**	**1.43E+01**
	F10	**SCA**	5.17E+03	6.71E+03	6.68E+03	7.29E+03	**4.23E+02**
		m-SCA	**2.37E+03**	**3.70E+03**	**3.73E+03**	**4.67E+03**	4.82E+02
	F11	**SCA**	6.37E+03	7.20E+03	7.16E+03	7.71E+03	**2.98E+02**
		m-SCA	**4.09E+03**	**4.94E+03**	**4.91E+03**	**5.69E+03**	3.70E+02
	F12	**SCA**	1.91E+00	2.46E+00	2.44E+00	2.97E+00	**2.75E-01**
		m-SCA	**9.18E-01**	**1.81E+00**	**1.81E+00**	**2.48E+00**	3.22E-01
	F13	**SCA**	3.05E+00	4.53E+00	4.80E+00	7.33E+00	1.25E+00
		m-SCA	**2.82E-01**	**3.89E-01**	**3.89E-01**	**4.82E-01**	**4.53E-02**
	F14	**SCA**	4.35E+01	1.01E+02	1.11E+02	2.42E+02	5.26E+01
		m-SCA	**1.55E-01**	**2.73E-01**	**2.71E-01**	**3.68E-01**	**3.84E-02**
	F15	**SCA**	1.21E+03	1.77E+04	3.99E+04	2.61E+05	5.34E+04
		m-SCA	**1.14E+01**	**1.53E+01**	**1.50E+01**	**1.79E+01**	**1.64E+00**
	F16	**SCA**	1.23E+01	1.31E+01	1.30E+01	1.36E+01	**2.39E-01**
		m-SCA	**1.13E+01**	**1.21E+01**	**1.21E+01**	**1.26E+01**	2.70E-01
Hybrid functions	**F17**	**SCA**	1.27E+06	1.49E+07	1.79E+07	6.65E+07	1.39E+07
		m-SCA	**1.31E+05**	**4.63E+05**	**5.39E+05**	**1.54E+06**	**3.30E+05**
	F18	**SCA**	6.47E+07	4.10E+08	8.74E+08	3.62E+09	9.30E+08
		m-SCA	**4.72E+04**	**1.57E+05**	**1.90E+05**	**5.87E+05**	**1.21E+05**
	F19	**SCA**	6.54E+01	1.40E+02	1.74E+02	5.05E+02	9.42E+01
		m-SCA	**1.18E+01**	**1.71E+01**	**1.76E+01**	**2.32E+01**	**2.84E+00**
	F20	**SCA**	7.76E+03	2.20E+04	3.26E+04	1.38E+05	2.85E+04
		m-SCA	**4.60E+03**	**1.20E+04**	**1.26E+04**	**2.01E+04**	**3.64E+03**
	F21	**SCA**	4.95E+05	3.23E+06	4.52E+06	2.40E+07	4.51E+06
		m-SCA	**2.16E+04**	**1.21E+05**	**1.18E+05**	**2.91E+05**	**4.95E+04**
	F22	**SCA**	6.17E+02	1.28E+03	1.29E+03	2.71E+03	4.33E+02
		m-SCA	**1.74E+02**	**2.55E+02**	**2.59E+02**	**3.75E+02**	**5.07E+01**

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Composite functions	**F23**	**SCA**	3.62E+02	4.05E+02	4.70E+02	1.05E+03	1.48E+02
		m-SCA	**3.19E+02**	**3.21E+02**	**3.21E+02**	**3.24E+02**	**1.43E+00**
	F24	**SCA**	2.01E+02	2.22E+02	2.32E+02	3.48E+02	3.18E+01
		m-SCA	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**4.63E-02**
	F25	**SCA**	**2.00E+02**	2.36E+02	2.41E+02	3.05E+02	1.93E+01
		m-SCA	**2.00E+02**	**2.00E+02**	**2.02E+02**	**2.13E+02**	**4.06E+00**
	F26	**SCA**	1.02E+02	1.05E+02	1.05E+02	1.09E+02	1.69E+00
		m-SCA	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.01E+02**	**4.48E-02**
	F27	**SCA**	5.18E+02	6.90E+02	7.30E+02	1.18E+03	1.74E+02
		m-SCA	**4.09E+02**	**4.26E+02**	**4.37E+02**	**4.94E+02**	**2.46E+01**
	F28	**SCA**	2.06E+03	3.00E+03	3.01E+03	4.34E+03	5.07E+02
		m-SCA	**6.23E+02**	**1.01E+03**	**1.14E+03**	**2.72E+03**	**4.32E+02**
	F29	**SCA**	9.94E+06	4.59E+07	5.13E+07	1.07E+08	2.05E+07
		m-SCA	**1.49E+04**	**3.66E+04**	**9.44E+04**	**7.59E+05**	**1.55E+05**
	F30	**SCA**	2.52E+05	7.23E+05	7.76E+05	2.00E+06	3.87E+05
		m-SCA	**1.75E+04**	**3.93E+04**	**4.05E+04**	**1.07E+05**	**1.52E+04**

Table 4.3. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 10-dimentional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	5.15E-10	+	**F16**	5.15E-10	+
F2	5.15E-10	+	**F17**	5.15E-10	+
F3	5.46E-10	+	**F18**	5.15E-10	+
F4	5.15E-10	+	**F19**	5.15E-10	+
F5	2.66E-07	+	**F20**	5.46E-10	+
F6	5.15E-10	+	**F21**	5.46E-10	+
F7	5.15E-10	+	**F22**	5.15E-10	+
F8	5.15E-10	+	**F23**	5.15E-10	+
F9	5.15E-10	+	**F24**	5.15E-10	+
F10	5.14E-10	+	**F25**	5.15E-10	+
F11	5.15E-10	+	**F26**	5.15E-10	+
F12	5.15E-10	+	**F27**	5.15E-10	+
F13	5.15E-10	+	**F28**	1.25E-09	+
F14	5.15E-10	+	**F29**	5.15E-10	+
F15	5.15E-10	+	**F30**	1.67E-09	+

Table 4.4. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 30-dimentional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	5.15E-10	+	**F16**	5.15E-10	+
F2	5.15E-10	+	**F17**	5.15E-10	+
F3	7.35E-10	+	**F18**	5.15E-10	+
F4	5.15E-10	+	**F19**	5.15E-10	+
F5	3.06E-03	+	**F20**	4.89E-08	+
F6	5.15E-10	+	**F21**	5.15E-10	+
F7	5.15E-10	+	**F22**	5.15E-10	+
F8	5.15E-10	+	**F23**	5.15E-10	+
F9	5.15E-10	+	**F24**	5.15E-10	+
F10	5.15E-10	+	**F25**	6.53E-10	+
F11	5.15E-10	+	**F26**	5.15E-10	+
F12	1.32E-09	+	**F27**	5.15E-10	+
F13	5.15E-10	+	**F28**	5.15E-10	+
F14	5.15E-10	+	**F29**	5.15E-10	+
F15	5.15E-10	+	**F30**	5.15E-10	+

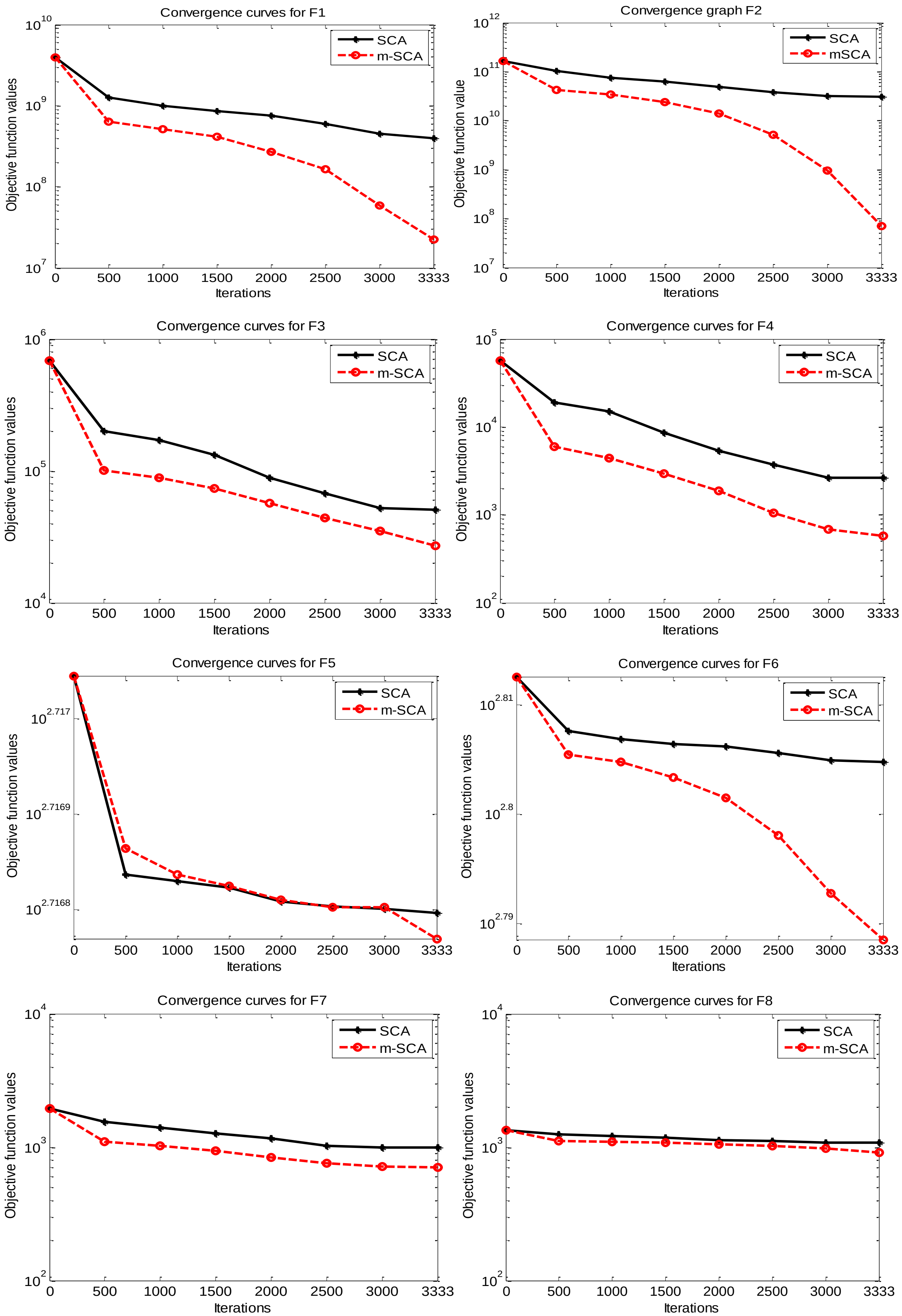


Fig 4.1. Convergence curves for 30-dimensional problems from F1 to F8 corresponding to elite candidate solution of each iteration

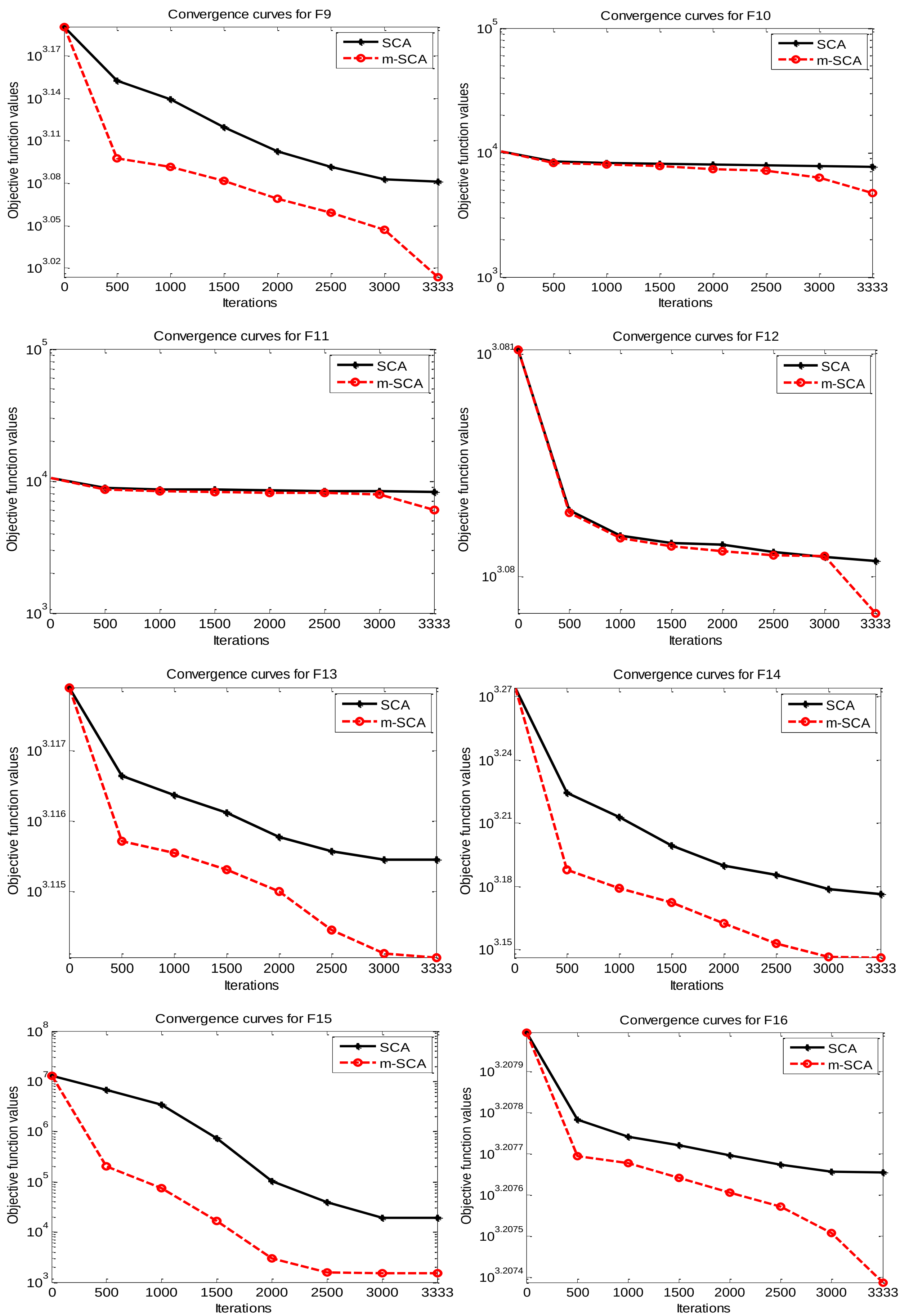


Fig 4.2. Convergence curves for 30-dimensional problems from F9 to F16 corresponding to elite candidate solution of each iteration

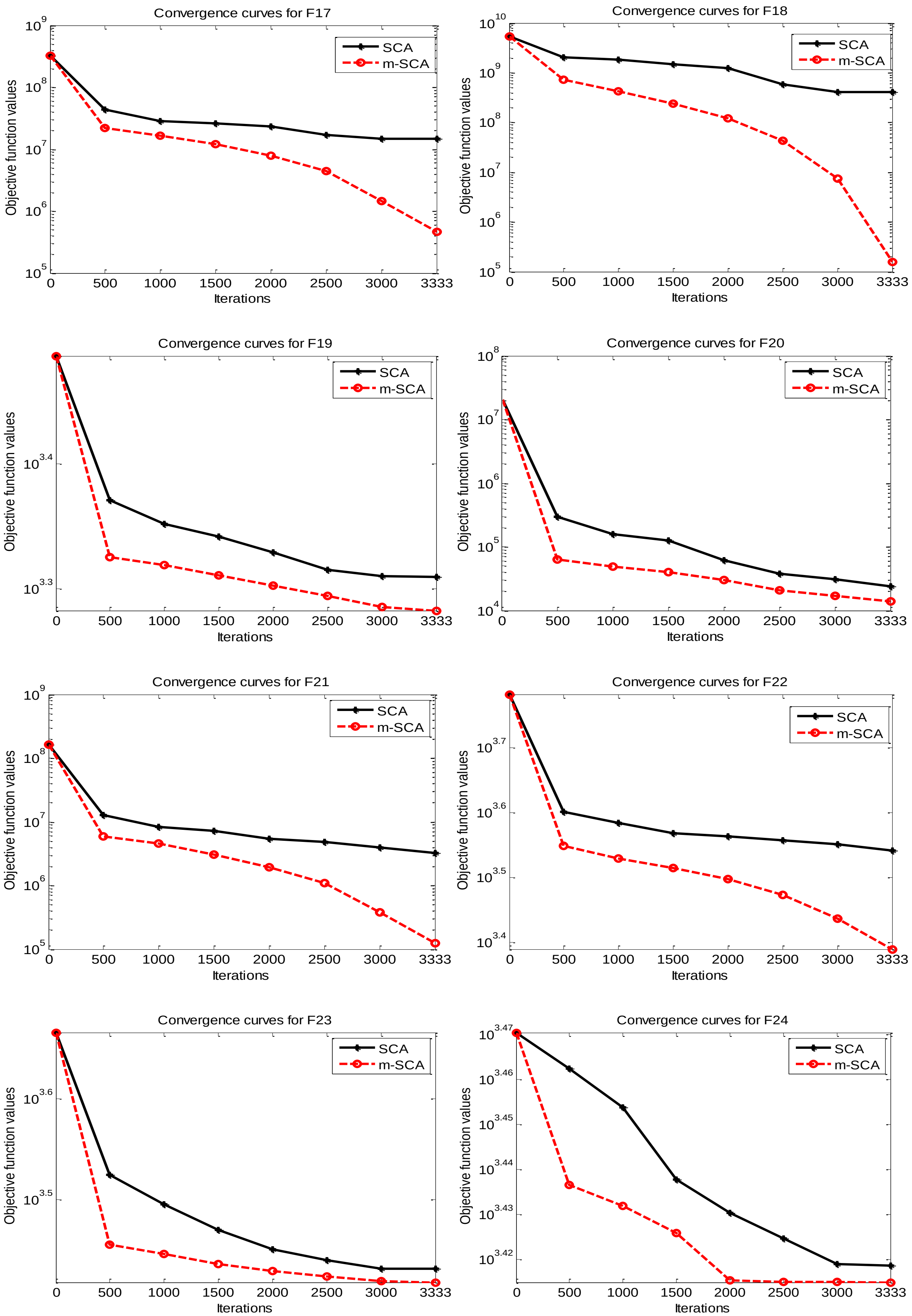


Fig 4.3. Convergence curves for 30-dimensional problems from F17 to F24 corresponding to elite candidate solution of each iteration

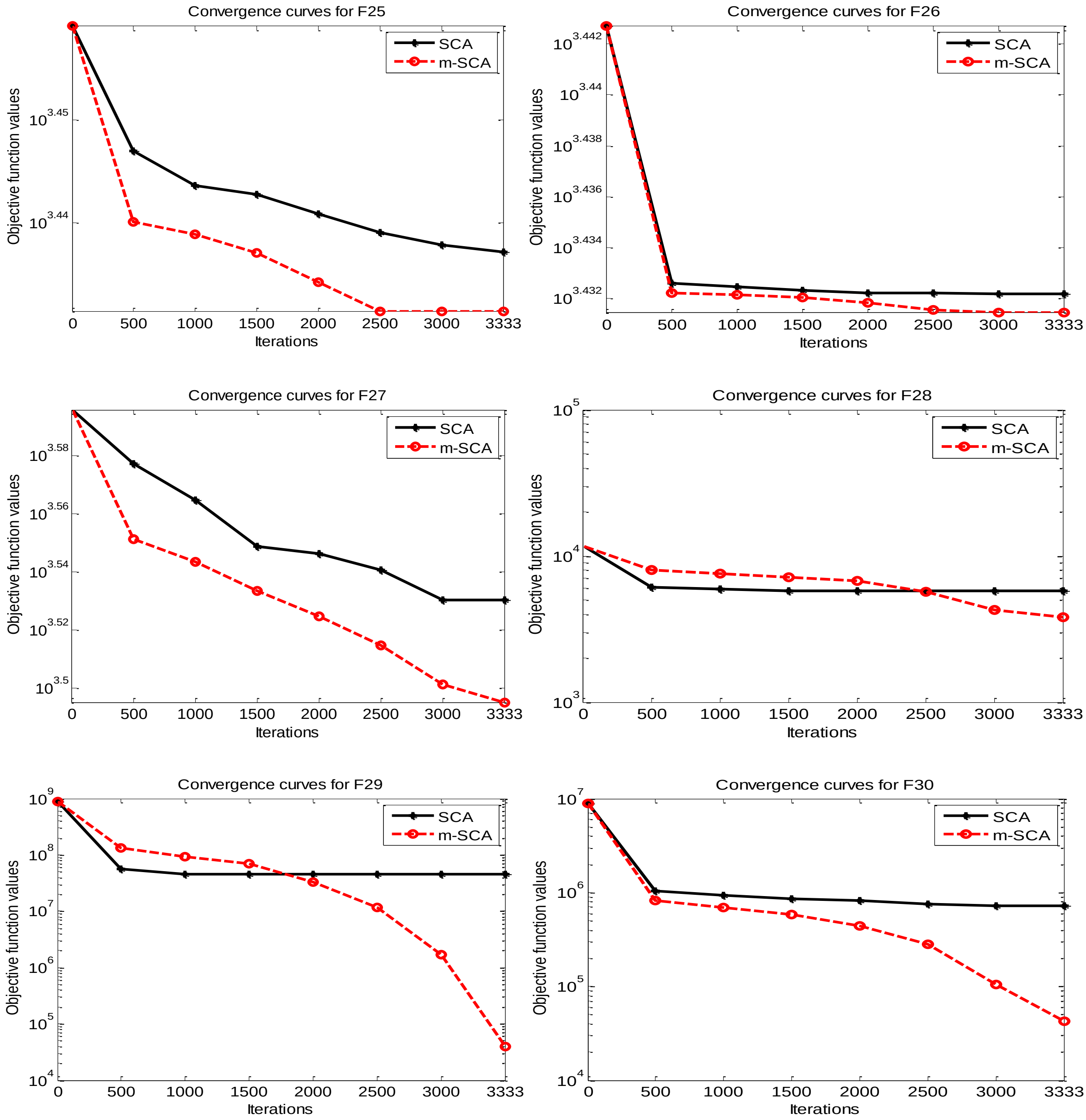


Fig 4.4. Convergence curves for 30-dimensional problems from F25 to F30 corresponding to elite candidate solution of each iteration

Chapter 5

Improved Sine Cosine Algorithm with Crossover Scheme for Unconstrained Optimization Problems

In this chapter another attempt is made to improve the performance of classical SCA.

5.1. Introduction

This chapter introduces a novel variant of classical SCA called ISCA which enhances the exploitation ability of candidate solutions and tries to establish an appropriate balance between exploration and exploitation. In order to accomplish these objectives, first the position update mechanism of the classical SCA is modified with the help of personal best state of candidate solutions and crossover operator. Then the greedy selection mechanism is used to avoid the divergence of candidate solutions from the discovered promising areas of the search space. The ISCA has been tested and compared with classical SCA on the IEEE CEC 2014 benchmark set of unconstrained optimization problems. In the chapter, the performance of the ISCA is also evaluated with respect to the m-SCA which was presented in the previous chapter.

The organization of the chapter is as follows: Section 5.2 provides a motivation behind proposing a new variant of classical SCA and detailed description of the proposed ISCA. Section 5.3 provides numerical experimentation, analysis and comparison of the ISCA with classical SCA. In Section 5.4, a comparison is shown between the m-SCA (introduced in Chapter 4) and ISCA. Finally, the chapter is closed with concluding remarks in Section 5.5.

5.2. Proposed Improved Sine Cosine Algorithm (ISCA)

5.2.1. Motivation and Proposed Strategies

Although, the SCA explores the search space very efficiently but like other population-based algorithms, it sometimes faces the high diversity (exploration). The high diversity may sometimes skips the true solutions of the problem if suitable balance between exploration and exploitation is not present in the algorithm.

The content of this chapter is published in:

Gupta, S., & Deep, K. (2019). Improved sine cosine algorithm with crossover scheme for global optimization. Knowledge-Based Systems, 165, 374-406. Elsevier.

In the search equations of SCA, a candidate solution is updated around the current state of a candidate solution and the area of a search space is decided by the coefficient A. In the prior iteration of the algorithm, candidate solutions are reallocated far from the current state as the coefficient A supports the exploration of the search space. The coefficient C also contributes to the exploration during the search process. Therefore, during the search process, in each iteration, a candidate solution losses its own features and always reallocate to a new position. The high diversity and the loss of personal best features by candidate solutions may skip the true solution and it is obvious that these skipped solutions may have a chance to provide better positions in the next iteration by exploiting the regions around them.

Therefore, to alleviate the above mentioned issues from classical SCA, some modifications have been done in the search strategy of classical SCA and a new variant called ISCA is proposed. The modifications which are introduced in classical SCA are presented as follows:

i. In the search equation of classical SCA, the fittest position X_α is replaced by the personal best states of candidate solutions in order to prevent from the situation of stagnation at local optima. The search process in the direction of personal best state of candidate solutions helps in exploring the more promising regions around the personal state of candidate solutions. This strategy also helps to escape from the situation when the elite candidate solution gets stuck in local optima and fails to guide the search.

ii. To integrate the personal best features of candidate solutions, a crossover is performed between the updated candidate solution through modified search mechanism and its personal best state obtained so far. This strategy helps to prevent the skipping of true solutions during the search.

iii. To reduce the high diversity and to provide a greedy direction of search, greedy selection is applied between updated and previous state of candidate solutions.

These modifications in the classical SCA are adopted to enhance the exploration as well as exploitation ability of candidate solutions with the help of personal best history of each candidate solution. The proposed strategies maintain the collaborative search in the algorithm. Greedy selection and crossover operator prevent the candidate solutions to diverge from discovered promising areas of the search space and to keep their best features respectively. The crossover which is used in the ISCA is described as follows:

Crossover

In the crossover process, two parent candidate solutions $X = (x_1, x_2, \dots, x_D)$ and $Y = (y_1, y_2, \dots, y_D)$ are hybridized to produce an offspring candidate solution $Z = (z_1, z_2, \dots, z_D)$ consisting the features of both the parent solutions. The crossover which is used in the present chapter is defined as follows:

$$z_j = \begin{cases} y_j & if\ r_j \le CR \\ x_j & otherwise \end{cases} \tag{5.1}$$

where CR is the crossover rate. In the chapter, the crossover probability CR is fixed to 0.3. The random number r_j is uniformly distributed in the interval (0,1).

5.2.2. Framework of Improved Sine Cosine Algorithm (ISCA)

Similar to the classical SCA, the ISCA also starts with the uniformly distributed population of candidate solutions generated randomly within the search space. After initialization, the search process for optima of the problem starts. The modified search equation introduced in ISCA is as follows:

$$\hat{X}_{i,t+1} = \begin{cases} X_{i,t} + \overbrace{A \sin(r_1) \left| CX_{i_{pbest}} - X_{i,t} \right|}^{\text{Cognitive Component}} + \overbrace{r_2 (X_\alpha - X_{i,t})}^{\text{Social Component}} & if\ r < 0.5 \\ X_{i,t} + \underbrace{A \cos(r_1) \left| CX_{i_{pbest}} - X_{i,t} \right|}_{\text{Cognitive Component}} + \underbrace{r_2 (X_\alpha - X_{i,t})}_{\text{Social Component}} & otherwise \end{cases} \tag{5.2}$$

where $X_{i,t}$ is the position of candidate solution at iteration t, $\hat{X}_{i,t+1}$ is the new updated position of the candidate solution $X_{i,t}$ at iteration $t+1$. $X_{i_{pbest}}$ is the personal best history of a candidate solution X_i, X_α represents the position of best candidate solution from the population, r_2 is a uniformly distributed random number between 0 and 1, rest of the parameters such as A, r_1, C and r are same as in classical SCA.

In equation (5.2), the second term on the right-hand side contributes the cognitive component in the search process and the third term contributes to the social component. The benefit of addressing these two components in the search process is to perform the local and global search during the search process. The cognitive and social components provide an efficient and promising directions to the current candidate solution by combining the directions along the solution's best and population's best states.

The candidate solution updated by equation (5.2) may have a chance to diverge from the current state when the search area provided by the coefficient A is very large. Therefore, in order to handle such situation and to integrate the personal best features of a candidate solution, the updated solution $\hat{X}_{i,t+1}$ is crossed with its personal best state $X_{i_{pbest}}$. This crossover is performed with the help of equation (5.1) and the new obtained position of a candidate solution is represented by $X_{i,t+1}$. After the crossover mechanism, the greedy selection is applied between the current and previous state of candidate solution. Greedy selection mechanism maintains the balance between exploration and exploitation in the search process and avoids the divergence of candidate solution from discovered promising areas of the search space. All the above steps are clearly summarized in Algorithm 5.1.

5.2.3. Computational Complexity

The time complexity of the proposed ISCA is discussed as follows:

1. The ISCA initializes the population of candidate solutions in $O(N \times D)$ time, where N is the population size and D represent the dimension of the problem.
2. Fitness evaluation of initial population requires $O(N)$ time.
3. Selection of best candidate solution from the population requires $O(N)$ time.
4. Position update of the candidate solutions and the crossover mechanism in the ISCA requires the $O(N \times D)$ time.
5. Fitness evaluation of updated candidate solutions requires $O(N)$ time.
6. The greedy selection requires an additional $O(N)$ time in the proposed ISCA.

In summary, the total computational complexity for the proposed ISCA is equal to $O(N \times D \times T)$ for maximum number of iterations T and this complexity is same as for classical SCA.

Algorithm 5.1. Improved Sine Cosine Algorithm (ISCA)

1. $For \quad Min\, F(X) \quad s.t. \quad X_{\min} \le X \le X_{\max},\ X = (x_1, x_2, \dots, x_D) \in R^D$
2. **Initialize** the population of candidate solutions X_i $(i = 1,2,\dots,N)$
3. **Evaluate** the fitness of each candidate solution
4. **Select** the best solution X_α from the population of candidate solutions
5. **Initialize** the algorithm parameters:

   T $-$ Maximum number of iterations

   $CR = 0.3$, crossover rate.
6. Store the personal best history of population as personal best position matrix as $\left[X_{i_{pbest}}\right]_{i=1}^{N} = [X_i]_{i=1}^{N}$
7. Initialize the iteration count $t = 0$
8. ***while*** $t < T$
9.   ***for*** each individual solution
10.   Update the position with the help of equation (5.2)
11.   Apply the crossover operator between personal best and updated state of candidate solution as described in equation (5.1)
12.   **Evaluate** the fitness of updated candidate solution
13.   ***for*** $i = 1,2,\dots,N$
14.   ***if*** $F(X_{i,t+1}) > F(X_{i,t})$
15. $X_{i,t+1} = X_{i,t}$
16.   ***end if***
17.   ***end for***
18.   Update the personal best state $X_{i_{pbest}}$ of candidate solution X_i
19.   Update the best solution X_α
20.   ***end of for***
21. $t = t + 1$
22. ***end of while***
23. Return the best solution X_α.

5.3. Experimental Results and Discussion

5.3.1. Benchmark Functions and Parameter Setting

In this chapter, the ISCA is evaluated on the same benchmark set as used in previous chapters and is given in IEEE CEC 2014 [185]. The experiments are conducted on 10 and 30-dimensional problems with population size $3 \times D$ and termination criteria $10^4 \times D$ function evaluations, where D is the dimension of the problem.

5.3.2. Analysis of the Results

In this section, the numerical results obtained by implementing classical SCA and proposed ISCA on IEEE CEC 2014 [185] benchmark problems are provided. The results are presented in the form of absolute error in objective function value and the better results are highlighted in bold face. For a feasible solution X and optima X^* to the problem F, the absolute error is calculated by $|F(X) - F(X^*)|$. The results are calculated in the form of various criteria, such as minimum, median, mean, maximum, standard deviation (STD), of the absolute errors in objective function values of test problems are presented. The performance of the ISCA on different categories of benchmarks corresponding to 10 and 30-dimensonal problems is analyzed as follows:

The results for 10 dimension

The results for 10-dimension problems are given in Table 5.1. On observing the results from the table, it is found that the ISCA performs better as compared to the classical SCA.

In 10-dimensional unimodal problems from F1 to F3, the ISCA provides better results in all the criteria as compared to the classical SCA.

In 10-dimensional multimodal problems F4, F6-F15, the ISCA provides better results in all the criteria as compared to the classical SCA. In problems F5 and F16, the ISCA is better than classical SCA for all criteria except standard deviation.

In all the 10-dimensional hybrid problems (F17-F22), the ISCA provides better results in terms of all the criteria as compared to classical SCA.

In all the 10 dimensional composite problems (F23-F30) except F25, F27 and F29, the ISCA provides better results in terms of all the criteria as compared to classical SCA. In F25 and F27, the ISCA is better than classical SCA for all criteria except standard deviation. In F29, the maximum error and standard deviation value is better in classical SCA while in other criteria, ISCA is better.

The results for 30 dimension

The results for 30-dimension problems are given in Table 5.2. On observing the results from the table, it is found that the ISCA performs better for 30-dimsnional problems also as compared to the classical SCA.

In all the 30-dimensional unimodal problems from F1 to F3, the ISCA provides better results in all the criteria as compared to the classical SCA.

In 30-dimensional multimodal problems F4, F6-F10, F13-F15, the ISCA is better than classical SCA in terms of all the criteria. In F5, the classical SCA provides better standard deviation than ISCA, while the provided minimum and median error is better in ISCA and in terms of mean and maximum error, both the algorithms are same. In problems F11, F12 and F16, the ISCA is better than classical SCA except for standard deviation.

In all the 30-dimensional hybrid problems (F17-F22), the ISCA provides better results in terms of all the criteria as compared to classical SCA.

In all the 30-dimensional composite problems (F23, F24 and F26-F30), the ISCA provides better results in terms of all the criteria as compared to classical SCA. In F25, the ISCA is better than the classical SCA for all the criteria except minimum error.

Hence, an overall analysis of the proposed ISCA on different category of benchmarks demonstrate the better exploration and exploitation ability of the ISCA as compared to the classical SCA. The experimental results also demonstrate that the proposed strategies in the ISCA establishes a more appropriate balance between exploration and exploitation as compared to the classical SCA.

5.3.3. Statistical Analysis

To ensure the improvement in ISCA, Wilcoxon signed rank test [186] has been applied with the same setting as used in previous chapters and the obtained statistical conclusions are presented in Tables 5.3 and 5.4. In the tables, $'+/=/-'$ sign are used to indicate that the ISCA is significantly better, equal or worse than the classical SCA. From the tables, it can be analyzed that in all the 30 problems, the ISCA is better than classical SCA corresponding to 10 as well as 30-dimensional benchmark problems. Overall, from the statistical results, it can be observed that the proposed ISCA has significantly improved the search efficiency and accuracy in obtaining the solution.

5.3.4. Convergence Behavior

The improvement in the solution for any problem can be analyzed through the elite solution of every iteration. Therefore, in this section, the history of the elite candidate solution is plotted in terms of convergence rate. In the Figs 5.1 to 5.4, the convergence history is shown for the 30-dimensional benchmark problems given in CEC 2014. The growth of iterations is shown on horizontal axis the vertical axis represents the average of objective function values. By inspecting the convergence curves, it is empirical to conclude that the proposed ISCA is better than classical SCA in terms of convergence rate.

5.4. Comparison Between m-SCA and ISCA Algorithms

In this section, the results of m-SCA which is proposed in chapter 4, and the proposed ISCA of current chapter are compared through Wilcoxon signed rank test. The comparison is conducted for 10 and 30-dimensional CEC 2014 problems. The statistical results are shown in Table 5.5. From the table, it is clear that in unimodal, multimodal, and hybrid problems, the ISCA outperform m-SCA. In composite problems for the dimension 10 m-SCA and for the dimension 30, ISCA performs better. Overall, the ISCA performs better as compared to m-SCA.

Although, both the variants m-SCA (proposed in the Chapter 4) and the ISCA (proposed in the current chapter) provide better results as compared to the classical SCA, but the comparison between these two variants is required on different categories of benchmark problems to elect the best performer. Therefore, in this section, a comparison is performed between the algorithms m-SCA and ISCA. For the sake of comparison, the results of the m-SCA are reproduced and presented in Tables 5.1 and 5.2 for 10 and 30-dimensional problems respectively.

The comparison for 10 dimension

The results for 10-dimension problems are given in Table 5.1. The description of the results on different category of benchmarks is as follows:

In 10-dimensional unimodal problems from F1 to F3, the ISCA provides better results in all the criteria as compared to the classical SCA and m-SCA.

In 10-dimensional multimodal problem F4, the ISCA provides better minimum and maximum value of errors than others and m-SCA provides better results in remaining criteria as compared to others. In F5, the ISCA is better than the classical SCA and m-SCA in terms of minimum, mean and maximum errors while for median error, m-SCA and for standard deviation, classical SCA is better than others. In problems F6, F7 and F14, ISCA provides better minimum, median and mean value of errors and the m-SCA provides better maximum and standard deviation of errors as compared to other algorithms. In problems F8, F10, F12 and F13, the ISCA provides better results in all the criteria as compared to m-SCA and classical SCA. In the problems F9, F11, F15 and F16, the ISCA provides better results in all the criteria except for standard deviation value as compared to classical SCA and m-SCA. In terms of standard deviation, the m-SCA is better in all these problems as compared to others.

In 10-dimensional hybrid problem F17, the ISCA is better than classical SCA and m-SCA in terms of minimum and maximum errors while in other criteria, m-SCA is better than others. In F18, the m-SCA is better than other in terms of all the criteria except for minimum error. For this problem, the minimum error is better in ISCA than others. In problems F19-F21, the ISCA is better in providing better minimum, median and mean value of errors as compared to the other algorithms and in other criteria, the m-SCA is better than others. In problem F22, the ISCA provides better results than other algorithms in all the criteria except for standard deviation which is better in m-SCA.

In 10-dimensional composite problem F23, the m-SCA is better than others in terms of minimum and median error. In terms of mean and maximum error both the algorithms m-SCA and ISCA are same and better than classical SCA while for the standard deviation, ISCA is better than others. In F24, the ISCA is better in terms of minimum and mean of errors as compared to classical SCA and m-SCA. In terms of median, maximum and standard deviation of errors, m-SCA is better than other algorithms. For F25, the minimum error is better in ISCA, mean, median and maximum error is better in m-SCA and standard deviation value is better in classical SCA as compared to other algorithms. In F26, the m-SCA and ISCA provide same results for all the criteria except for standard deviation. The standard deviation value is better in m-SCA as compared to the other algorithms for F26. In F27, except for the minimum error value, the m-SCA is better than the classical SCA and ISCA. The minimum is better in ISCA than the other algorithms for F27. In F28, the median value of errors is better in m-SCA while in terms of other criteria, the ISCA is better than the other algorithms. In F29, ISCA is better than others in terms of minimum and mean of error values while for the other criteria, the m-SCA is better than the others. In F30, except for standard deviation (which is better in m-SCA), the ISCA is better than the classical SCA and m-SCA.

The comparison for 30 dimension

The results for 30-dimension problems are given in Table 5.2. The description of the results on different category of benchmarks is as follows:

In 30-dimensional unimodal problems F1, the ISCA provides better results than classical SCA and m-SCA in all criteria except for standard deviation which is better in m-SCA. In problem F2, the m-SCA is better than other except for minimum error which is better in ISCA. In problem F3, the ISCA provides better results in terms of all the criteria as compared to classical SCA and m-SCA.

In 30-dimensional multimodal problem F4, F6, F8 and F10, the ISCA provides results in terms of all the criteria as compared to the classical SCA and m-SCA. In F5, the m-SCA and ISCA provides same results but better than classical SCA for all criteria except for standard deviation. The standard deviation value for F5 is better in classical SCA than other algorithms. In F7, the m-SCA is better than the others in terms of all criteria except for minimum error which is better in ISCA. In F9, the ISCA is better than the others in terms of all criteria except for standard deviation. The standard deviation is better in m-SCA than other algorithms for F9. In F11, F12 and F16, the ISCA is better than the others in terms of all the criteria except for standard deviation which is better in classical SCA. In F13 and F15, the ISCA is better than others except for maximum and standard deviation which is better in m-SCA. In F14, the m-SCA is better than classical SCA and ISCA for all the criteria.

In 30-dimensional hybrid problem F17-F20, the ISCA is better than classical SCA and m-SCA in terms of all the criteria. In F21, the m-SCA is better than others in terms of all the criteria. In problems F22, the ISCA is better than other algorithms in terms of all the criteria except for standard deviation which is better in m-SCA.

In 30-dimensional composite problem F23, the ISCA is better than others in terms of all criteria except for standard deviation which is better in m-SCA. In F24, m-SCA and ISCA provide same results in terms of minimum, median, mean and maximum error value but for the standard deviation ISCA is better than other algorithms. In F25, the m-SCA is better than others in terms for minimum, median and mean value of errors while in terms of other criteria, ISCA is better than the others. In F26, m-SCA and ISCA are same and better than classical SCA in terms of all the criteria except for standard deviation. The standard deviation value for F26 is better in m-SCA. In F27, F29 and F30, the ISCA is better than classical SCA and m-SCA in all the criteria. In F28, the ISCA is better than the others in all the criteria except for minimum error which is better in m-SCA.

Although, the numerical results demonstrate the differences in providing the results, but in order to analyze the best performer corresponding to each category of benchmarks and to make concrete conclusions about the significance of differences in the performance of algorithms, statistical comparison between m-SCA and ISCA is performed through Wilcoxon signed rank test. The comparison is performed using same parameter setting as used in previous chapters. The statistical results are listed in Table 5.5 and the best performer is listed in the same table. The convergence behavior of classical SCA, m-SCA and ISCA is also compared in the Figs 5.1 to 5.4.

In most of the problems except F2, and F5, the convergence rate is found better in ISCA as compared to classical SCA and m-SCA. In problems F2, the convergence of m-SCA is found better than ISCA at the end of iteration. In F5, only for some initial iterations (up to 800) the convergence is better in m-SCA as compared to ISCA. Overall, from all the comparison analysis the following remarks can be made:

1. In terms of worst time complexity calculated through big$-O$ notation, all the algorithms classical SCA, m-SCA and ISCA are identical.
2. On unimodal problems, the ISCA performs better as compared to the classical SCA and m-SCA.
3. On multimodal and hybrid problems, the ISCA is more successful as compared to the classical SCA and m-SCA.
4. On the set of composite problems, the m-SCA and ISCA are very competitive to each other.
5. The convergence rate is better in ISCA as compared to classical SCA and m-SCA.

5.5. Concluding Remarks

The present chapter introduces an improved version of SCA called ISCA with the help of crossover operator and personal best state of candidate solutions. In the ISCA, the search mechanism is modified by integrating the personal best state in place of the global best state to decide the search region around the personal best state of a candidate solution and to prevent from the situation of stagnation at local optima. In the search equation, global best or social component is also added with random step size to enhance the search towards the best available candidate solution. The greedy selection mechanism and crossover operator help to reduce the problem of inefficient diversity during the search process. To examine the impact of integrated strategies of the ISCA, it has been tested on standard IEEE CEC 2014 benchmark set. The analysis of results through various metrics such as statistical test and convergence behavior analysis ensures that the ISCA is better optimizer than classical SCA for all category of benchmark optimization problems.

The comparison between classical SCA, m-SCA (proposed in Chapter 4) and ISCA (proposed in current chapter) through various metrics shows that the ISCA is better optimizer to solve unimodal, multimodal and hybrid benchmark problems as compared to classical SCA and m-SCA. For the composite benchmark problems, both the algorithms m-SCA and ISCA are very competitive to each other and outperformed the classical SCA.

Table 5.1. Error values in objective function obtained by classical SCA, m-SCA and ISCA for 10-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal problems	**F1**	**SCA**	5.46E+06	2.06E+07	1.54E+07	8.10E+07	1.42E+07
		m-SCA	3.57E+05	1.80E+06	1.66E+06	4.28E+06	1.03E+06
		ISCA	**2.36E+04**	**3.47E+05**	**2.14E+05**	**3.48E+06**	**5.26E+05**
	F2	**SCA**	3.54E+08	3.01E+09	2.25E+09	8.90E+09	2.15E+09
		m-SCA	5.63E+04	2.17E+05	2.04E+05	4.23E+05	8.91E+04
		ISCA	**1.35E+03**	**6.02E+03**	**3.39E+03**	**1.55E+04**	**4.70E+03**
	F3	**SCA**	1.97E+03	2.12E+04	1.52E+04	7.26E+04	1.79E+04
		m-SCA	6.39E+02	1.85E+03	1.38E+03	6.09E+03	1.23E+03
		ISCA	**1.45E+01**	**2.76E+02**	**1.20E+02**	**1.09E+03**	**2.83E+02**
Multimodal problems	**F4**	**SCA**	6.40E+01	3.88E+02	2.42E+02	1.55E+03	3.31E+02
		m-SCA	1.04E+00	**2.05E+01**	**1.83E+01**	3.85E+01	**1.17E+01**
		ISCA	**6.56E-01**	2.76E+01	3.50E+01	**3.52E+01**	1.30E+01
	F5	**SCA**	2.02E+01	2.04E+01	2.04E+01	2.07E+01	**1.03E-01**
		m-SCA	4.92E+00	**1.94E+01**	2.03E+01	2.04E+01	2.92E+00
		ISCA	**2.31E-01**	1.98E+01	**2.02E+01**	**2.03E+01**	2.80E+00
	F6	**SCA**	6.05E+00	9.76E+00	1.00E+01	1.15E+01	1.22E+00
		m-SCA	6.89E-01	1.40E+00	1.28E+00	**3.02E+00**	**5.69E-01**
		ISCA	**1.52E-01**	**1.04E+00**	**7.87E-01**	3.53E+00	7.81E-01
	F7	**SCA**	6.66E+00	6.62E+01	7.58E+01	1.19E+02	3.19E+01
		m-SCA	4.28E-01	6.98E-01	7.08E-01	**9.06E-01**	**9.64E-02**
		ISCA	**1.17E-01**	**4.21E-01**	**3.63E-01**	1.20E+00	1.98E-01
	F8	**SCA**	4.27E+01	7.41E+01	7.11E+01	1.09E+02	1.58E+01
		m-SCA	4.22E+00	1.02E+01	9.68E+00	1.76E+01	2.67E+00
		ISCA	**2.94E-03**	**1.15E+00**	**1.01E+00**	**3.98E+00**	**8.59E-01**
	F9	**SCA**	2.60E+01	7.95E+01	8.36E+01	1.18E+02	2.11E+01
		m-SCA	6.66E+00	1.12E+01	1.13E+01	1.78E+01	**2.46E+00**
		ISCA	**2.99E+00**	**7.41E+00**	**6.44E+00**	**1.40E+01**	2.74E+00
	F10	**SCA**	6.91E+02	1.32E+03	1.33E+03	1.76E+03	2.21E+02
		m-SCA	9.27E+01	3.08E+02	2.93E+02	6.50E+02	1.35E+02
		ISCA	**4.79E+00**	**4.12E+01**	**2.30E+01**	**1.56E+02**	**4.50E+01**
	F11	**SCA**	1.20E+03	1.64E+03	1.72E+03	2.02E+03	2.52E+02
		m-SCA	1.37E+02	4.32E+02	4.41E+02	7.62E+02	**1.45E+02**
		ISCA	**1.28E+00**	**1.87E+02**	**1.44E+02**	**7.09E+02**	1.65E+02
	F12	**SCA**	8.00E-01	1.53E+00	1.45E+00	2.65E+00	4.37E-01
		m-SCA	2.62E-01	6.47E-01	6.17E-01	1.04E+00	1.74E-01
		ISCA	**1.37E-01**	**3.52E-01**	**3.45E-01**	**6.03E-01**	**1.22E-01**
	F13	**SCA**	5.74E-01	2.80E+00	2.85E+00	4.81E+00	1.25E+00
		m-SCA	1.53E-01	2.43E-01	2.36E-01	4.37E-01	5.35E-02
		ISCA	**8.08E-02**	**1.70E-01**	**1.64E-01**	**3.12E-01**	**4.60E-02**
	F14	**SCA**	7.00E-01	1.45E+01	1.48E+01	2.77E+01	7.30E+00
		m-SCA	1.19E-01	2.37E-01	2.35E-01	**3.63E-01**	**6.22E-02**
		ISCA	**6.70E-02**	**1.87E-01**	**1.90E-01**	6.07E-01	8.03E-02
	F15	**SCA**	6.49E+00	1.11E+04	7.92E+03	4.67E+04	1.23E+04
		m-SCA	1.02E+00	1.78E+00	1.73E+00	2.77E+00	**3.89E-01**
		ISCA	**6.30E-01**	**1.34E+00**	**1.25E+00**	**2.29E+00**	4.16E-01

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
	F16	**SCA**	2.92E+00	3.90E+00	3.98E+00	4.37E+00	3.16E-01
		m-SCA	1.59E+00	2.53E+00	2.58E+00	2.99E+00	**2.77E-01**
		ISCA	**3.54E-01**	**1.80E+00**	**1.81E+00**	**2.83E+00**	5.73E-01
Hybrid problems	**F17**	**SCA**	3.82E+03	3.44E+05	1.71E+05	1.88E+06	3.97E+05
		m-SCA	8.84E+02	**2.15E+03**	**1.82E+03**	8.18E+03	**1.17E+03**
		ISCA	**1.67E+02**	2.68E+03	1.88E+03	**7.95E+03**	2.35E+03
	F18	**SCA**	8.46E+03	2.81E+06	5.83E+05	3.87E+07	6.49E+06
		m-SCA	2.92E+02	**2.37E+03**	**1.70E+03**	**7.32E+03**	**1.88E+03**
		ISCA	**4.78E+01**	4.92E+03	3.12E+03	1.62E+04	5.03E+03
	F19	**SCA**	6.50E+00	1.13E+01	1.11E+01	2.27E+01	3.55E+00
		m-SCA	1.56E+00	2.02E+00	2.00E+00	**2.51E+00**	**1.83E-01**
		ISCA	**1.15E+00**	**1.59E+00**	**1.57E+00**	2.78E+00	2.36E-01
	F20	**SCA**	3.78E+02	1.61E+05	5.16E+04	1.10E+06	2.42E+05
		m-SCA	9.71E+01	6.54E+02	4.10E+02	**2.72E+03**	**5.57E+02**
		ISCA	**7.17E+00**	**3.82E+02**	**3.76E+01**	4.37E+03	9.61E+02
	F21	**SCA**	1.89E+03	1.23E+05	5.03E+04	8.77E+05	1.90E+05
		m-SCA	5.24E+02	1.47E+03	1.30E+03	**3.92E+03**	**6.09E+02**
		ISCA	**5.59E+01**	**1.41E+03**	**4.05E+02**	5.18E+03	1.70E+03
	F22	**SCA**	5.40E+01	2.27E+02	2.32E+02	5.04E+02	1.12E+02
		m-SCA	2.51E+01	3.04E+01	2.96E+01	4.13E+01	**3.75E+00**
		ISCA	**1.27E+00**	**1.69E+01**	**2.16E+01**	**4.02E+01**	9.66E+00
Composite problems	**F23**	**SCA**	3.40E+02	4.32E+02	4.30E+02	5.36E+02	4.62E+01
		m-SCA	**1.35E+01**	**3.10E+02**	**3.30E+02**	**3.30E+02**	7.00E+01
		ISCA	3.30E+02	3.30E+02	**3.30E+02**	**3.30E+02**	**2.29E-02**
	F24	**SCA**	1.48E+02	2.02E+02	2.03E+02	2.37E+02	2.43E+01
		m-SCA	1.08E+02	**1.16E+02**	1.15E+02	**1.23E+02**	**2.97E+00**
		ISCA	**1.07E+02**	1.17E+02	**1.13E+02**	2.00E+02	1.35E+01
	F25	**SCA**	1.70E+02	2.06E+02	2.07E+02	2.22E+02	**7.83E+00**
		m-SCA	1.27E+02	**1.51E+02**	**1.52E+02**	**1.83E+02**	1.32E+01
		ISCA	**1.21E+02**	1.83E+02	1.99E+02	2.02E+02	2.84E+01
	F26	**SCA**	1.01E+02	1.03E+02	1.02E+02	1.04E+02	8.56E-01
		m-SCA	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	**3.34E-02**
		ISCA	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.00E+02**	3.93E-02
	F27	**SCA**	1.23E+02	4.45E+02	4.61E+02	6.11E+02	1.16E+02
		m-SCA	2.84E+00	**4.58E+00**	**4.46E+00**	**8.52E+00**	**1.19E+00**
		ISCA	**2.38E+00**	2.79E+02	3.45E+02	4.01E+02	1.66E+02
	F28	**SCA**	4.14E+02	5.49E+02	5.49E+02	7.39E+02	7.74E+01
		m-SCA	2.22E+02	**4.15E+02**	4.35E+02	5.32E+02	7.28E+01
		ISCA	**2.04E+02**	4.16E+02	**3.78E+02**	**5.07E+02**	**6.11E+01**
	F29	**SCA**	5.59E+03	1.52E+05	4.72E+04	1.83E+06	3.14E+05
		m-SCA	2.77E+02	**5.04E+02**	4.60E+02	**1.03E+03**	**1.61E+02**
		ISCA	**2.50E+02**	1.09E+05	**4.01E+02**	2.10E+06	4.42E+05
	F30	**SCA**	1.29E+03	8.13E+03	4.88E+03	4.71E+04	9.54E+03
		m-SCA	1.18E+03	1.68E+03	1.68E+03	2.27E+03	**2.41E+02**
		ISCA	**4.84E+02**	**6.71E+02**	**5.94E+02**	**2.08E+03**	2.72E+02

Table 5.2. Error values in objective function obtained by classical SCA, m-SCA and ISCA for 30-dimentional IEEE CEC 2014 benchmark problems

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
Unimodal problems	F1	SCA	1.61E+08	3.93E+08	5.14E+08	1.27E+09	2.77E+08
		m-SCA	9.86E+06	2.23E+07	2.26E+07	4.60E+07	**6.35E+06**
		ISCA	**4.57E+06**	**1.35E+07**	**1.50E+07**	**3.75E+07**	6.57E+06
	F2	SCA	1.58E+10	3.08E+10	3.35E+10	6.50E+10	1.28E+10
		m-SCA	1.82E+07	**7.00E+07**	**8.42E+07**	**2.16E+08**	**5.50E+07**
		ISCA	**2.83E+06**	3.04E+08	3.18E+08	8.45E+08	1.82E+08
	F3	SCA	3.01E+04	5.05E+04	5.48E+04	1.20E+05	1.85E+04
		m-SCA	1.60E+04	2.69E+04	2.70E+04	4.01E+04	6.53E+03
		ISCA	**5.55E+02**	**1.84E+03**	**2.44E+03**	**6.85E+03**	**1.57E+03**
Multimodal problems	F4	SCA	8.80E+02	2.22E+03	3.44E+03	1.67E+04	2.97E+03
		m-SCA	1.37E+02	1.76E+02	1.81E+02	2.31E+02	2.31E+01
		ISCA	**8.94E+01**	**1.43E+02**	**1.44E+02**	**2.23E+02**	**3.29E+01**
	F5	SCA	2.08E+01	2.10E+01	2.09E+01	2.10E+01	**4.13E-02**
		m-SCA	**2.07E+01**	**2.09E+01**	**2.09E+01**	**2.10E+01**	6.41E-02
		ISCA	**2.07E+01**	**2.09E+01**	**2.09E+01**	**2.10E+01**	6.11E-02
	F6	SCA	3.15E+01	3.79E+01	3.77E+01	4.20E+01	2.85E+00
		m-SCA	8.20E+00	1.45E+01	1.45E+01	2.17E+01	3.08E+00
		ISCA	**4.82E+00**	**8.96E+00**	**8.82E+00**	**1.33E+01**	**1.96E+00**
	F7	SCA	1.29E+02	2.91E+02	3.49E+02	8.76E+02	1.68E+02
		m-SCA	1.26E+00	**1.86E+00**	**1.95E+00**	**3.19E+00**	**5.00E-01**
		ISCA	**1.19E+00**	3.71E+00	4.46E+00	1.13E+01	2.57E+00
	F8	SCA	2.25E+02	2.78E+02	2.84E+02	3.87E+02	3.51E+01
		m-SCA	7.74E+01	1.14E+02	1.13E+02	1.35E+02	1.17E+01
		ISCA	**1.66E+01**	**2.97E+01**	**2.99E+01**	**5.34E+01**	**7.42E+00**
	F9	SCA	2.22E+02	3.05E+02	3.13E+02	4.41E+02	4.18E+01
		m-SCA	1.04E+02	1.32E+02	1.35E+02	1.82E+02	**1.43E+01**
		ISCA	**3.08E+01**	**5.91E+01**	**6.05E+01**	**9.33E+01**	1.47E+01
	F10	SCA	5.17E+03	6.71E+03	6.68E+03	7.29E+03	4.23E+02
		m-SCA	2.37E+03	3.70E+03	3.73E+03	4.67E+03	4.82E+02
		ISCA	**1.67E+02**	**5.33E+02**	**5.89E+02**	**1.24E+03**	**2.49E+02**
	F11	SCA	6.37E+03	7.20E+03	7.16E+03	7.71E+03	**2.98E+02**
		m-SCA	4.09E+03	4.94E+03	4.91E+03	5.69E+03	3.70E+02
		ISCA	**1.37E+03**	**2.51E+03**	**2.49E+03**	**3.56E+03**	5.63E+02
	F12	SCA	1.91E+00	2.46E+00	2.44E+00	2.97E+00	**2.75E-01**
		m-SCA	9.18E-01	1.81E+00	1.81E+00	2.48E+00	3.22E-01
		ISCA	**7.93E-01**	**1.60E+00**	**1.59E+00**	**2.27E+00**	3.93E-01
	F13	SCA	3.05E+00	4.53E+00	4.80E+00	7.33E+00	1.25E+00
		m-SCA	2.82E-01	3.89E-01	3.89E-01	**4.82E-01**	**4.53E-02**
		ISCA	**1.98E-01**	**3.54E-01**	**3.50E-01**	5.54E-01	6.00E-02
	F14	SCA	4.35E+01	1.01E+02	1.11E+02	2.42E+02	5.26E+01
		m-SCA	**1.55E-01**	**2.73E-01**	**2.71E-01**	**3.68E-01**	**3.84E-02**
		ISCA	2.14E-01	7.19E-01	6.44E-01	8.76E-01	2.25E-01
	F15	SCA	1.21E+03	1.77E+04	3.99E+04	2.61E+05	5.34E+04
		m-SCA	1.14E+01	1.53E+01	1.50E+01	**1.79E+01**	**1.64E+00**
		ISCA	**6.77E+00**	**1.34E+01**	**1.31E+01**	1.96E+01	2.83E+00

	Function	Algorithm	Minimum	Median	Mean	Maximum	STD
	F16	**SCA**	1.23E+01	1.31E+01	1.30E+01	1.36E+01	**2.39E-01**
		m-SCA	1.13E+01	1.21E+01	1.21E+01	1.26E+01	2.70E-01
		ISCA	**9.48E+00**	**1.08E+01**	**1.08E+01**	**1.21E+01**	5.64E-01
Hybrid problems	**F17**	**SCA**	1.27E+06	1.49E+07	1.79E+07	6.65E+07	1.39E+07
		m-SCA	1.31E+05	4.63E+05	5.39E+05	1.54E+06	3.30E+05
		ISCA	**5.52E+04**	**2.56E+05**	**3.36E+05**	**1.41E+06**	**2.58E+05**
	F18	**SCA**	6.47E+07	4.10E+08	8.74E+08	3.62E+09	9.30E+08
		m-SCA	4.72E+04	1.57E+05	1.90E+05	5.87E+05	1.21E+05
		ISCA	**1.36E+03**	**3.02E+03**	**3.61E+03**	**9.07E+03**	**1.82E+03**
	F19	**SCA**	6.54E+01	1.40E+02	1.74E+02	5.05E+02	9.42E+01
		m-SCA	1.18E+01	1.71E+01	1.76E+01	2.32E+01	2.84E+00
		ISCA	**6.79E+00**	**1.23E+01**	**1.20E+01**	**2.27E+01**	**2.65E+00**
	F20	**SCA**	7.76E+03	2.20E+04	3.26E+04	1.38E+05	2.85E+04
		m-SCA	4.60E+03	1.20E+04	1.26E+04	2.01E+04	3.64E+03
		ISCA	**5.13E+02**	**1.65E+03**	**2.31E+03**	**9.38E+03**	**1.98E+03**
	F21	**SCA**	4.95E+05	3.23E+06	4.52E+06	2.40E+07	4.51E+06
		m-SCA	**2.16E+04**	**1.21E+05**	**1.18E+05**	**2.91E+05**	**4.95E+04**
		ISCA	4.99E+04	1.32E+05	1.79E+05	5.41E+05	1.30E+05
	F22	**SCA**	6.17E+02	1.28E+03	1.29E+03	2.71E+03	4.33E+02
		m-SCA	1.74E+02	2.55E+02	2.59E+02	3.75E+02	**5.07E+01**
		ISCA	**7.30E+01**	**1.89E+02**	**2.11E+02**	**3.57E+02**	5.95E+01
Composite problems	**F23**	**SCA**	3.62E+02	4.05E+02	4.70E+02	1.05E+03	1.48E+02
		m-SCA	3.19E+02	3.21E+02	3.21E+02	3.24E+02	**1.43E+00**
		ISCA	**3.16E+02**	**3.18E+02**	**3.18E+02**	**3.21E+02**	1.18E+00
	F24	**SCA**	2.01E+02	2.22E+02	2.32E+02	3.48E+02	3.18E+01
		m-SCA	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	4.63E-02
		ISCA	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.00E+02**	**2.17E-03**
	F25	**SCA**	**2.00E+02**	2.36E+02	2.41E+02	3.05E+02	1.93E+01
		m-SCA	**2.00E+02**	**2.00E+02**	**2.02E+02**	2.13E+02	4.06E+00
		ISCA	2.04E+02	2.07E+02	2.07E+02	**2.11E+02**	**1.61E+00**
	F26	**SCA**	1.02E+02	1.05E+02	1.05E+02	1.09E+02	1.69E+00
		m-SCA	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.01E+02**	**4.48E-02**
		ISCA	**1.00E+02**	**1.00E+02**	**1.00E+02**	**1.01E+02**	6.49E-02
	F27	**SCA**	5.18E+02	6.90E+02	7.30E+02	1.18E+03	1.74E+02
		m-SCA	4.09E+02	4.26E+02	4.37E+02	4.94E+02	2.46E+01
		ISCA	**4.04E+02**	**4.09E+02**	**4.10E+02**	**4.23E+02**	**4.64E+00**
	F28	**SCA**	2.06E+03	3.00E+03	3.01E+03	4.34E+03	5.07E+02
		m-SCA	**6.23E+02**	1.01E+03	1.14E+03	2.72E+03	4.32E+02
		ISCA	7.78E+02	**8.64E+02**	**8.66E+02**	**9.88E+02**	**4.60E+01**
	F29	**SCA**	9.94E+06	4.59E+07	5.13E+07	1.07E+08	2.05E+07
		m-SCA	1.49E+04	3.66E+04	9.44E+04	7.59E+05	1.55E+05
		ISCA	**4.41E+03**	**1.84E+04**	**2.32E+04**	**9.11E+04**	**1.55E+04**
	F30	**SCA**	2.52E+05	7.23E+05	7.76E+05	2.00E+06	3.87E+05
		m-SCA	1.75E+04	3.93E+04	4.05E+04	1.07E+05	1.52E+04
		ISCA	**3.42E+03**	**8.81E+03**	**9.58E+03**	**1.77E+04**	**3.23E+03**

Table 5.3. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 10-dimensional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	5.145E-10	+	**F16**	5.145E-10	+
F2	5.145E-10	+	**F17**	5.145E-10	+
F3	5.145E-10	+	**F18**	5.462E-10	+
F4	5.145E-10	+	**F19**	5.145E-10	+
F5	6.528E-10	+	**F20**	5.145E-10	+
F6	5.145E-10	+	**F21**	5.462E-10	+
F7	5.145E-10	+	**F22**	5.145E-10	+
F8	5.145E-10	+	**F23**	5.145E-10	+
F9	5.145E-10	+	**F24**	5.145E-10	+
F10	5.145E-10	+	**F25**	7.433E-08	+
F11	5.145E-10	+	**F26**	5.145E-10	+
F12	5.145E-10	+	**F27**	3.289E-05	+
F13	5.145E-10	+	**F28**	1.486E-09	+
F14	5.145E-10	+	**F29**	1.520E-06	+
F15	5.145E-10	+	**F30**	5.145E-10	+

Table 5.4. Statistical conclusions with p–values obtained by conducting Wilcoxon signed rank test on 30-dimensional IEEE CEC 2014 benchmark problems

Function	p-value	conclusion	Function	p-value	conclusion
F1	5.15E-10	+	**F16**	5.15E-10	+
F2	5.15E-10	+	**F17**	5.15E-10	+
F3	5.15E-10	+	**F18**	5.15E-10	+
F4	5.15E-10	+	**F19**	5.15E-10	+
F5	1.04E-06	+	**F20**	5.15E-10	+
F6	5.15E-10	+	**F21**	5.15E-10	+
F7	5.15E-10	+	**F22**	5.15E-10	+
F8	5.15E-10	+	**F23**	5.15E-10	+
F9	5.15E-10	+	**F24**	5.15E-10	+
F10	5.15E-10	+	**F25**	6.15E-10	+
F11	5.15E-10	+	**F26**	5.15E-10	+
F12	7.35E-10	+	**F27**	5.15E-10	+
F13	5.15E-10	+	**F28**	5.15E-10	+
F14	5.15E-10	+	**F29**	5.15E-10	+
F15	5.15E-10	+	**F30**	5.15E-10	+

Table 5.5. Comparison between m-SCA and ISCA algorithms

Function	Dimension =10		Dimension =30	
	p-value	winner	p-value	winner
F1	1.67E-08	**ISCA**	1.26E-06	**ISCA**
F2	5.15E-10	**ISCA**	1.25E-09	**m-SCA**
F3	5.15E-10	**ISCA**	5.15E-10	**ISCA**
F4	8.80E-02	**same**	2.29E-07	**ISCA**
F5	1.56E-02	**m-SCA**	4.92E-03	**ISCA**
F6	7.83E-08	**ISCA**	8.27E-10	**ISCA**
F7	2.31E-06	**ISCA**	2.08E-08	**m-SCA**
F8	5.15E-10	**ISCA**	5.15E-10	**ISCA**
F9	1.04E-06	**ISCA**	5.15E-10	**ISCA**
F10	5.15E-10	**ISCA**	5.15E-10	**ISCA**
F11	2.07E-07	**ISCA**	5.15E-10	**ISCA**
F12	2.87E-08	**ISCA**	7.34E-03	**ISCA**
F13	2.21E-06	**ISCA**	2.21E-04	**ISCA**
F14	3.66E-02	**ISCA**	5.53E-09	**m-SCA**
F15	1.58E-08	**ISCA**	4.10E-04	**ISCA**
F16	1.27E-08	**ISCA**	5.15E-10	**ISCA**
F17	3.73E-01	**same**	1.07E-03	**ISCA**
F18	1.73E-02	**m-SCA**	5.15E-10	**ISCA**
F19	8.65E-09	**ISCA**	1.49E-09	**ISCA**
F20	2.97E-09	**ISCA**	5.15E-10	**ISCA**
F21	4.01E-03	**ISCA**	2.11E-02	**m-SCA**
F22	3.94E-09	**ISCA**	5.79E-05	**ISCA**
F23	9.87E-10	**m-SCA**	7.35E-10	**ISCA**
F24	4.49E-02	**m-SCA**	5.15E-10	**ISCA**
F25	9.37E-04	**m-SCA**	7.32E-09	**m-SCA**
F26	1.69E-05	**m-SCA**	4.37E-05	**m-SCA**
F27	1.69E-07	**m-SCA**	5.46E-10	**ISCA**
F28	2.29E-07	**m-SCA**	1.20E-06	**ISCA**
F29	8.47E-04	**m-SCA**	5.53E-09	**ISCA**
F30	5.15E-10	**ISCA**	5.15E-10	**ISCA**

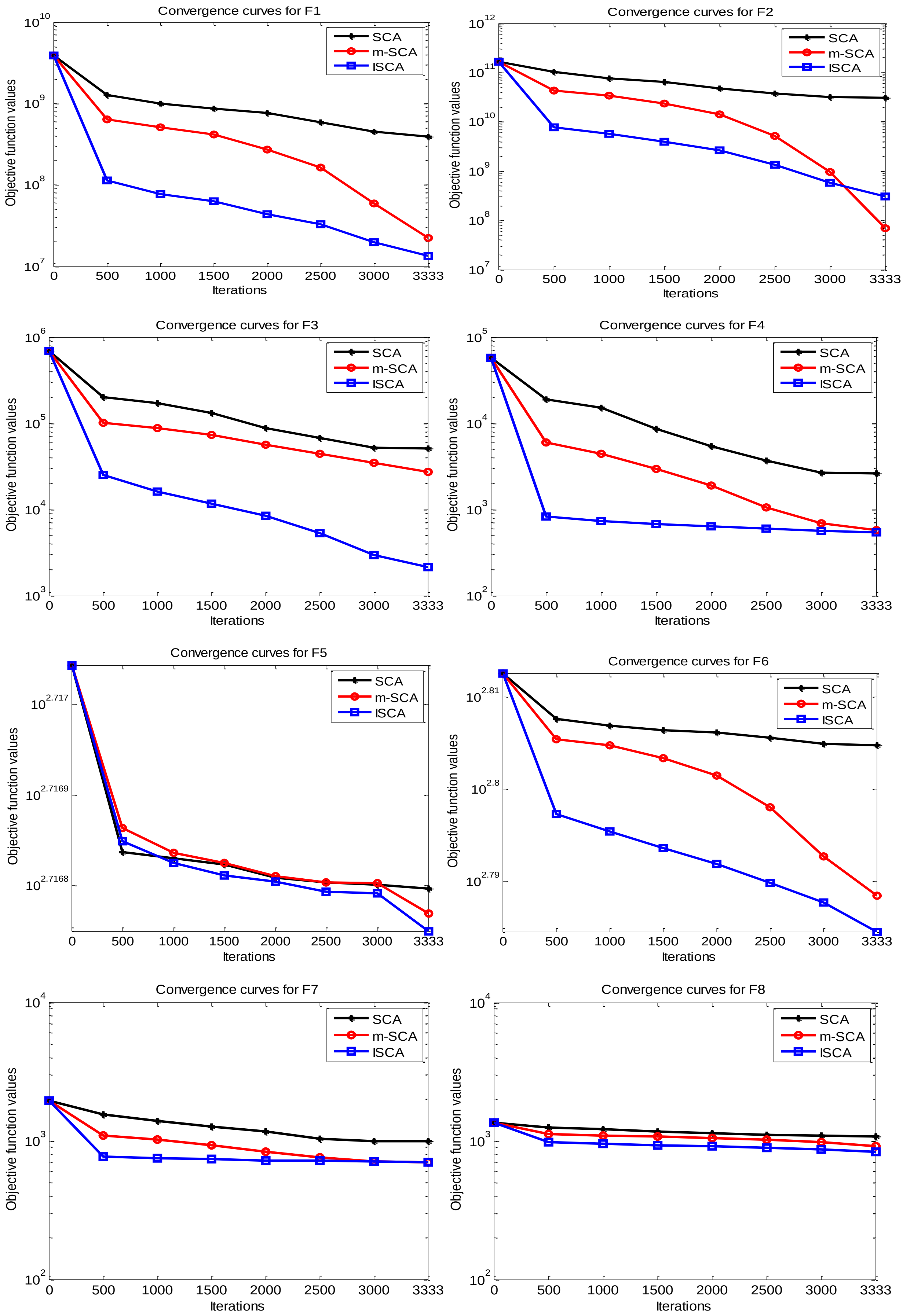


Fig 5.1. Convergence curves for 30-dimensional problems from F1 to F8 corresponding to elite candidate solution of each iteration

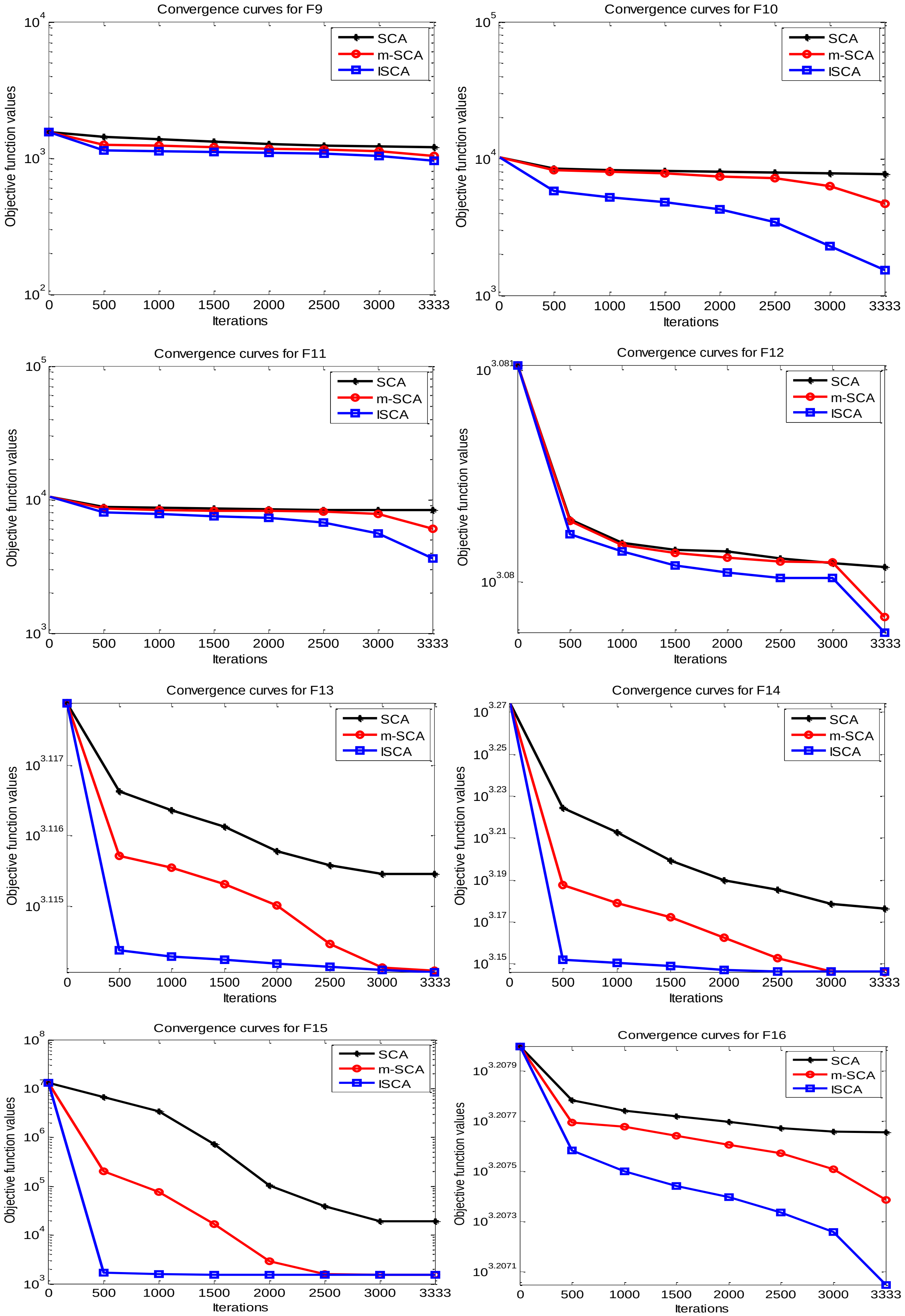


Fig 5.2. Convergence curves for 30-dimensional problems from F9 to F16 corresponding to elite candidate solution of each iteration

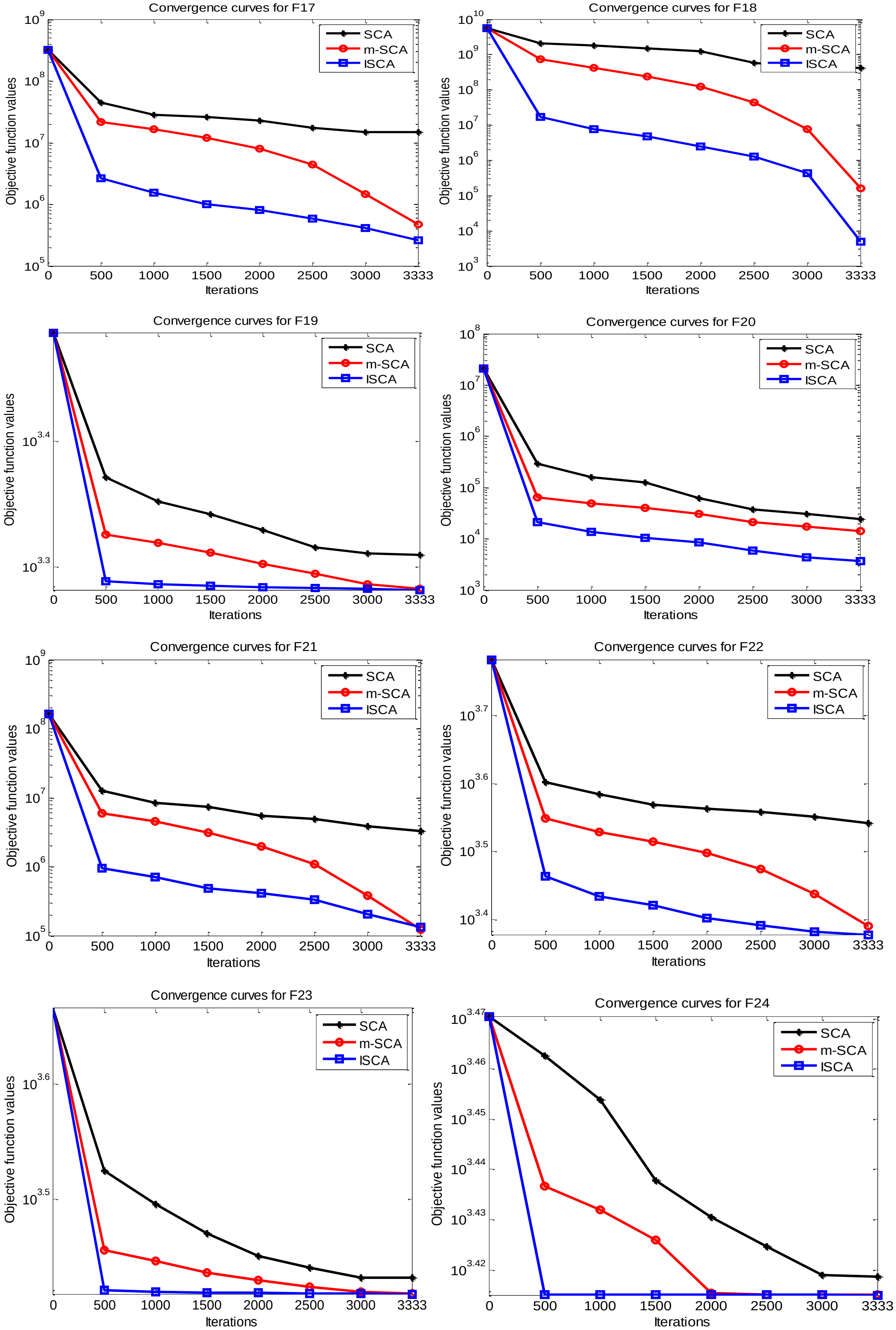


Fig 5.3. Convergence curves for 30-dimensional problems from F17 to F24 corresponding to elite candidate solution of each iteration

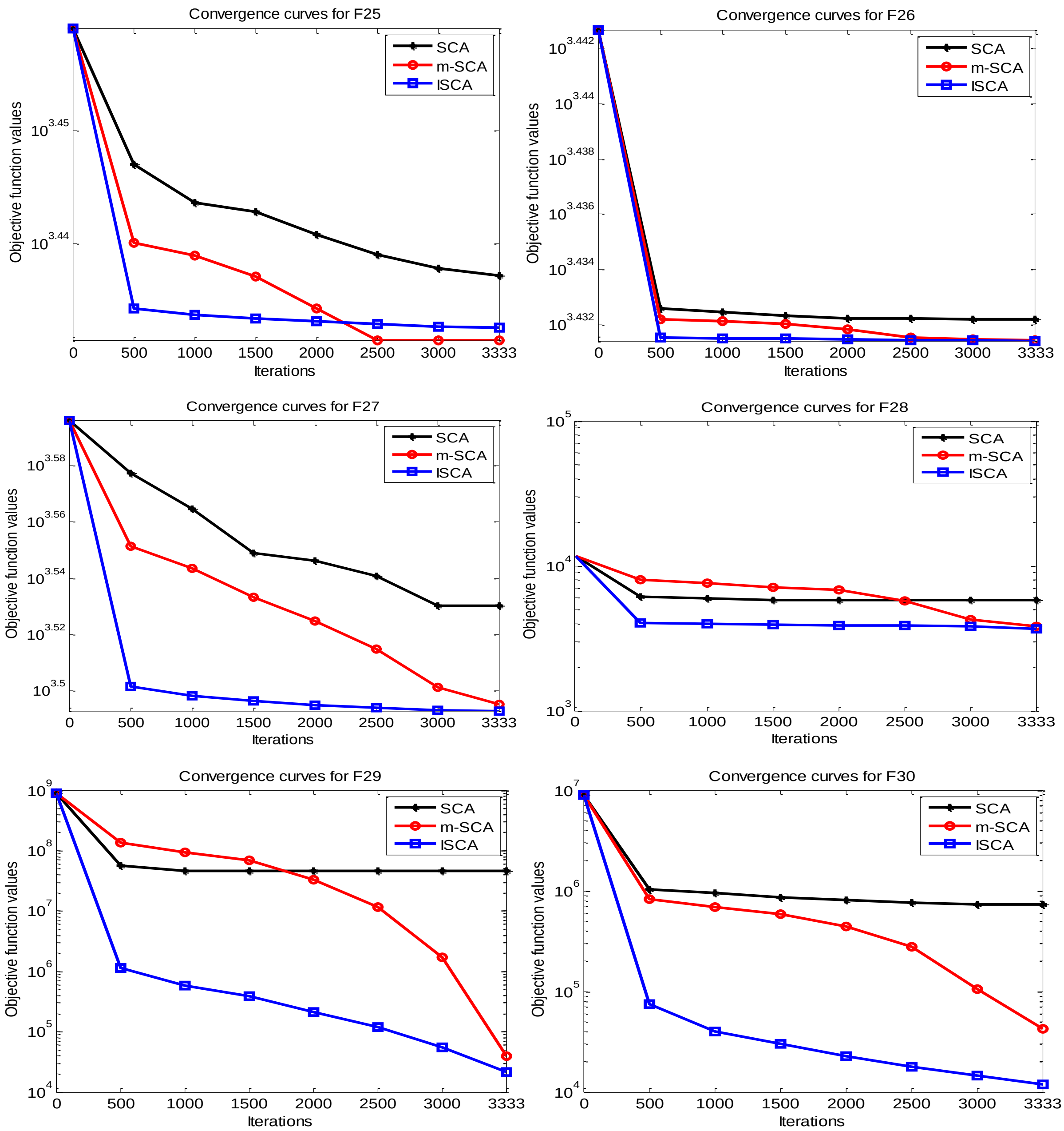


Fig 5.4. Convergence curves for 30-dimensional problems from F25 to F30 corresponding to elite candidate solution of each iteration

Chapter 6

Performance of GWO and SCA Variants on Constrained Optimization Problems

In this chapter, the performance of the classical GWO, classical SCA and their proposed variants which are presented in Chapters 2, 3, 4 and 5 are evaluated on constrained optimization problems.

6.1. Introduction

In the previous chapters of this Thesis, two variants of GWO called RW-GWO and mGWO, and two variants of SCA called m-SCA and ISCA are proposed. These variants have shown their enhanced search efficiency and superior performance as compared to their classical versions on unconstrained optimization problems. In order to analyze the search ability on constrained optimization problems, the present chapter evaluates these variants on constrained benchmark problems given in IEEE CEC 2006. The constrained optimization problems are considered as more difficult problems than unconstrained optimization problems because linear/non-linear constraints reduce the search space of the problem. The reduced feasible search space evaluates the random and guided search ability of any metaheuristic search algorithm.

The organization of the chapter is as follows: Section 6.2 provides the detail and working of the constraint handling technique which is used to tackle the constraints of the problem. In Section 6.3, numerical experimentation, analysis and comparison is shown between classical GWO, classical SCA and their proposed variants. Finally the chapter concludes with Section 6.4.

6.2. Constraint Handling Technique

The constraint handling technique plays a significant role for the constrained optimization problems to handle the constraints. In this chapter, a parameter-free constraint handling approach based on constraint violation [191, 192] is used to select best wolves/candidate solutions from the population.

The partial content of this chapter is published in:

Gupta, S., & Deep, K. (2018). Random walk grey wolf optimizer for constrained engineering optimization problems. Computational Intelligence, 34(4), 1025-1045. Wiley.

For a general optimization problem (1.1) – (1.4), the constraint violation ($viol_X$) corresponding to a solution X can be calculated as follows:

$$viol_X = \sum_{j=1}^{J} G_j(X) + \sum_{k=1}^{K} H_k(X) \quad (6.1)$$

where

$$G_j(X) = \begin{cases} g_j(X) & if\ g_j(X) > 0 \\ 0 & otherwise \end{cases} \quad (6.2)$$

$$H_k(X) = \begin{cases} |h_k(X)| & if\ |h_k(X)| > \in \\ 0 & otherwise \end{cases} \quad (6.3)$$

and $\in$ is predefined tolerance parameter which is fixed as 10^{-4} in the chapter.

In each iteration, the population of wolves/candidate solutions is arranged as follows:

1. The feasible wolves/candidate solutions are placed before the infeasible wolves/candidate solutions.
2. The feasible wolves/candidate solutions are placed in increasing order of their objective function values.
3. The infeasible wolves/candidate solutions are placed in increasing order of their constraint violation values.

From the above sorted list, best wolves/candidate solutions can be selected.

It can be easily analyzed that the constraint handling technique described above is quite easy and common way of picking best wolves/candidate solutions from the population for a constrained optimization problem. Also this technique can be thought of as indirect form of Deb's techniques [191] in which each wolf/candidate solution is compared with remaining wolves/candidate solutions using Deb's feasibility rules, which are as follows:

1. Between two feasible solutions, the solution which has better fitness is selected.
2. Between two infeasible solutions, a solution with less constraint violation is selected, and
3. Between a feasible and an infeasible solution, a feasible solution is selected.

This process provides a new population of wolves/candidate solutions when we arrange them after comparing each wolf/candidate solution with all other remaining wolves/candidate solutions. This new population of wolves/candidate solutions is same as the population obtained by applying the constrained handling technique described above. The main feature in this technique is the simplicity and easy implementation. This constraint handling technique does not require any extra parameters other than the algorithm parameters.

6.3. Numerical Experiments and Analysis of the Results

6.3.1. Benchmark Functions and Parameter Setting

In this section, the performance of classical GWO, RW-GWO, mGWO, classical SCA, m-SCA and ISCA is studied on the basis of constrained benchmark problems given in IEEE CEC 2006 [193]. This benchmark test set consists of 24 test problems with inequality and/or equality constraints. As per the guidelines provided by IEEE CEC 2006, 25 runs of each algorithm are performed corresponding to each test problem. The population size for each problem is fixed as $3 \times D$, where D represent the dimension of the problem.

6.3.2. Analysis of the Results

Based on the benchmark criteria stated in Section 6.3.1, the algorithms – classical GWO, RW-GWO, mGWO, classical SCA, m-SCA, and ISCA are implemented on IEEE CEC 2006 benchmark set and the obtained results are presented in Table 6.1. In this table, various statistical metrics such as minimum, median, average, maximum, and standard deviation of objective function values are recorded corresponding to each test problem.

From the Table 6.1, it can be observed that in the problems g05, g17, g20-g22 all the algorithms are unable to enter in a feasible region of the problem. In the problems, g05, g20-g22, the estimated ratio between feasible region and search space is 0.0000% and in g17, it is 0.0003% which makes the problem more difficult in terms of finding the feasible region of the search space. From now onwards the discussion is based on those problems where the algorithm is able to provide a feasible solution.

From the Table 6.1, it can be observed that the classical GWO is able to provide the 100% feasibility in the 14 problems namely g01-g04, g07, g09-g12, g14, g16, g18, g19, g24. RW-GWO, mGWO and m-SCA provide 100% feasibility in 15 problems namely, g01-g04, g06-g12, g16, g18, g19, g24. The ISCA provides 100% feasibility in 13 problems namely, g01- g04, g07, g09-g12, g16, g18, g19 and g24. The classical SCA provides 100% feasibility in 13 problems namely, g01, g03, g04, g07, g09-g12, g16, g18, g19, g23 and g24. Hence, in terms of feasibility rate the RW-GWO, mGWO and m-SCA are more successful as compared to other algorithms.

In terms of minimum objective function value, the classical GWO is better in g04, RW-GWO is better in g01, g10 and g19, mGWO is better in g06, g07, g09, g13, g16 and g18, m-SCA is better in g15 and ISCA is better in g14 and g23 as compared to other algorithms. In g02, RW-GWO and mGWO provides better minimum value of objective function as compared to all other algorithms and this achieved value of objective function is optimal. Similarly, in g03, classical GWO and RW-GWO provides better minimum value of objective function than the others and this achieved value of objective function is optimal. In g08, g11 and g12, all the algorithms provide same minimum value (equal to the optimal value) of objective function. In g24, all the algorithms except SCA and m-SCA provide same minimum value (equal to the optimal value) of objective function.

In terms of median objective function value, the classical GWO is better than others in g23. In problem g08, the algorithms, classical GWO, mGWO, RW-GWO, m-SCA and ISCA provide same median value (equal to the optimal value) of the objective function. In problem g11, the classical GWO and mGWO provides same median value (equal to the optimal value) of objective function. In problem g12, except for the classical SCA, all the other algorithms provides same median value (equal to the optimal value) of the objective function. The RW-GWO is better than others in problems g01, g03, g04 and g13 in terms of median value of objective function. The mGWO is better than others in problems g02, g06, g07, g09, g10, g16, g19 and g24 in terms of median value of objective function. The m-SCA is better than other in problems g14 and g15. The ISCA is better than other algorithms in problem g18 only, in terms of median value of objective function. In g23, the classical SCA and ISCA are better than other algorithms in providing median value of objective function.

In terms of mean objective function value, the classical GWO is better in g04, g11 and g23, RW-GWO is better in g01, g07 and g18, mGWO is better in g02, g03, g06, g09, g10, g16, g19 and g24, m-SCA is better in g14 and g15 as compared to rest of the competitive algorithms. In g08 and g12, all the algorithms except classical SCA and m-SCA provide same mean value (equal to the optimal value) of objective function. In g13, mGWO and ISCA provide same mean value (equal to the optimal value) of objective function.

In terms of worst value (maximum) of objective function, the classical GWO is better in problems g04, g11 and g23, the RW-GWO is better in g01 only, mGWO is better in g02, g03, g06, g07, g09, g10, g13, g16, g19 and g24. The m-SCA is better than others in problems g14 and g15, ISCA is better than other algorithms in g18 only. In problem g08 and g12, except for the classical SCA all the algorithms provide same maximum value (equal to the optimal value) of the objective function.

In terms of the standard deviation value, the classical GWO is better in g04 and g11, RW-GWO is better in g01 and g07, mGWO is better in the problems, g02, g09, g10, g16, g19 and g24, classical SCA is better in g03 only, m-SCA is better in g14 and g15, ISCA is better in g13 and g18. In the problems g06, the standard deviation value of mGWO and classical SCA is same but better than other algorithms. In problem g08, except for the classical SCA and m-SCA and in problem g12, except for the classical SCA, the standard deviation value is equal for all the algorithms and better than these exceptional algorithms. In g23, the classical versions of GWO and SCA provide better standard deviation than other algorithms.

Hence, from the experimental results and comparison, it can be observed that the mGWO algorithm performs better than the classical GWO and RW-GWO, and ISCA performs better than the classical SCA and m-SCA.

6.3.3. Statistical Analysis

In order to test the statistical validity of the results obtained by classical GWO, RW-GWO, mGWO, classical SCA, m-SCA and ISCA, a non-parametric Wilcoxon rank sum test [186] is applied. The test has been conducted at 5% level of significance and the obtained statistical results are presented in Tables 6.2 to 6.4. The results clearly demonstrate that the mGWO is significantly better than classical GWO and RW-GWO algorithms in most of the test problems. Similarly, the ISCA is significantly better than classical SCA and m-SCA in most of the test problems.

6.4. Concluding Remarks

In the present chapter, the performance of the constrained versions of classical GWO, classical SCA and their proposed variants which are presented in Chapters 2, 3, 4 and 5 is evaluated on constrained problems given in IEEE CEC 2006. In these algorithms, a simple constraint handling technique based on constraint violation is used to handle the constraints of problems. The comparison of results shows that the mGWO algorithm has outperformed classical GWO and RW-GWO algorithms. Similarly, the ISCA algorithm has outperformed classical SCA and m-SCA. Overall comparison between all the algorithms shows that the mGWO algorithm has outperformed all other variants of GWO and SCA and their classical versions.

Table 6.1. Comparison of results between classical GWO, classical SCA and their proposed variants on IEEE CEC 2006 constrained problems

Function	Algorithm	FR	Optima	Minimum	Median	Maximum	Mean	STD
g01	**GWO**	100	-15	-14.9985	-11.9767	-11.2994	-7.0664	2.1313
	RW-GWO	100		**-14.9990**	**-14.9983**	**-14.9983**	**-14.9975**	**0.0004**
	mGWO	100		-14.9872	-14.9796	-14.9780	-14.9627	0.0066
	SCA	100		-8.8773	-6.0000	-6.1097	-4.0000	1.1824
	m-SCA	100		-13.3405	-12.4317	-11.8868	-7.8646	1.5036
	ISCA	100		-14.9967	-11.9990	-11.9588	-9.0000	1.4844
g02	**GWO**	100	−0.8036	-0.8034	-0.7519	-0.7407	-0.6208	0.0488
	RW-GWO	100		**-0.8036**	-0.7856	-0.7763	-0.7277	0.0223
	mGWO	100		**-0.8036**	**-0.8035**	**-0.8010**	**-0.7924**	**0.0044**
	SCA	84		-0.6103	-0.5096	-0.5076	-0.4253	0.0444
	m-SCA	100		-0.7368	-0.5793	-0.5911	-0.4739	0.0727
	ISCA	100		-0.8034	-0.7979	-0.7968	-0.7719	0.0078
g03	**GWO**	100	−1.0005	**-1.0005**	0.0000	-0.3551	0.0000	0.4827
	RW-GWO	100		**-1.0005**	**-1.0004**	-0.6784	0.0000	0.4538
	mGWO	100		-1.0004	-0.9986	**-0.9942**	**-0.9371**	0.0144
	SCA	100		0.0000	0.0000	0.0000	0.0000	**0.0000**
	m-SCA	100		-0.6556	-0.4023	-0.3885	-0.1197	0.1493
	ISCA	100		-0.7288	-0.0231	-0.0855	0.0000	0.1555
g04	**GWO**	**100**	−30665.5387	**-30665.5183**	-30665.3806	**-30665.3902**	**-30665.2531**	**0.0672**
	RW-GWO	**100**		-30665.4839	**-30665.3816**	-30665.3706	-30665.2289	0.0751
	mGWO	**100**		-30665.0443	-30664.7606	-30664.6129	-30663.6828	0.3633
	SCA	**100**		-30651.2013	-30622.2407	-30620.7484	-30539.6262	23.7919
	m-SCA	**100**		-30640.0754	-30620.0943	-30620.4884	-30595.1192	9.5673
	ISCA	**100**		-30665.4319	-30665.0172	-30665.0400	-30664.3676	0.2875
g05	**GWO**	**0**	5126.4967	NA	NA	NA	NA	NA
	RW-GWO	**0**		NA	NA	NA	NA	NA
	mGWO	**0**		NA	NA	NA	NA	NA
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**0**		NA	NA	NA	NA	NA

Function	Algorithm	FR	Optima	Minimum	Median	Maximum	Mean	STD
g06	**GWO**	**88**	-6961.8139	-6961.0712	-6950.8032	-6939.2355	-6874.8493	26.7092
	RW-GWO	**100**		-6960.6666	-6946.1456	-6940.6530	-6877.2450	22.3092
	mGWO	**100**		**-6961.8139**	**-6961.8139**	**-6961.8139**	**-6961.8139**	**0.0000**
	SCA	**4**		-6933.0557	-6933.0557	-6933.0557	-6933.0557	**0.0000**
	m-SCA	**100**		-6957.0843	-6865.2033	-6766.3928	-6245.1956	221.5029
	ISCA	**60**		-6958.7833	-6936.6853	-6929.2806	-6838.7383	31.8411
g07	**GWO**	**100**	24.3062	24.7910	29.6234	37.8751	136.2662	29.4923
	RW-GWO	**100**		24.4283	24.6792	24.7998	**25.2596**	**0.2948**
	mGWO	**100**		**24.3822**	**24.5326**	**24.7117**	28.9498	0.8893
	SCA	**100**		49.2121	56.4263	57.8688	71.2271	6.5811
	m-SCA	**100**		25.8248	27.3936	27.4437	29.6968	1.0367
	ISCA	**100**		24.5267	24.9213	25.8465	29.1879	1.8025
g08	**GWO**	**92**	−0.0958	**-0.0958**	**-0.0958**	**-0.0958**	**-0.0958**	**0.0000**
	RW-GWO	**100**		**-0.0958**	**-0.0958**	**-0.0958**	**-0.0958**	**0.0000**
	mGWO	**100**		**-0.0958**	**-0.0958**	**-0.0958**	**-0.0958**	**0.0000**
	SCA	**72**		**-0.0958**	-0.0956	-0.0954	-0.0945	0.0005
	m-SCA	**100**		**-0.0958**	**-0.0958**	**-0.0958**	-0.0952	0.0001
	ISCA	**96**		**-0.0958**	**-0.0958**	**-0.0958**	**-0.0958**	**0.0000**
g09	**GWO**	**100**	680.6301	680.7987	683.2719	685.3362	711.5630	6.5790
	RW-GWO	**100**		680.7119	681.0041	681.1276	684.0611	0.6376
	mGWO	**100**		**680.6355**	**680.6432**	**680.6506**	**680.7161**	**0.0202**
	SCA	**100**		686.5866	689.3029	689.3715	693.5485	1.9691
	m-SCA	**100**		680.7154	680.8972	681.0712	682.8519	0.5007
	ISCA	**100**		680.6989	680.9866	680.9683	681.1777	0.1270
g10	**GWO**	**100**	7049.2480	7088.7510	7561.6617	7568.9886	8300.0585	373.9541
	RW-GWO	**100**		**7057.5545**	7427.2296	7360.1055	7521.8361	130.4502
	mGWO	**100**		7120.5616	**7256.4523**	**7256.7773**	**7453.3711**	**81.1893**
	SCA	**100**		8687.6159	9561.1045	9505.0424	10395.6235	541.2416
	m-SCA	**100**		7220.8305	7532.5190	7631.7571	9031.4537	395.1446
	ISCA	**100**		7353.6872	7698.4176	7698.9003	8402.9482	221.9554

Function	Algorithm	FR	Optima	Minimum	Median	Maximum	Mean	STD
g11	**GWO**	**100**	0.7499	**0.7499**	**0.7499**	**0.7502**	**0.7570**	**0.0014**
	RW-GWO	**100**		**0.7499**	0.7502	0.7861	0.9999	0.0704
	mGWO	**100**		**0.7499**	**0.7499**	0.7532	0.7852	0.0089
	SCA	**100**		**0.7499**	0.7507	0.7566	0.8046	0.0134
	m-SCA	**100**		**0.7499**	0.7500	0.7517	0.7812	0.0063
	ISCA	**100**		**0.7499**	0.7511	0.7870	0.9999	0.0671
g12	**GWO**	**100**	-1.0000	**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**0.0000**
	RW-GWO	**100**		**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**0.0000**
	mGWO	**100**		**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**0.0000**
	SCA	**100**		**-1.0000**	-0.9999	-0.9996	-0.9937	0.0012
	m-SCA	**100**		**-1.0000**	**-1.0000**	**-1.0000**	-0.9999	**0.0000**
	ISCA	**100**		**-1.0000**	**-1.0000**	**-1.0000**	**-1.0000**	**0.0000**
g13	**GWO**	**36**	0.0539	0.9796	0.9999	1.0028	1.0538	0.0203
	RW-GWO	**40**		0.8535	**0.9993**	0.9676	1.0003	0.0544
	mGWO	**52**		**0.4745**	0.9999	**0.9007**	**1.0000**	0.1870
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**68**		0.9998	1.0000	1.0000	**1.0000**	**0.0000**
g14	**GWO**	**100**	−47.7649	-46.5398	-41.5971	-41.8303	-37.4389	2.4773
	RW-GWO	**96**		-44.9855	-40.6523	-40.8448	-36.1438	2.5789
	mGWO	**72**		-46.1973	-42.3700	-42.5725	-40.2289	1.8280
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**20**		-44.9596	**-43.1657**	**-43.2747**	**-42.0671**	**1.2464**
	ISCA	**52**		**-47.4498**	-40.2601	-40.9230	-35.4323	3.2081
g15	**GWO**	**56**	961.7150	961.7168	968.9334	968.2013	972.3099	4.1736
	RW-GWO	**52**		961.7390	964.4088	966.0403	972.2926	4.0576
	mGWO	**56**		961.7187	962.1567	963.1587	966.4812	1.7412
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**20**		**961.7155**	**961.7346**	**961.7560**	**961.8616**	**0.0608**
	ISCA	**88**		961.7181	962.2054	964.2342	972.3151	3.8551

Function	Algorithm	FR	Optima	Minimum	Median	Maximum	Mean	STD
g16	**GWO**	**100**	−1.9051	-1.9037	-1.8376	-1.8436	-1.7443	0.0489
	RW-GWO	**100**		-1.9017	-1.8833	-1.8781	-1.8356	0.0210
	mGWO	**100**		**-1.9051**	**-1.9044**	**-1.9044**	**-1.9024**	**0.0007**
	SCA	**100**		-1.7776	-1.5277	-1.5102	-1.2027	0.1609
	m-SCA	**100**		-1.8913	-1.8010	-1.8152	-1.7315	0.0552
	ISCA	**100**		-1.9038	-1.8694	-1.8672	-1.8200	0.0309
g17	**GWO**	**0**	8853.5397	NA	NA	NA	NA	NA
	RW-GWO	**0**		NA	NA	NA	NA	NA
	mGWO	**0**		NA	NA	NA	NA	NA
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**0**		NA	NA	NA	NA	NA
g18	**GWO**	**100**	−0.8660	-0.8659	-0.8570	-0.7981	-0.4990	0.1126
	RW-GWO	**100**		-0.8659	-0.8605	-0.8436	**-0.6717**	0.0524
	mGWO	**100**		**-0.8660**	-0.8494	-0.7901	-0.6631	0.0922
	SCA	**100**		-0.7695	-0.7016	-0.6719	-0.4594	0.0875
	m-SCA	**100**		-0.8644	-0.8577	-0.7879	-0.5324	0.1028
	ISCA	**100**		-0.8658	**-0.8626**	**-0.8496**	-0.6609	**0.040**4
g19	**GWO**	**100**	32.6556	33.4456	35.5593	38.5591	66.3511	8.0215
	RW-GWO	**100**		**32.7399**	34.2740	36.2645	50.6187	4.9496
	mGWO	**100**		33.4757	**34.0840**	**34.1651**	**35.6638**	**0.4988**
	SCA	**100**		45.3745	89.5503	151.5567	880.0015	178.8423
	m-SCA	**100**		47.6534	65.5455	68.6097	124.6039	16.7807
	ISCA	**100**		32.9363	34.7730	37.0597	50.8037	5.5059
g20	**GWO**	**0**	0.2050	NA	NA	NA	NA	NA
	RW-GWO	**0**		NA	NA	NA	NA	NA
	mGWO	**0**		NA	NA	NA	NA	NA
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**0**		NA	NA	NA	NA	NA

Function	Algorithm	FR	Optima	Minimum	Median	Maximum	Mean	STD
g21	**GWO**	**0**	193.7245	NA	NA	NA	NA	NA
	RW-GWO	**0**		NA	NA	NA	NA	NA
	mGWO	**0**		NA	NA	NA	NA	NA
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**0**		NA	NA	NA	NA	NA
g22	**GWO**	**0**	236.4310	NA	NA	NA	NA	NA
	RW-GWO	**0**		NA	NA	NA	NA	NA
	mGWO	**0**		NA	NA	NA	NA	NA
	SCA	**0**		NA	NA	NA	NA	NA
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**0**		NA	NA	NA	NA	NA
g23	**GWO**	**4**	-400.0551	-0.0018	**-0.0018**	**-0.0018**	**-0.0018**	**0.0000**
	RW-GWO	**0**		NA	NA	NA	NA	NA
	mGWO	**0**		NA	NA	NA	NA	NA
	SCA	**100**		0.0000	**0.0000**	0.0000	0.0000	**0.0000**
	m-SCA	**0**		NA	NA	NA	NA	NA
	ISCA	**88**		**-0.0030**	**0.0000**	163.6358	900.0000	355.2942
g24	**GWO**	**100**	-5.5080	**-5.5080**	-5.5079	-5.3676	-2.0000	0.7016
	RW-GWO	**100**		**-5.5080**	-5.5016	-5.5000	-5.4854	0.0063
	mGWO	**100**		**-5.5080**	**-5.5080**	**-5.5080**	**-5.5080**	**0.0000**
	SCA	**100**		-5.4992	-5.4689	-5.3599	-3.0000	0.4932
	m-SCA	**100**		-5.5079	-5.4968	-5.4947	-5.4658	0.0102
	ISCA	**100**		**-5.5080**	-5.5028	-5.4992	-5.4724	0.0099

Table 6.2. Results of statistical test between classical GWO and its proposed variants on IEEE CEC 2006 benchmark problems

Function	GWO vs RW-GWO		GWO vs mGWO	
	p-value	conclusion	p-value	conclusion
g01	1.17E-08	+	1.14E-05	+
g02	8.32E-03	+	3.99E-08	+
g03	1.33E-02	+	3.24E-03	+
g04	4.97E-01	=	7.57E-09	-
g05	NA	NA	NA	NA
g06	5.16E-01	=	2.17E-10	+
g07	1.17E-08	+	3.26E-09	+
g08	3.35E-02	+	1.64E-09	+
g09	1.67E-04	+	1.42E-09	+
g10	2.32E-02	+	1.79E-03	+
g11	1.19E-03	-	4.26E-01	=
g12	7.89E-04	-	9.73E-11	+
g13	4.00E-01	=	5.93E-01	=
g14	1.77E-01	=	3.07E-01	=
g15	1.67E-01	=	2.62E-03	+
g16	4.57E-02	+	2.90E-09	+
g17	NA	NA	NA	NA
g18	5.47E-02	+	9.38E-01	=
g19	3.44E-02	+	1.12E-03	+
g20	NA	NA	NA	NA
g21	NA	NA	NA	NA
g22	NA	NA	NA	NA
g23	NA	NA	NA	NA
g24	2.66E-06	-	4.77E-10	+

Table 6.3. Results of statistical test between classical SCA and its proposed variants on IEEE CEC 2006 benchmark problems

Function	SCA vs m-SCA		SCA vs ISCA	
	p-value	**conclusion**	**p-value**	**conclusion**
g01	2.64E-09	+	1.14E-09	+
g02	1.65E-05	+	1.41E-09	+
g03	9.73E-11	+	1.46E-08	+
g04	4.15E-01	=	1.42E-09	+
g05	NA	NA	NA	NA
g06	5.05E-01	=	8.75E-01	=
g07	1.42E-09	+	1.42E-09	+
g08	2.56E-04	+	4.32E-08	+
g09	1.42E-09	+	1.42E-09	+
g10	2.90E-09	+	1.42E-09	+
g11	1.17E-02	+	8.46E-01	=
g12	3.53E-06	+	1.42E-09	+
g13	NA	NA	NA	NA
g14	NA	NA	NA	NA
g15	NA	NA	NA	NA
g16	4.64E-09	+	1.42E-09	+
g17	NA	NA	NA	NA
g18	1.67E-04	+	1.31E-08	+
g19	3.61E-02	+	9.29E-09	+
g20	NA	NA	NA	NA
g21	NA	NA	NA	NA
g22	NA	NA	NA	NA
g23	NA	NA	1.68E-01	=
g24	9.73E-11	+	3.02E-07	+

Table 6.4. Results of statistical test between proposed variants of GWO and proposed variants of SCA on IEEE CEC 2006 benchmark problems

Function	RW-GWO vs mGWO		m-SCA vs ISCA	
	p-value	conclusion	p-value	conclusion
g01	7.57E-09	RW-GWO	1.16E-01	same
g02	2.12E-06	mGWO	1.42E-09	ISCA
g03	1.07E-01	same	1.61E-07	m-SCA
g04	7.57E-09	RW-GWO	1.42E-09	ISCA
g05	NA	NA	NA	NA
g06	9.73E-11	mGWO	1.40E-02	m-SCA
g07	3.61E-03	mGWO	4.79E-04	ISCA
g08	7.73E-10	mGWO	1.36E-05	ISCA
g09	1.60E-09	mGWO	1.35E-01	same
g10	1.79E-03	mGWO	2.70E-02	m-SCA
g11	1.30E-02	mGWO	1.57E-01	same
g12	9.73E-11	mGWO	6.42E-05	ISCA
g13	9.26E-01	same	NA	NA
g14	3.17E-02	mGWO	NA	NA
g15	4.94E-02	mGWO	NA	NA
g16	1.42E-09	mGWO	2.11E-04	ISCA
g17	NA	NA	NA	NA
g18	1.40E-01	same	5.21E-03	ISCA
g19	9.69E-01	same	1.60E-09	ISCA
g20	NA	NA	NA	NA
g21	NA	NA	NA	NA
g22	NA	NA	NA	NA
g23	NA	NA	NA	NA
g24	4.77E-10	mGWO	5.72E-02	same

Chapter 7

Multilevel Thresholding Using Proposed Variants of GWO and SCA

In this chapter, the classical versions of GWO and SCA and their proposed variants which are presented is Chapters 2, 3, 4 and 5 are used to solve an unconstrained, nonlinear and discrete optimization problem arising in the field of image processing. The idea is to observe the effectiveness of solving real life problem using the proposed algorithms in the Thesis.

7.1. Introduction

In the field of image processing, the image segmentation is the process of partitioning an image into multiple segments which are called as super-pixels. In other words, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics. The aim of the segmentation is to change or/and simplify the representation of an image into some meaning form which is easier to analyze. Image segmentation is widely used in face detection, brake light detection, fingerprint recognition, traffic control system, Machine vision, surgery planning, video surveillance and in many other fields. The simplest method of image segmentation is called the thresholding method. This method is based on determining the clip-level(s) or threshold value(s). Several popular methods which are used in industry for thresholding includes maximum entropy method, hybrid thresholding, Otsu's between class variance method and k-means clustering.

The present chapter analyzes the performances of classical GWO, classical SCA, and their proposed variants called RW-GWO, mGWO, m-SCA, ISCA which are proposed in Chapters 2, 3, 4, and 5 respectively in finding the optimal thresholds for grey images. The set of nine benchmark images is taken to compare the performance of the algorithms. The optimal thresholds in these algorithms are determined by maximizing the fitness value provided by the Otsu method. The various statistical metrics and the image quality metric are used for the fair comparison among the algorithms.

The partial content of this Chapter is published in

Gupta, S., & Deep, K. (2019). Improved sine cosine algorithm with crossover scheme for global optimization. Knowledge-Based Systems, 165, 374-406. Elsevier.

The organization of the chapter is as follows: Section 7.2 provides a detailed description of multilevel thresholding problem. Section 7.3 provides a numerical experimentation, analysis and comparison between classical variants of GWO and SCA, and their proposed variants, which are presented in Chapters 2, 3, 4 and 5. Finally the chapter concludes with Section 7.4.

7.2. Multilevel Thresholding

Multilevel thresholding is a popular method for image segmentation. The bi-level thresholding partitions an image into two classes namely the object and its background. But, some of the images consists of homogeneous regions such as grey level and colour which imply the possibilities of effective segmentation. The bi-level thresholding is not very effective in these cases. Therefore, to segment an image into multiple classes, the multilevel thresholding method can be used. In multilevel thresholding, the determination of optimal thresholds is very crucial for the proper segmentation of an image. Multilevel thresholding is a widely used basic operation in image processing and in the literature, it is used as an efficient segmentation algorithm.

To find the optimal thresholds, two approaches are available in the literature – parametric and non-parametric approaches. In parametric approaches, first, a statistical model is assumed to fit the grey level distribution of an image. Secondly, the set of parameters that controls the fitness of the model are determined using the histogram of image. In [194], a parametric global thresholding method is developed which finds the optimal thresholds by estimating the parameters based on expectation-maximization function under the assumption that the object and its background classes follow the generalized gaussian distribution. In non-parametric approaches, the thresholds are determined by optimizing the objective function such as maximizing between class variance [195], maximizing entropy [196, 197]. In the literature [198], clustering techniques are also applied to solve the image segmentation problem. The segmentation is usually achieved using a histogram of images. The histogram is a distribution of grey level pixels in the image. The finding of optimal thresholds may not be straightforward always as it depends on the shape of histogram. Sometimes the histogram of image contains large number of wide valleys and peaks and therefore, the threshold allocation is difficult in such cases. The nature inspired algorithms are meaningful in these cases as they work independently from the nature of the problem and are capable to locate the deep valleys and peaks. In some past recent years, the nature inspired optimization algorithms [199-203] have attracted a lot of attention in the field of multilevel thresholding to segment images into multiple segments.

7.2.1. Problem Description

In this section, the multilevel thresholding problem is presented using Otsu method. Image segmentation is process of splitting an image into multiple segments with an aim of simplify the representation of objects within an image. Usually, the segmentation is achieved by using a histogram of the image. But in some cases, due to the complex structure of histogram of an image, thresholding is not straightforward. The deep valleys and peaks in the histogram increases the difficulty of the process of thresholding. The Otsu method helps in such situation.

7.2.2. Otsu Method [195]

Otsu method is an unsupervised and non-parametric thresholding method. In this method, the optimal thresholds are determined by optimizing the class variance between segmented classes. For an image which is represented in L grey levels $0, 1, 2, \dots, L-1$, the image histogram $H = \{f_0, f_1, f_2, \dots, f_{L-1}\}$ can be constructed. Here, f_i represent the frequency of occurrence of i^{th} grey level in the image. Let N be the total number of pixels in an image. The occurrence probability of i^{th} grey level is given by

$$p_i = \frac{f_i}{N} \tag{7.1}$$

Let us suppose that there are k thresholds namely $t_1, t_2, \dots, t_k$ are to be determined. Obviously these thresholds will divide the image into $k+1$ classes say $c_0, c_1, c_2, \dots, c_k$. The objective function in the Otsu method which is to be maximized is as follows:

$$F(t_1, t_2, \dots t_k) = v_0^2 + v_1^2 + \cdots v_k^2 \tag{7.2}$$

where,

$$v_0^2 = w_0(u_0 - u_T)^2, \qquad w_0 = \sum_{i=0}^{t_1-1} p_i, \qquad u_0 = \sum_{i=0}^{j=t_1-1} \frac{i\, p_i}{w_0} \tag{7.3}$$

$$v_j^2 = w_j\left(u_j - u_T\right)^2, \qquad w_j = \sum_{i=t_j}^{t_{j+1}-1} p_i, \qquad u_j = \sum_{i=t_j}^{t_{j+1}-1} \frac{i\, p_i}{w_i},\ j = 1,2,\dots,k-1 \tag{7.4}$$

$$v_k^2 = w_k(u_k - u_T)^2, \qquad w_k = \sum_{i=t_k}^{L-1} p_i, \qquad u_k = \sum_{i=t_k}^{L-1} \frac{i\, p_i}{w_i} \tag{7.5}$$

$v_0^2, v_1^2, \dots, v_k^2$ are the variances; $w_0, w_1, \dots, w_k$ are the class probabilities; $u_0, u_1, \dots, u_k$ are the mean levels of the segmented classes $c_0, c_1, c_2, \dots, c_k$. u_T is the mean level for the image which can be calculated as: $u_T = \sum_{i=0}^{k} w_i u_i$ and $\sum_{i=0}^{k} w_i = 1$.

Since the Otsu method provides an objective function which is to be maximized. This objective function can be transformed into minimization type as follows:

$$\hat{F}(t_1, t_2, \ldots t_k) = \frac{1}{1 + F(t_1, t_2, \ldots t_k)} \tag{7.6}$$

7.3. Experimental Results

This section presents an experimental analysis of the proposed algorithms to solve the segmentation problem. The nine benchmark test images are taken from USC-SIPI and BSD 500 image database for the experimentation. These benchmark images are presented in Fig 7.1. 30 runs of each algorithms are performed with 12 population size and 100 iterations. This parameter setting is adopted from the literature. To compare the algorithms, various statistical metrics such as mean, median and best fitness of the objective function defined by Otsu method is presented in the chapter. To confirm the quality of the segmented image, peak signal-to-noise ratio (PSNR) is calculated which is given by

$$PSNR = 10 \log_{10} \frac{255^2}{MSE} \tag{7.7}$$

where, $MSE = \frac{1}{MN}\left[\sum_{i=1}^{M} \sum_{j=1}^{N} \left(O_{i,j} - S_{i,j}\right)^2\right]$

where O and S are the original and segmented images respectively. The obtained results on the test images are presented in Tables 7.1 to 7.3. The best, mean and median value of the objective function is recorded in these tables for higher number of thresholds 3, 4, 5 and 6. For the less number of thresholds, all the variants of GWO and SCA provide the same results.

From the tables, it can be seen that the proposed variant called RW-GWO and mGWO outperform other algorithms in providing the better objective fitness in most of the test images. It can be seen from the tables that in some of the test images, the performance of these algorithms is similar to ISCA algorithm. The ISCA algorithms provides better objective fitness in terms of median as well as average of objective function than classical SCA and m-SCA. Overall, in most of the test images the algorithm RW-GWO and mGWO algorithms provides significantly better results than other proposed variants and classical versions of GWO and SCA. By analyzing the median and mean of objective fitness, the algorithm RW-GWO can be considered as a reliable optimizer as compared to all other algorithms. As an example, the optimal thresholds and the segmented images obtained by the RW-GWO algorithm are presented in Figs 7.2 to 7.5. The quality of the segmented images is compared with the metric PSNR. The mean and median value of PSNR is shown in Tables 7.4

and 7.5 respectively. The higher value of PSNR indicates the better quality of the segmented image. From these tables, it can be analyzed that the variants RW-GWO and mGWO provide better quality of segmented images as compared to the classical GWO, classical SCA, m-SCA and ISCA.

Fig 7.1. Benchmark test images for image segmentation

7.4. Concluding Remarks

In the present chapter, the classical GWO, classical SCA, and their proposed variants which are presented in Chapters 2, 3, 4, 5 called RW-GWO, mGWO, m-SCA, and ISCA respectively are employed to determine the optimal thresholds for grey images. The experimental work has been conducted on nine standard benchmark grey images. The analysis of the results is conducted through various statistical measures such as best, mean, median of objective fitness and the image quality metric PSNR. From the experimental results, it can be observed that the proposed variants of GWO called mGWO and RW-GWO both perform better than classical GWO. Similarly, the ISCA outperforms classical SCA and m-SCA. By analyzing the median and mean value of objective fitness, it can be concluded that the RW-GWO algorithm is more reliable optimizer as compared to all other algorithms. Overall, the proposed RW-GWO and mGWO algorithms outperform other algorithms and therefore, these algorithm can be recommended for thresholding of grey images as compared to other proposed variants and classical versions of GWO and SCA.

Table 7.1. Mean value of the Otsu's objective function using classical GWO, classical SCA and their proposed variants

Test image	Thresholds	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
Cameraman	**3**	3727.37	**3727.40**	**3727.40**	3723.52	3726.32	3727.36
	4	3782.08	**3782.34**	3782.32	3771.81	3779.47	3782.08
	5	3812.40	3813.49	**3813.50**	3798.04	3809.30	3813.14
	6	3830.00	**3833.46**	3832.40	3816.65	3827.59	3831.73
Bridge	**3**	2722.34	2722.34	**2722.36**	2715.23	2721.74	2722.34
	4	2820.06	2822.41	**2822.50**	2809.07	2820.48	2822.43
	5	2869.73	2873.28	**2873.95**	2853.91	2868.65	2873.42
	6	2900.73	**2905.02**	**2905.02**	2882.24	2898.29	2904.75
Couple	**3**	1380.31	1383.15	**1383.16**	1372.55	1382.63	1383.11
	4	1448.83	1448.86	**1449.74**	1438.74	1447.34	1449.62
	5	1491.64	1496.05	**1497.27**	1479.19	1492.83	1496.39
	6	1521.46	1523.14	**1523.93**	1506.51	1518.92	1522.05
Peppers	**3**	2703.54	**2703.57**	2703.56	2695.27	2701.75	2703.52
	4	2766.23	**2766.34**	2766.24	2754.77	2761.24	2766.04
	5	2808.85	**2810.56**	2810.34	2787.94	2797.77	2809.59
	6	2829.19	**2833.11**	2831.22	2810.09	2821.36	2830.99
Airplane	**3**	975.85	**975.88**	975.87	973.04	974.26	975.65
	4	1010.93	1018.68	**1020.99**	1008.57	1014.91	1020.29
	5	1037.24	**1038.46**	1038.36	1029.79	1027.41	1037.33
	6	1052.91	**1054.94**	1054.04	1039.98	1046.73	1053.26
Male	**3**	3126.83	**3126.86**	3126.85	3120.63	3126.25	3126.82
	4	3208.63	3208.75	**3208.76**	3195.81	3206.51	3208.61
	5	3253.94	**3254.50**	3254.45	3237.64	3250.18	3254.22
	6	3280.04	**3283.61**	3283.35	3260.82	3278.75	3282.87
Lena	**3**	2128.28	**2128.30**	2128.27	2119.24	2126.52	2128.20
	4	2191.72	**2191.82**	2191.67	2176.40	2185.67	2191.40
	5	2216.72	**2217.12**	2216.41	2201.05	2204.08	2215.42
	6	2234.86	**2238.02**	2236.57	2216.79	2222.83	2233.42
Bridge2	**3**	3567.57	**3567.59**	3567.58	3560.25	3566.48	3567.56
	4	3660.11	**3660.28**	**3660.28**	3648.91	3658.81	3660.16
	5	3709.31	**3710.08**	**3710.08**	3688.10	3706.49	3709.78
	6	3732.60	**3735.82**	3735.41	3715.79	3729.90	3734.48
Lady	**3**	2211.47	**2211.49**	**2211.49**	2205.08	2210.65	2211.47
	4	2263.38	**2264.26**	**2264.26**	2251.84	2261.67	2264.08
	5	2292.77	2295.12	**2295.16**	2278.71	2291.35	2294.71
	6	2313.16	2315.51	**2315.62**	2296.67	2310.77	2314.85

Table 7.2. Median value of the Otsu's objective function using classical GWO, classical SCA and their proposed variants

Test images	Thresholds	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
Cameraman	**3**	3727.40	**3727.41**	3727.40	3723.78	3726.69	3727.40
	4	3782.27	**3782.36**	3782.35	3772.97	3779.42	3782.23
	5	3813.32	**3813.62**	3813.56	3797.34	3809.32	3813.21
	6	3832.42	**3833.57**	3832.93	3814.59	3827.90	3832.43
Bridge	**3**	**2722.36**	**2722.36**	**2722.36**	2716.08	2722.17	**2722.36**
	4	2822.07	**2822.71**	**2822.71**	2808.79	2821.24	**2822.71**
	5	2873.32	**2874.24**	2874.19	2855.78	2869.01	2873.43
	6	2904.47	2905.69	**2906.01**	2880.13	2898.95	2905.36
Couple	**3**	**1383.16**	**1383.16**	**1383.16**	1373.54	1382.76	1383.11
	4	1449.73	**1449.76**	1449.75	1440.70	1447.64	1449.71
	5	1496.45	1497.21	**1497.39**	1477.67	1493.23	1497.10
	6	1519.54	1524.52	**1525.10**	1506.93	1518.91	1522.00
Peppers	**3**	2703.56	**2703.57**	**2703.57**	2698.65	2702.65	2703.56
	4	2766.39	**2766.40**	2766.28	2756.30	2762.16	2766.23
	5	2810.53	**2810.69**	2810.57	2787.91	2802.43	2809.91
	6	2832.18	**2833.52**	2832.40	2809.93	2822.64	2832.16
Airplane	**3**	975.86	**975.89**	975.89	973.50	974.76	975.84
	4	1020.11	**1021.17**	1021.10	1009.48	1016.56	1020.66
	5	1036.62	**1037.50**	1036.75	1030.09	1031.07	1036.56
	6	1054.58	**1055.15**	1054.51	1040.47	1046.95	1053.87
Male	**3**	**3126.86**	**3126.86**	**3126.86**	3121.79	3126.45	**3126.86**
	4	3208.70	**3208.77**	3208.75	3197.33	3207.06	3208.68
	5	3254.30	**3254.66**	3254.55	3238.38	3250.33	3254.37
	6	3281.87	**3283.79**	3283.70	3261.02	3279.26	3283.16
Lena	**3**	2128.29	**2128.31**	2128.29	2120.71	2127.04	2128.25
	4	2191.78	**2191.87**	2191.72	2178.44	2187.55	2191.54
	5	2216.59	**2217.42**	2216.20	2202.36	2207.61	2215.62
	6	2237.47	**2238.16**	2237.60	2217.72	2223.66	2235.53
Bridge2	**3**	3567.57	**3567.60**	**3567.60**	3560.70	3566.90	3567.56
	4	3660.20	**3660.30**	**3660.30**	3650.98	3659.19	3660.22
	5	3710.07	3710.18	**3710.21**	3688.03	3706.92	3709.89
	6	3733.86	**3736.01**	3735.74	3715.31	3730.56	3735.26
Lady	**3**	**2211.49**	**2211.49**	**2211.49**	2206.24	2210.79	**2211.49**
	4	2264.09	**2264.30**	**2264.30**	2251.73	2262.29	2264.19
	5	2294.67	**2295.31**	2295.30	2280.41	2291.85	2294.98
	6	2314.84	**2315.81**	2315.74	2297.08	2311.28	2314.99

Table 7.3. Best value of the Otsu's objective function using classical GWO, classical SCA and their proposed variants.

Test images	Thresholds	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
Cameraman	**3**	**3727.41**	**3727.41**	**3727.41**	3726.41	3727.37	**3727.41**
	4	3782.37	**3782.40**	**3782.40**	3780.80	3782.27	3782.37
	5	**3813.74**	3813.73	**3813.74**	3811.42	3813.11	3813.71
	6	3833.67	**3833.72**	3833.71	3828.45	3832.23	3833.53
Bridge	**3**	**2722.36**	**2722.36**	**2722.36**	2722.17	**2722.36**	**2722.36**
	4	**2822.71**	**2822.71**	**2822.71**	2821.39	**2822.71**	**2822.71**
	5	**2874.24**	**2874.24**	**2874.24**	2864.24	2874.08	**2874.24**
	6	2906.01	**2906.04**	**2906.04**	2902.75	2905.10	**2906.04**
Couple	**3**	**1383.16**	**1383.16**	**1383.16**	1382.82	**1383.16**	**1383.16**
	4	**1449.81**	**1449.81**	**1449.81**	1448.82	1449.48	**1449.81**
	5	1497.61	1497.61	**1497.62**	1491.57	1497.42	1497.57
	6	1525.73	**1525.83**	1525.82	1522.92	1525.08	1525.72
Peppers	**3**	**2703.57**	**2703.57**	**2703.57**	2703.18	2703.57	**2703.57**
	4	**2766.46**	**2766.46**	**2766.46**	2764.22	2765.99	**2766.46**
	5	**2810.84**	2810.82	**2810.84**	2803.79	2808.55	2810.77
	6	2833.68	**2833.74**	2833.59	2823.02	2829.25	2833.40
Airplane	**3**	**975.89**	**975.89**	**975.89**	975.77	975.77	**975.89**
	4	1021.21	**1021.22**	**1021.22**	1018.80	1020.41	**1021.22**
	5	1041.60	**1041.62**	1041.52	1040.55	1039.59	1040.97
	6	1055.16	**1055.24**	**1055.24**	1049.64	1054.03	1054.97
Male	**3**	**3126.86**	**3126.86**	**3126.86**	3126.46	**3126.86**	**3126.86**
	4	**3208.81**	**3208.81**	**3208.81**	3205.76	3208.41	3208.78
	5	3254.76	**3254.80**	3254.76	3252.71	3253.58	**3254.80**
	6	3284.06	**3284.10**	**3284.10**	3275.43	3282.34	3283.89
Lena	**3**	**2128.31**	**2128.31**	**2128.31**	2127.90	2128.16	**2128.31**
	4	**2191.87**	**2191.87**	**2191.87**	2191.05	2191.32	**2191.87**
	5	2217.77	**2217.78**	2217.68	2210.09	2215.01	2217.62
	6	**2238.40**	2238.39	2238.35	2231.56	2233.98	2237.93
Bridge2	**3**	**3567.60**	**3567.60**	**3567.60**	3566.55	3567.54	**3567.60**
	4	**3660.34**	**3660.34**	**3660.34**	3657.68	3660.05	**3660.34**
	5	**3710.26**	**3710.26**	**3710.26**	3707.40	3709.06	**3710.26**
	6	3736.19	3736.20	**3736.21**	3729.97	3734.71	3736.18
Lady	**3**	**2211.49**	**2211.49**	**2211.49**	2211.01	2211.40	**2211.49**
	4	2264.33	**2264.35**	**2264.35**	2261.46	2263.97	**2264.35**
	5	2295.60	2295.60	**2295.61**	2291.43	2295.00	2295.48
	6	2316.17	**2316.19**	2316.17	2309.06	2315.30	2316.02

Table 7.4. Mean PSNR value using classical GWO, classical SCA and their proposed variants

Test images	Thresholds	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
Cameraman	**3**	**20.21**	**20.21**	**20.21**	19.96	20.17	20.20
	4	21.41	**21.50**	21.49	21.27	21.47	21.47
	5	23.17	23.22	**23.25**	22.24	23.00	23.22
	6	23.71	23.73	23.78	23.03	23.59	**23.81**
Bridge	**3**	16.64	16.52	16.59	16.70	**16.79**	16.62
	4	18.98	18.78	**18.99**	18.63	18.96	18.98
	5	20.59	20.51	20.59	20.03	20.42	**20.61**
	6	22.08	22.17	22.17	21.05	21.77	**22.24**
Couple	**3**	17.22	**17.28**	**17.28**	16.99	17.26	17.27
	4	20.36	20.30	**20.41**	19.45	20.17	20.38
	5	20.78	21.47	21.47	20.70	21.43	**21.51**
	6	22.04	22.63	**22.65**	21.79	22.63	22.55
Peppers	**3**	18.47	**18.48**	**18.48**	18.38	18.40	18.47
	4	20.65	**20.66**	**20.66**	20.10	20.38	**20.66**
	5	22.25	**22.33**	22.32	21.39	21.71	22.25
	6	23.32	**23.47**	23.37	22.20	22.74	23.40
Airplane	**3**	19.27	19.30	19.30	19.23	19.06	**19.34**
	4	20.51	21.28	**21.51**	20.70	21.30	21.48
	5	22.82	**23.22**	23.20	22.53	22.33	23.06
	6	24.41	24.58	**24.59**	23.05	23.85	24.48
Male	**3**	**19.43**	**19.43**	**19.43**	19.31	19.34	**19.43**
	4	20.95	**20.98**	**20.98**	20.61	20.90	**20.98**
	5	22.54	**22.58**	**22.58**	21.90	22.40	22.56
	6	23.71	**23.93**	23.91	22.92	23.71	23.91
Lena	**3**	17.31	17.30	**17.32**	17.13	**17.32**	**17.32**
	4	**18.64**	18.62	**18.64**	18.46	18.62	18.59
	5	19.64	19.50	19.66	19.38	19.54	**19.71**
	6	**21.10**	20.72	20.74	20.29	20.52	20.90
Bridge2	**3**	18.34	18.34	**18.35**	18.31	18.31	18.34
	4	20.67	**20.70**	**20.70**	20.28	20.64	20.68
	5	22.10	**22.17**	**22.17**	21.53	22.05	22.14
	6	23.27	**23.44**	23.43	22.59	23.17	23.38
Lady	**3**	18.87	18.87	18.86	18.74	18.88	**18.91**
	4	21.15	21.15	21.17	20.89	21.05	**21.18**
	5	23.06	23.25	**23.29**	21.86	23.03	23.24
	6	24.30	24.56	**24.58**	23.11	24.13	24.51

Table 7.5. Median PSNR value using classical GWO, classical SCA and their proposed variants

Test images	Thresholds	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
Cameraman	**3**	**20.21**	**20.21**	**20.21**	20.06	20.19	**20.21**
	4	21.39	**21.52**	21.47	21.39	21.48	21.49
	5	23.20	23.25	23.26	22.33	23.07	**23.27**
	6	23.64	**23.73**	23.70	22.90	23.58	23.72
Bridge	**3**	16.61	16.47	16.59	16.69	**16.78**	16.58
	4	**19.01**	18.80	**19.01**	18.81	**19.01**	19.00
	5	**20.68**	20.47	20.59	20.04	20.48	20.57
	6	22.22	22.16	22.19	20.98	21.87	**22.26**
Couple	**3**	**17.28**	**17.28**	**17.28**	16.98	17.27	**17.28**
	4	**20.43**	20.42	20.41	19.73	20.28	20.40
	5	21.42	**21.48**	21.47	20.97	21.45	**21.48**
	6	22.41	22.94	**22.97**	21.85	22.70	22.90
Peppers	**3**	18.46	**18.49**	**18.49**	18.42	18.40	18.47
	4	20.66	20.67	20.66	20.07	20.38	**20.68**
	5	22.31	**22.34**	**22.34**	21.49	21.78	22.29
	6	23.37	**23.48**	23.40	22.27	22.75	23.41
Airplane	**3**	19.27	19.33	19.33	19.30	19.09	**19.35**
	4	21.13	21.46	**21.54**	21.01	21.42	**21.54**
	5	22.54	23.27	**23.35**	22.67	22.79	22.98
	6	24.39	**24.61**	24.51	23.11	24.03	24.47
Male	**3**	**19.43**	**19.43**	**19.43**	19.28	19.42	**19.43**
	4	20.96	20.97	20.97	20.63	20.89	**20.99**
	5	22.57	**22.59**	**22.59**	21.91	22.44	22.56
	6	23.79	**23.95**	23.93	22.90	23.71	23.91
Lena	**3**	**17.33**	17.29	**17.33**	17.17	17.32	**17.33**
	4	**18.65**	18.60	**18.65**	18.49	18.62	18.60
	5	19.42	19.35	19.67	19.16	19.40	**19.69**
	6	20.82	20.72	20.70	20.04	20.29	**20.85**
Bridge2	**3**	**18.34**	**18.34**	**18.34**	18.32	18.32	**18.34**
	4	20.68	**20.71**	**20.71**	20.39	20.64	20.68
	5	22.15	**22.18**	**22.18**	21.53	22.08	22.15
	6	23.30	**23.46**	**23.46**	22.53	23.16	23.40
Lady	**3**	18.85	18.83	18.83	18.76	18.90	**18.94**
	4	21.15	21.14	**21.19**	21.08	20.99	21.18
	5	23.22	23.27	**23.30**	22.14	23.04	23.25
	6	24.40	24.59	**24.60**	23.31	24.14	24.55

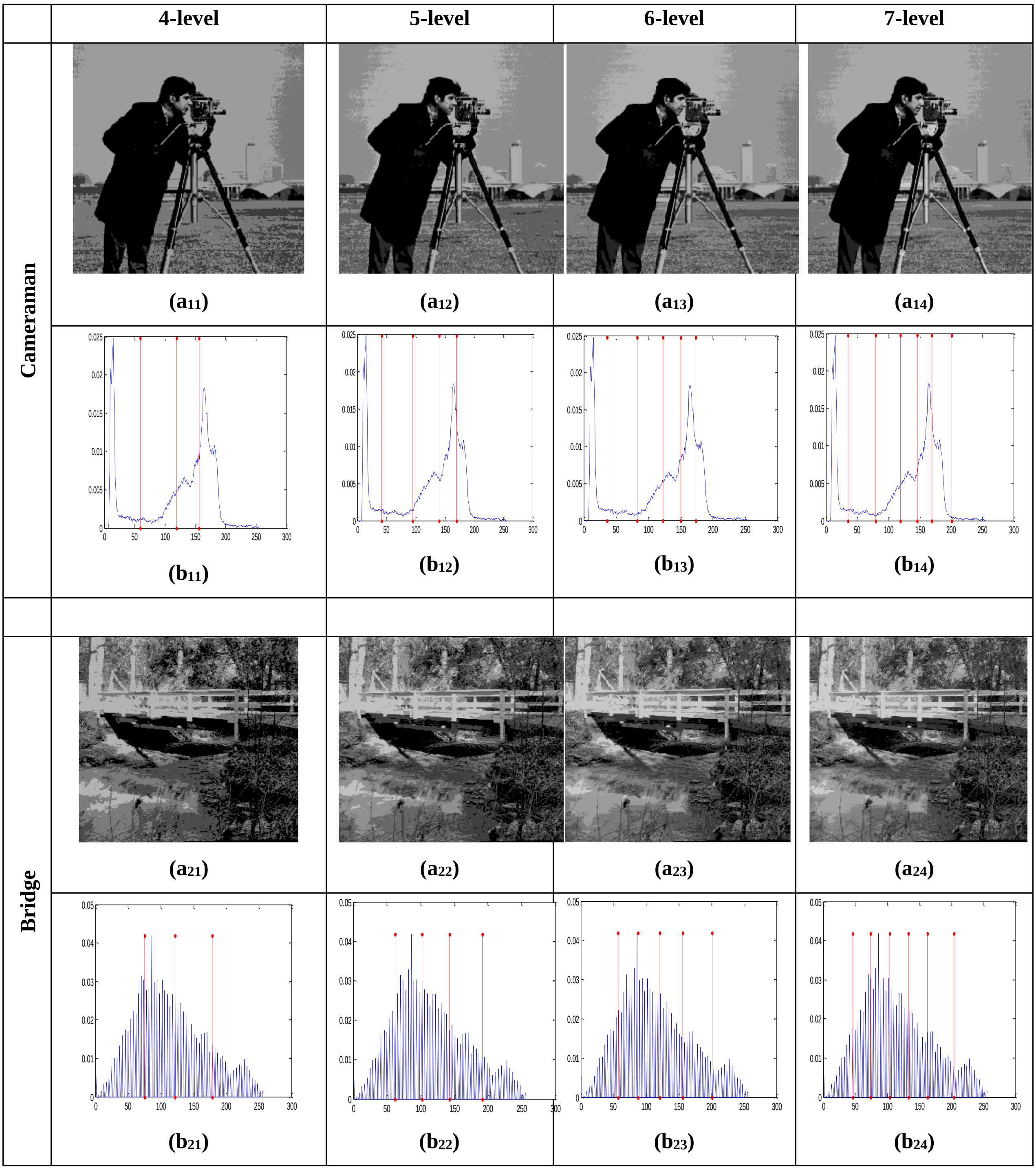


Fig 7.2. Segmented results obtained by RW-GWO algorithm. (a) represents the segmented image and (b) represents the optimal thresholds for test image (A) and (B)

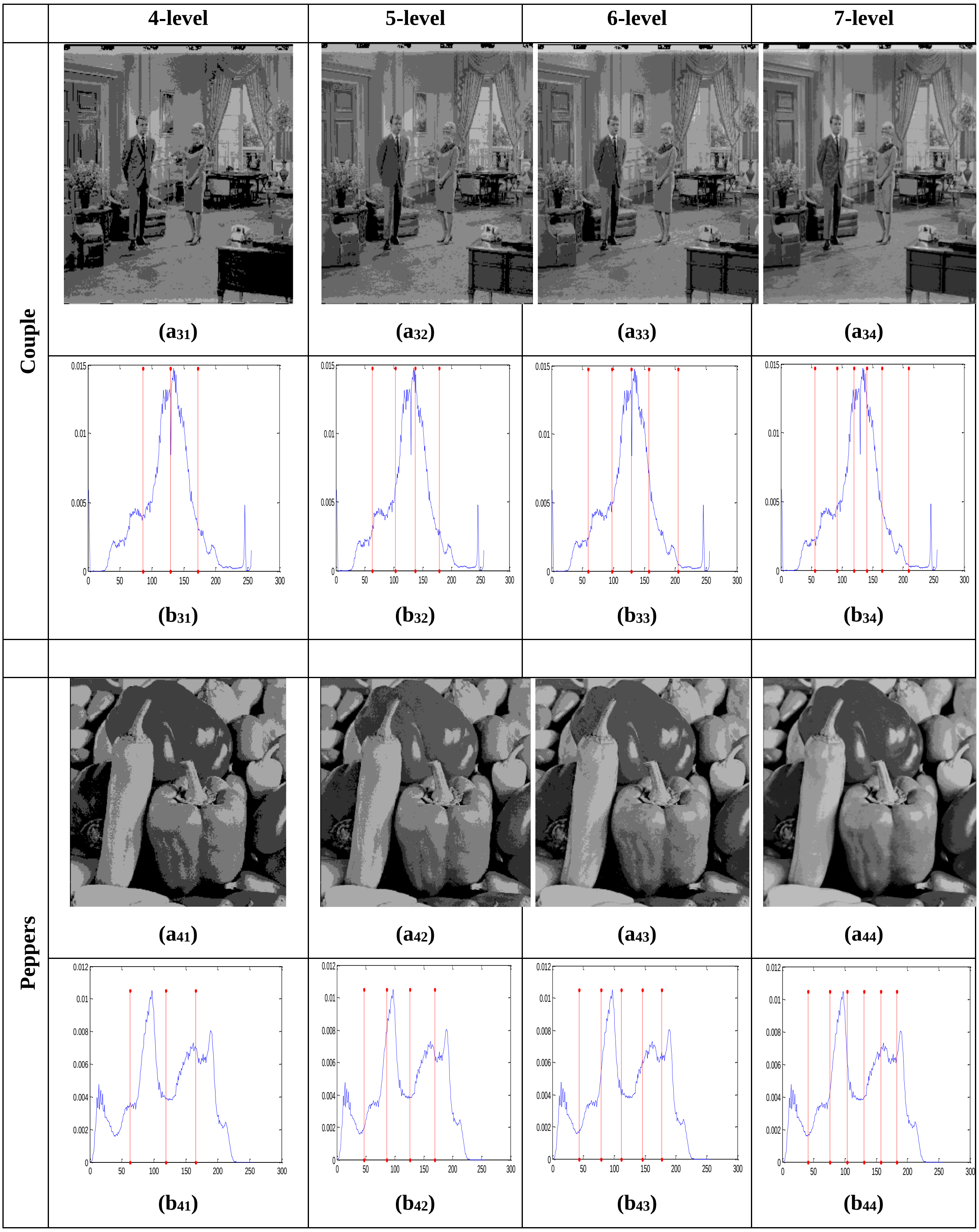


Fig 7.3. Segmented results obtained by RW-GWO algorithm. (a) represents the segmented image and (b) represents the optimal thresholds for test image (C) and (D)

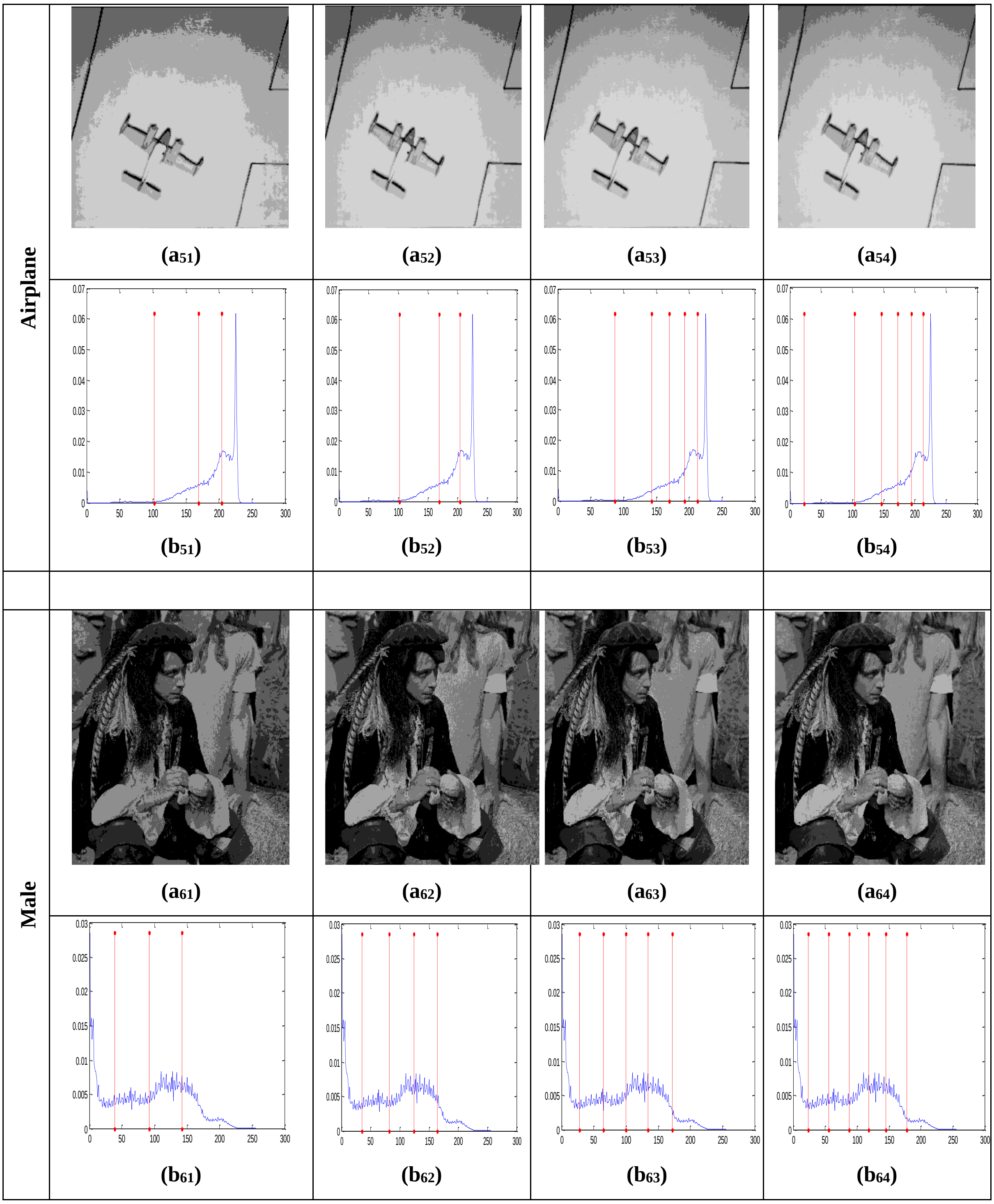


Fig 7.4. Segmented results obtained by RW-GWO algorithm. (a) represents the segmented image and (b) represents the optimal thresholds for test image (E) and (F)

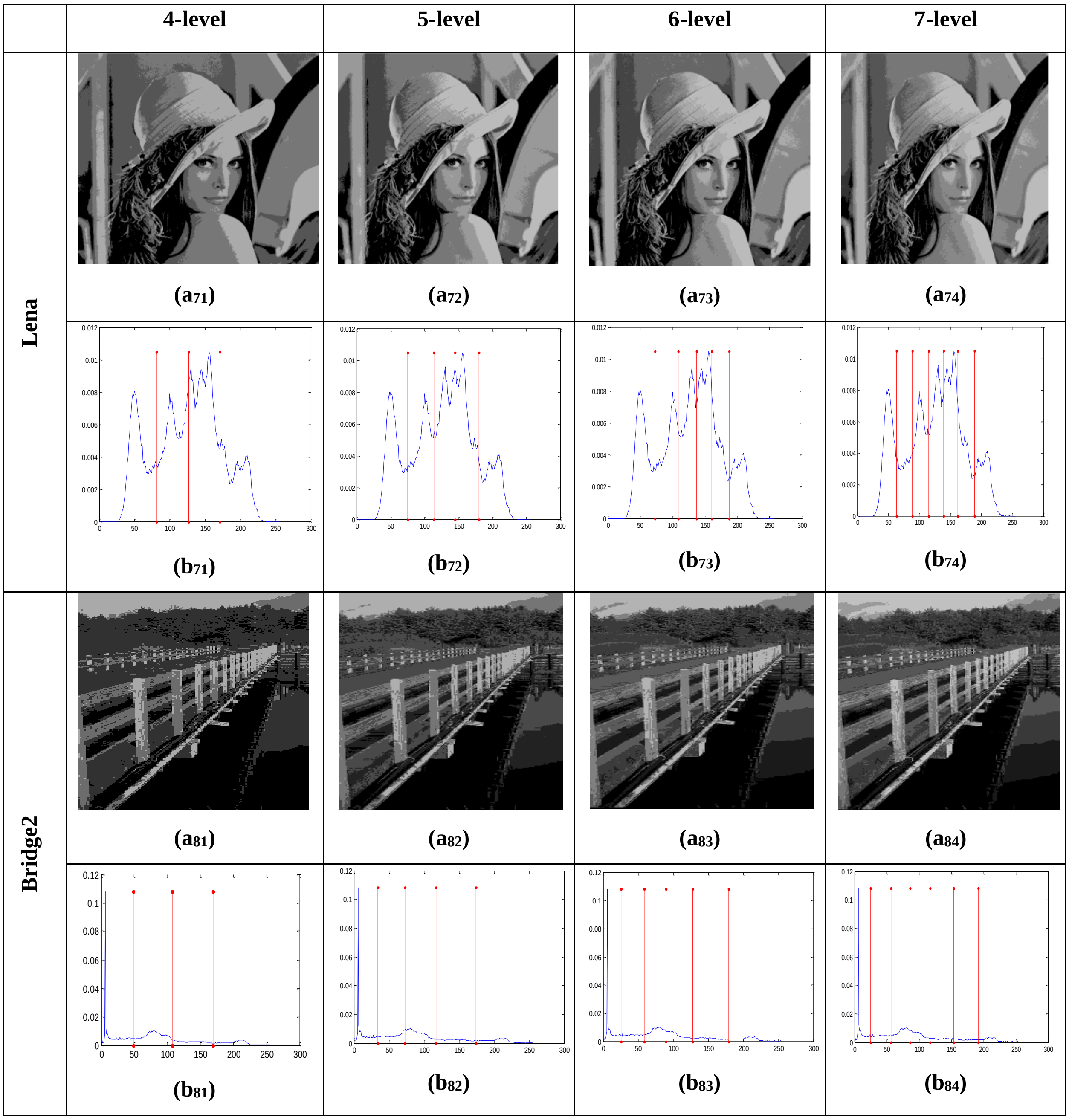


Fig 7.5. Segmented results obtained by RW-GWO algorithm. (a) represents the segmented image and (b) represents the optimal thresholds for test image (G) and (H)

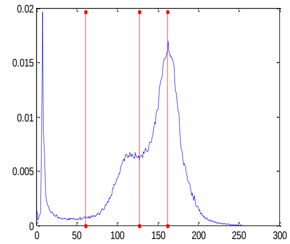
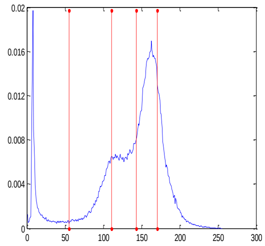
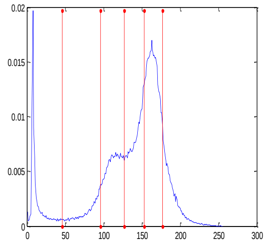
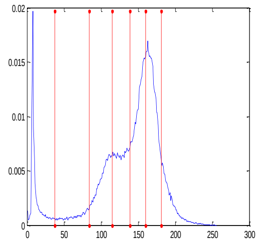
	4-level	5-level	6-level	7-level
Lady	 **(a_{91})**	 **(a_{92})**	 **(a_{93})**	 **(a_{94})**
	 **(b_{91})**	 **(b_{92})**	 **(b_{93})**	 **(b_{94})**

Fig 7.6. Segmented results obtained by RW-GWO algorithm. (a) represents the segmented image and (b) represents the optimal thresholds for test image (I)

Chapter 8

Optimal Coordination of Overcurrent Relays Using Proposed Variants of GWO and SCA

In this chapter, the classical versions of GWO and SCA and their proposed variants which are presented is Chapters 2, 3, 4 and 5 are used to solve a real life problem arising in the field of electrical engineering. The idea is to observe the effectiveness of solving real life problem using the proposed algorithms of the Thesis.

8.1. Introduction

Coordination of relays is a non-linear constrained optimization problem in a large distribution network where the operating time of all relays is to be minimized. Proper coordination of protective devices is a very crucial task for appropriate functioning of the electrical power system with distributed power generating stations. In the present chapter, the proposed variants of GWO and SCA along with their classical versions are employed to find the optimal setting for the proper coordination of overcurrent relays. The experiments are performed on IEEE 3, 4, 6, and 14-bus systems.

The organization of the chapter is as follows: Section 8.2 provides a detailed description of the relay coordination problem. Section 8.3 provides a numerical experimentation, analysis and comparison between classical GWO, classical SCA, and their proposed variants which are presented in Chapters 2, 3, 4, and 5. Finally the chapter concludes with Section 8.4.

8.2. Relay Coordination Problem

8.2.1. Background

The relay coordination problem has its origin in electrical power system and depends upon finding the optimal values of the decision variables for the devices called "Relays" which controls the act of isolation of faulty lines from the system without disturbing the healthy lines. Directional overcurrent relays (DOCRs) are provided in electrical power system to isolate the faulty lines only,

---

The content of this chapter is published in:

Gupta, S., & Deep, K. (2019). Optimal coordination of overcurrent relays using improved leadership based Grey Wolf Optimizer. Arabian Journal for Science and Engineering, Springer. (Accepted).

when the fault occurs in the system. These relays are placed at both the ends of each line. Hence, the number of DOCRs in an electrical power system is twice to the number of lines. In order to maintain the continuity of supply to healthy sections and to isolate the faulty sections relays are coordinated. This coordination ensures that the minimum lines are disrupted in the system when the fault occurs. This is accomplished in DOCRs by properly fixing the two adjustable parameters called “Time Dial Setting (TS)” and “Plug Setting (PS)” of each relay. The above stated problem of coordination between DOCRs can be modeled mathematically as a non-linear constrained optimization problem where the objective is to minimize the sum of operating time of all primary relays which are expected to operate when the fault occurs in the system.

Generally, there are two approaches that are used for directional overcurrent relay, conventional techniques and nature inspired techniques. In conventional techniques, first, the fault analysis is conducted and after that meshed networks are broken into radial form. Then relay at far end is set and in the last setup of back relays is established and this process is repeated for all relays. Final Time Dial Setting and Plug Setting depends on the selection of initial relays which are known as breakpoints [204]. These breakpoints are selected by using graph theory. The number of iterations here depends on the selection of breakpoints. It has been observed that the values of Time Dial Setting and Plug Setting determined by conventional techniques are not optimal [205]. Coordination of overcurrent relays in a large distribution network with multi-source and multi-looped network by using conventional optimization becomes infeasible [205]. Therefore, the unconventional parameter optimization techniques that are designed especially for highly non-linear or non-differentiable problems, are more effective for these type of problems. These techniques are called as nature inspired optimization techniques or metaheuristic techniques. The description and various examples of nature inspired optimization techniques are provided in the Chapter 1 of Thesis.

In [206], first time the optimization theory has been utilized to deal coordination of directional overcurrent relay. The practical importance of these problems inspires to apply different optimization strategies to solve this type of problems. In [207], linear programming is also used to find the optimal setting of parameters in overcurrent relay.

Various unconventional optimization algorithms are applied to find the optimal setting for the proper coordination of relays. For example: Genetic Algorithm (GA) and its improved variants are used [208-210] to determine the optimal coordination. In [211-213], Differential Evolution along with its modified variants are employed on the coordination of overcurrent relay problem to find the optimum setting of decision parameters TS and PS. In [214-219], Particle Swarm Optimization and its modified variants are used to solve coordination problem with mixed-integer programming formulation. In [220], Random Search Technique (RST) is applied to solve the coordination problem. Seeker Algorithm is applied in [221] with step length and adaptive search direction to find the decision parameters so that the operating time of all relay can be minimized.

In the present chapter the classical GWO, classical SCA and their proposed variants are evaluated on solving the relay coordination problem.

8.2.2. Problem Description

The relay operating time of directional overcurrent relay is a nonlinear function (T) of variables Time Dial Setting (TS), power flow i_f and Plug Setting (PS). A mathematical form of the problem [222] can be stated as –

$$T = \frac{\mu \times TS}{\left(\frac{i_f}{PS \times CT_{primary\,rating}}\right)^{\alpha} - \beta} \tag{8.1}$$

In the above problem, only TS and PS are the decision parameters. The constants μ, α and β represents the characteristics of a relay and are chosen as 0.14, 0.02 and 1.0 respectively as per the IEEE standards [223]. i_f is the fault current passing through a relay. $CT_{primary\,rating}$ is current transformer's primary rating. If the secondary rating of CT is 1.0 then the current sensed by the relay is –

$$i_r = \frac{i_f}{CT_{primary\,rating}} \tag{8.2}$$

A fault that occurs near to relay is called near-end-fault or close-in fault and the fault that occurs at other end is known as far-end-fault or far-bus fault for the same relay. n_c represents the number of relays responding to close-in fault and n_f is the number of relays responding to far-bus fault. The objective (or fitness) function of the problem is the integration of operating time of all primary relays which can be defined as –

$$obj = \sum_{j=1}^{n_c} T^{j}_{pr\_cl\_in} + \sum_{k=1}^{n_f} T^{k}_{pr\_far\_bus} \tag{8.3}$$

where
$$T^{j}_{pr\_cl\_in} = \frac{0.14 \times TDS^{j}}{\left(\frac{i_f^j}{PS^j \times CT^j_{primary\ rating}}\right)^{0.02} - 1} \tag{8.4}$$

and
$$T^{k}_{pr\_far\_bus} = \frac{0.14 \times TDS^{k}}{\left(\frac{i_f^k}{PS^k \times CT^k_{primary\ rating}}\right)^{0.02} - 1} \tag{8.5}$$

represent the response time of relay j and k to clear close-in and far-bus fault respectively.

Constraints

Bounds constraints for variable TS

For each relay, the value of TS should lie within their lower and upper limits i.e.

$$TS^{j}_{min} \leq TS^{j} \leq TS^{j}_{max} \qquad \forall j = 1,2, \dots, n_c \tag{8.6}$$

The values of TS^{j}_{min} and TS^{j}_{max} for each j is fixed as 0.05s and 1.10s.

Bounds constraints for variable PS

For each relay, the value of PS should lie within their lower and upper limits i.e.

$$PS^{j}_{min} \leq PS^{j} \leq PS^{j}_{max} \qquad \forall j = 1,2, \dots, n_f \tag{8.7}$$

The values of PS^{j}_{min} and PS^{j}_{max} for each j is fixed as 1.25 and 1.5.

Constraints for primary operation time

Each term of the objective function (obj) should be within the limit of 0.05s and 1s.

Selectivity constraints for relay pairs

A situation when the primary relays fail, backup relay operates at that time to prevent from mal-operation. The selectivity constraints maintain the selectivity of primary and backup relays. The sum of operating time of circuit breaker associated with primary relay and overshoot time is known as coordinate time interval (CTI). In order to maintain a suitable coordination between two overcurrent relays, the difference between the operating time of backup and primary relay should be greater than or equal to the coordinate time interval (CTI), i.e.

$$TM = T_{backup} - T_{primary} \geq CTI \tag{8.8}$$

$$CTI - \frac{0.14 \times TDS^n}{\left(\frac{i_f^j}{PS^j \times CT^j_{primary\ rating}}\right)^{0.02} - 1} + \frac{0.14 \times TDS^m}{\left(\frac{i_f^j}{PS^j \times CT^j_{primary\ rating}}\right)^{0.02} - 1} \leq 0 \tag{8.9}$$

where (m, n) is the combination of a primary relay m and backup relay n. The value of CTI is fixed and often set between 0.2s to 0.6s.

The constraint handling that has been used to solve this problems is as follows:

8.2.3. Constraint Handling

In the chapter, the two different constraint handling are utilized to solve the coordination of relays problem. In the first technique, a simple constraint handing technique based on constraint violation as described in Chapter 6 of the Thesis is used. The second constraint handling techniques which is applied is known as static penalty approach [224]. In this constraint handling technique, the objective function is defined as

$$obj = \begin{cases} F(X) & if\ the\ solution\ X\ is\ feasible \\ F(X) + \sum_{j=1}^{J} c_j \times \left[\max\left(0, g_j(X)\right)\right]^2 & otherwise \end{cases} \tag{8.10}$$

where c_j $(j = 1,2, \ldots, J)$ are the penalty parameters.

8.3. Experimental Setup and Results

In this chapter, the classical GWO, classical SCA, and their proposed variants namely RW-GWO (presented in Chapter 2), mGWO (presented in Chapter 3), m-SCA (presented in Chapter 4) and ISCA (presented in Chapter 5) are applied to solve this complex non-linear relay coordination problem. The population size is a very crucial parameter for any metaheuristic algorithm. The small size of the population fails to explore the search space of the problem while the large population size may fail to determine an efficient solution. Therefore, a suitable population size for algorithm should be chosen. In the present study, the population size is fixed as $10 \times$ dimension of the problem. For each algorithm 30 runs are performed for each bus-system and the best obtained results are recorded. The maximum number of function evaluations are fixed to 10^5 for IEEE 3, 4 and 6-bus system and 10^6 for the 14-bus system. For the comparison Random Search Technique (RST) [220], Differential Evolution (DE) [211], modified variants of Differential Evolution such as MDE1 [211], MDE2 [211], MDE3 [211], MDE4 [211], MDE5 [211], OCDE1

[213], OCDE2 [213], LX-PM [222] have been considered to analyse the comparative ability of proposed variants of GWO and SCA in finding the optimal setting for relay coordination problem. An algorithm corresponding to the first constraint handing which is based on the constraint violation is represented by A_1 and the algorithm corresponding to the second constraint handing which is based on the penalty approach is represented by A_2. Here, A represents the applied algorithm which can be GWO, RW-GWO, SCA, m-SCA or ISCA.

8.3.1. Test Models

Model I: IEEE 3-Bus, Model II: IEEE 4-Bus, Model III: 6-Bus system and Model IV: 14-Bus system

In the present section, the IEEE 3-Bus, 4-Bus, 6-Bus and 14-bus systems are used to evaluate the performance of classical GWO, classical SCA and their proposed variants which are presented in Chapters 2, 3, 4, and 5.

In the 3-bus system, a synchronous generator is used. The number of lines in this bus-system is 3 and number of relays is 6. In this system, the total number of decision parameters are 12 ($TS^1 - TS^6$ and $PS^1 - PS^6$). To optimize the parameter setting of this model the coordination of the setting of all the six relays is required. The values of $i_f^j, CT_{primary\ rating}^j, i_f^k$ and $CT_{primary\ rating}^k$ for the 3-bus system is presented in Table C1 of Appendix C, and the line diagram of this model is shown in Fig 8.1. The value of CTI is set to 0.3 in this model. This model includes 8 selectivity constraints and for this, the values of i_f and $CT_{primary\ rating}$ corresponding to primary and backup relay are presented in Table C2 of Appendix C. The obtained optimal decision variables by classical GWO, classical SCA, RW-GWO, mGWO, m-SCA, and ISCA are presented in Table 8.1.

IEEE 4-bus system consists of 16 decision parameters ($TS^1 - TS^8$ and $PS^1 - PS^8$), 2 generators and 8 relays. In the 4-Bus system number of selectivity constraints are 9. The values of $i_f^j, CT_{primary\ rating}^j, i_f^k$ and $CT_{primary\ rating}^k$ for the 4-bus system is presented in Table C3 of Appendix C and the line diagram of this system is shown in Fig 8.2. The value of CTI is fixed to 0.3 in this model and the values of i_f and $CT_{primary\ rating}$ corresponding to primary and backup relay are presented in Table C4 of Appendix C. The obtained optimal decision parameters (TS and PS) from various algorithms are presented in Table 8.2.

IEEE 6-Bus system consists of 7 lines, 3 generators and 14 relays. In this problem, 28 decision variables ($TS^1 - TS^{14}$ and $PS^1 - PS^{14}$) are involved. The values of $i_f^j, CT_{primary\ rating}^j, i_f^k$ and $CT_{primary\ rating}^k$ for the 6-bus system is presented in Table C5 of Appendix C. The line diagram of this model is presented in Fig 8.3. The value of CTI is 0.2 for this model. In this model, 48 selectivity constraints are involves corresponding to all possible near-end and far-end faults that are sensed by relays of the power system. The values of i_f and $CT_{primary\ rating}$ corresponding to primary and backup relay are presented in Table C6 of Appendix C. The obtained optimal decision variables TS and PS obtained from various algorithms are presented in Table 8.3.

In a 14-Bus system, 80 decision variables ($TS^1 - TS^{40}$ and $PS^1 - PS^{40}$) are involved. The data for the 14-bus system can be found in [222] and is provided in Appendix C. The line diagram of this model is presented in Fig 8.4. The value of CTI is 0.2 for this model. In this model, 145 selectivity constraints are involves corresponding to all possible near-end and far-end faults that are sensed by relays of the power system. The values of $i_f^j, CT_{primary\ rating}^j, i_f^k$ and $CT_{primary\ rating}^k$ for the 6-bus system is presented in Table C7 of Appendix C. The values of i_f and $CT_{primary\ rating}$ corresponding to primary and backup relay are presented in Table C8 of Appendix C. The obtained optimal decision variables TS and PS from various algorithms are presented in Tables 8.4 and 8.5 respectively.

The obtained objective fitness value (minimum operating time of all relay) is reported in Table 8.6 corresponding to all bus systems. By analyzing the results presented in this table, it can be observed that the results obtained from DE and its variants (MDE1, MDE2, MDE3, MDE4, MDE5) [211], OCDE1 and OCDE2 [213], are infeasible while the optimal setting obtained by RST [220], LX-PM [222], classical GWO [75], classical SCA [145], RW-GWO, mGWO, m-SCA, and ISCA is feasible for 3 and 4-bus systems. For 6 and 14-bus systems, LX-PM, classical-GWO, RW-GWO, mGWO, and ISCA gives feasible solution. From the results, presented in Table 8.6, it can be concluded that the results obtained from RW-GWO are better than other algorithms under consideration.

8.4. Concluding Remarks

Coordination of directional overcurrent relays is a very trending and complex non-linear problem in the field of electrical engineering. The problem consists of a large number of constraints which make the problem more difficult to solve as compared to unconstrained problems. The number of decision parameters are also large in this problem. In the present chapter, to find the optimal setting for overcurrent relays, classical versions of GWO and SCA, and their proposed variants are employed. The comparative analysis of the results between these algorithms and some state-of-the-art algorithms available in the literature demonstrate the better search efficiency of the proposed RW-GWO algorithm to provide not only feasible solutions but also better and efficient solutions.

Table 8.1. Optimal decision variables for 3-bus system obtained by classical GWO, classical SCA and their proposed variants

Variable	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
$\boldsymbol{TS^1}$	0.0500	0.0500	0.0500	0.0503	0.0584	0.0500
$\boldsymbol{TS^2}$	0.1982	0.1979	0.1989	0.2462	0.2347	0.2026
$\boldsymbol{TS^3}$	0.0500	0.0500	0.0500	0.0500	0.0540	0.0500
$\boldsymbol{TS^4}$	0.2103	0.2096	0.2095	0.2729	0.2574	0.2153
$\boldsymbol{TS^5}$	0.1818	0.1830	0.1830	0.2142	0.1990	0.1845
$\boldsymbol{TS^6}$	0.1823	0.1807	0.1818	0.1924	0.2040	0.1810
$\boldsymbol{PS^1}$	1.2503	1.2506	1.2506	1.2946	1.2826	1.2527
$\boldsymbol{PS^2}$	1.4967	1.4976	1.4865	1.5000	1.3796	1.4661
$\boldsymbol{PS^3}$	1.2500	1.2500	1.2518	1.2500	1.3065	1.2512
$\boldsymbol{PS^4}$	1.4838	1.4984	1.5000	1.2500	1.4646	1.4268
$\boldsymbol{PS^5}$	1.4897	1.4666	1.4692	1.5000	1.4525	1.4552
$\boldsymbol{PS^6}$	1.4723	1.4999	1.4816	1.5000	1.2950	1.5000

Table 8.2. Optimal decision variables for 4-bus system obtained by classical GWO, classical SCA and their proposed variants

Variable	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
TS^1	0.0500	0.0501	0.0500	0.0728	0.0795	0.0500
TS^2	0.2158	0.2128	0.2127	0.2405	0.2394	0.2137
TS^3	0.0500	0.0500	0.0500	0.0500	0.0596	0.0501
TS^4	0.1257	0.1260	0.1259	0.1454	0.1440	0.1280
TS^5	0.1267	0.1273	0.1268	0.1448	0.1809	0.1272
TS^6	0.0500	0.0500	0.0500	0.0500	0.0552	0.0500
TS^7	0.1351	0.1341	0.1342	0.1664	0.1779	0.1402
TS^8	0.0500	0.0500	0.0500	0.0500	0.0549	0.0502
PS^1	1.2623	1.2524	1.2773	1.5000	1.3207	1.2616
PS^2	1.4274	1.4881	1.4924	1.5000	1.4964	1.4856
PS^3	1.2515	1.2500	1.2500	1.2500	1.2921	1.2505
PS^4	1.5000	1.4948	1.4933	1.5000	1.3265	1.4778
PS^5	1.4951	1.4760	1.4981	1.5000	1.4132	1.4977
PS^6	1.2526	1.2513	1.2516	1.3413	1.3813	1.2500
PS^7	1.4731	1.4936	1.4928	1.2500	1.3792	1.3703
PS^8	1.2511	1.2529	1.2500	1.2500	1.4249	1.2500

Table 8.3. Optimal decision variables for 6-bus systems obtained by classical GWO, classical SCA and their proposed variants

Variable	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
TS^1	0.1061	0.1157	0.1109	0.2538	0.2507	0.1190
TS^2	0.1897	0.1867	0.2024	0.2367	0.3983	0.2051
TS^3	0.0974	0.0967	0.1011	0.1072	0.2388	0.1071
TS^4	0.1078	0.1046	0.1095	0.2412	0.3435	0.1187
TS^5	0.0503	0.0503	0.0500	0.0813	0.1120	0.0507
TS^6	0.0502	0.0528	0.0503	0.0500	0.1852	0.0698
TS^7	0.0500	0.0500	0.0500	0.0500	0.0795	0.0505
TS^8	0.0502	0.0500	0.0501	0.0908	0.0851	0.0501
TS^9	0.0501	0.0500	0.0501	0.0500	0.0910	0.0509
TS^{10}	0.0669	0.0531	0.0647	0.0643	0.2583	0.0691
TS^{11}	0.0655	0.0655	0.0667	0.0750	0.2122	0.0798
TS^{12}	0.0527	0.0512	0.0598	0.1129	0.2260	0.0689
TS^{13}	0.0500	0.0502	0.0503	0.1032	0.2361	0.0524
TS^{14}	0.0739	0.0809	0.0888	0.0687	0.3635	0.0901
PS^1	1.4485	1.2937	1.4088	1.5000	1.3761	1.4263
PS^2	1.4596	1.5000	1.3148	1.4166	1.3126	1.3426
PS^3	1.2554	1.2577	1.2578	1.2500	1.2658	1.2500
PS^4	1.3534	1.4333	1.3205	1.2500	1.2985	1.2549
PS^5	1.2625	1.2500	1.2580	1.2500	1.3296	1.2515
PS^6	1.4120	1.3483	1.3999	1.5000	1.4309	1.2500
PS^7	1.2500	1.2506	1.2500	1.2500	1.3146	1.2500
PS^8	1.2500	1.2504	1.2512	1.2500	1.2967	1.2711
PS^9	1.2522	1.2624	1.2523	1.5000	1.3741	1.2706
PS^{10}	1.2516	1.4814	1.3723	1.2500	1.3019	1.3990
PS^{11}	1.4966	1.4946	1.4839	1.5000	1.4213	1.3455
PS^{12}	1.4751	1.4916	1.3320	1.5000	1.3897	1.2852
PS^{13}	1.4518	1.4419	1.4545	1.2669	1.2891	1.4468
PS^{14}	1.4552	1.3397	1.2592	1.5000	1.3194	1.3382

Table 8.4. Optimal decision variables (TS) for 14-bus systems obtained by classical GWO, classical SCA and their proposed variants

Variable	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
TS^1	0.06198	0.05477	0.0506	0.0620	0.2489	0.0582
TS^2	0.05004	0.05001	0.0500	0.1361	0.8453	0.0500
TS^3	0.05232	0.05781	0.0698	0.1055	0.1374	0.0633
TS^4	0.05021	0.05024	0.0530	0.0519	0.1556	0.0568
TS^5	0.06395	0.09155	0.0548	0.0864	0.2400	0.0617
TS^6	0.07145	0.09512	0.0776	0.1899	0.1994	0.1082
TS^7	0.05610	0.05105	0.0529	0.0688	0.2129	0.0776
TS^8	0.14305	0.09676	0.0964	0.0763	0.3722	0.1280
TS^9	0.05625	0.06262	0.0767	0.0588	0.4121	0.0682
TS^{10}	0.16654	0.16577	0.1523	0.2152	0.3565	0.1377
TS^{11}	0.09641	0.14441	0.1177	0.0774	0.4902	0.1168
TS^{12}	0.26447	0.22398	0.2239	0.3101	0.4827	0.2306
TS^{13}	0.14515	0.13514	0.1462	0.1248	0.3288	0.1384
TS^{14}	0.15452	0.12838	0.1222	0.1055	0.3682	0.1337
TS^{15}	0.13928	0.12454	0.1212	0.1135	0.2842	0.1354
TS^{16}	0.22346	0.22072	0.2345	0.1076	0.2929	0.2329
TS^{17}	0.11400	0.11502	0.1358	0.1287	0.3118	0.1274
TS^{18}	0.22352	0.22099	0.2265	0.2623	0.2662	0.2289
TS^{19}	0.13513	0.16106	0.1471	0.2138	0.3652	0.1595
TS^{20}	0.16437	0.15358	0.1206	0.0835	0.4001	0.1520
TS^{21}	0.21800	0.22633	0.2159	0.3729	0.4081	0.2041
TS^{22}	0.44256	0.42088	0.4464	0.0786	0.8469	0.4718
TS^{23}	0.06458	0.05155	0.0667	0.1509	0.1116	0.0509
TS^{24}	0.39345	0.39445	0.3832	0.3193	0.3265	0.3878
TS^{25}	0.05125	0.05730	0.0536	0.0553	0.1906	0.0566
TS^{26}	0.30668	0.31637	0.3035	0.2377	0.5692	0.2961
TS^{27}	0.34629	0.34502	0.3622	0.1872	0.4572	0.3437
TS^{28}	0.14909	0.06079	0.0656	0.0797	0.3596	0.0682
TS^{29}	0.13063	0.12487	0.1347	0.1165	0.3014	0.1354
TS^{30}	0.28402	0.28458	0.3003	0.2040	0.3822	0.2781
TS^{31}	0.27896	0.25340	0.2636	0.5663	0.5148	0.2789
TS^{32}	0.45489	0.45093	0.4498	0.2273	0.6004	0.4373
TS^{33}	0.23319	0.23236	0.2351	0.0514	0.3353	0.2280
TS^{34}	0.25396	0.23126	0.2469	0.1857	0.4755	0.2570
TS^{35}	0.40449	0.38248	0.4044	0.0956	0.9037	0.4183
TS^{36}	0.36261	0.36562	0.3794	0.4979	0.6438	0.3561
TS^{37}	0.22788	0.22194	0.2863	0.2761	0.3936	0.2418
TS^{38}	0.45167	0.44692	0.4569	0.3095	0.6175	0.4391
TS^{39}	0.19112	0.16921	0.1979	0.1699	0.4795	0.1905
TS^{40}	0.38386	0.36702	0.3887	0.1173	0.6361	0.3684

Table 8.5. Optimal decision variables (PS) for 14-bus systems obtained by classical GWO, classical SCA and their proposed variants

variable	GWO	RW-GWO	mGWO	SCA	m-SCA	ISCA
PS^1	1.28920	1.36180	1.4214	1.4202	1.3329	1.4581
PS^2	1.49830	1.50000	1.5000	1.2508	1.4945	1.5000
PS^3	1.30480	1.25060	1.2512	1.3871	1.3629	1.2730
PS^4	1.26240	1.25070	1.2509	1.4278	1.3155	1.2783
PS^5	1.39130	1.31220	1.4485	1.2906	1.2780	1.4587
PS^6	1.25930	1.25320	1.2542	1.2763	1.4349	1.4143
PS^7	1.25160	1.43290	1.3427	1.3905	1.3853	1.4292
PS^8	1.47730	1.26050	1.3581	1.3123	1.2915	1.3512
PS^9	1.38950	1.30900	1.3562	1.4754	1.3920	1.4528
PS^{10}	1.31950	1.25570	1.3195	1.4147	1.4734	1.4292
PS^{11}	1.40660	1.33260	1.3361	1.2935	1.3846	1.4350
PS^{12}	1.26210	1.26000	1.3574	1.4756	1.3533	1.3811
PS^{13}	1.25860	1.34520	1.3050	1.4877	1.3325	1.4654
PS^{14}	1.38460	1.40520	1.2872	1.3925	1.4456	1.3643
PS^{15}	1.37680	1.40250	1.3931	1.2935	1.3333	1.3256
PS^{16}	1.35750	1.39590	1.3626	1.2580	1.3335	1.3524
PS^{17}	1.39810	1.29270	1.2577	1.3858	1.4552	1.3357
PS^{18}	1.25340	1.31380	1.3648	1.2642	1.3661	1.3569
PS^{19}	1.37660	1.31660	1.4157	1.3959	1.3599	1.4132
PS^{20}	1.38790	1.25650	1.4933	1.2991	1.3018	1.2935
PS^{21}	1.36230	1.26820	1.4460	1.5000	1.3110	1.4924
PS^{22}	1.39730	1.25210	1.3349	1.3861	1.3753	1.2547
PS^{23}	1.29350	1.44090	1.2790	1.2524	1.4014	1.4419
PS^{24}	1.30200	1.33530	1.4463	1.4410	1.4308	1.3981
PS^{25}	1.30060	1.30140	1.2918	1.2890	1.2737	1.4678
PS^{26}	1.37090	1.25520	1.4613	1.3333	1.4747	1.4576
PS^{27}	1.44080	1.37480	1.3083	1.2875	1.2767	1.4156
PS^{28}	1.38720	1.42320	1.3882	1.4941	1.3910	1.3109
PS^{29}	1.46160	1.38970	1.2978	1.4929	1.4694	1.3008
PS^{30}	1.39240	1.39440	1.2832	1.3930	1.2626	1.4508
PS^{31}	1.34730	1.31250	1.3661	1.4477	1.4452	1.2760
PS^{32}	1.27690	1.35400	1.4199	1.4523	1.3626	1.3890
PS^{33}	1.46560	1.47690	1.4826	1.2956	1.4146	1.4831
PS^{34}	1.34450	1.25950	1.4822	1.3733	1.4436	1.4537
PS^{35}	1.41080	1.30740	1.2582	1.2904	1.3302	1.3930
PS^{36}	1.44230	1.44850	1.3281	1.4775	1.3086	1.4275
PS^{37}	1.29490	1.28660	1.4536	1.4233	1.4183	1.3153
PS^{38}	1.30330	1.26930	1.2928	1.4723	1.2654	1.2998
PS^{39}	1.44550	1.28450	1.3824	1.2939	1.4865	1.3001
PS^{40}	1.27460	1.43590	1.2918	1.4468	1.3092	1.4107

Table 8.6. Comparison of results on IEEE 3, 4, 6 and 14-bus systems obtained by various algorithms. * represents the infeasibility of the obtained solution.

Algorithm	3-bus system	4-bus system	6-bus system	14-bus system
RST	4.8354	3.7050	-	-
DE	4.8422*	3.6774*	10.6272*	42.7843*
MDE1	4.8070*	3.6694*	10.5067*	-
MDE2	4.7873*	3.6734*	10.6238*	-
MDE3	4.7822*	3.6692*	10.4370*	-
MDE4	4.7806*	3.6674*	10.3812*	-
MDE5	4.7806*	3.6694*	10.3514*	-
OCDE1	4.7806*	3.6674*	10.3479*	37.3540*
OCDE2	4.7806*	3.6674*	10.3286*	37.4603*
LX-PM	4.8340	3.7029	10.4581	37.2881
GWO_1	4.7865	3.5710	10.3722	44.2596
GWO_2	4.7865	3.5727	10.3590	38.8120
RW-GWO_1	4.7876	3.5698	10.4104	42.7644
RW-GWO_2	**4.7852**	**3.5695**	**10.3368**	**37.3062**
mGWO_1	4.8262	3.5989	11.2420	44.1539
mGWO_2	4.7896	3.5706	10.5678	38.0012
SCA_1	5.4779	4.0469	11.0386*	28.3478*
SCA_2	5.4761	4.0469	15.0366*	28.3478*
m-SCA_1	5.4978	4.6601	30.8802*	45.3325*
m-SCA_2	5.4347	4.4143	29.5899*	45.3325*
$ISCA_1$	4.8160	3.5892	11.1415	40.1895
$ISCA_2$	4.8161	3.5907	11.2142	39.0246

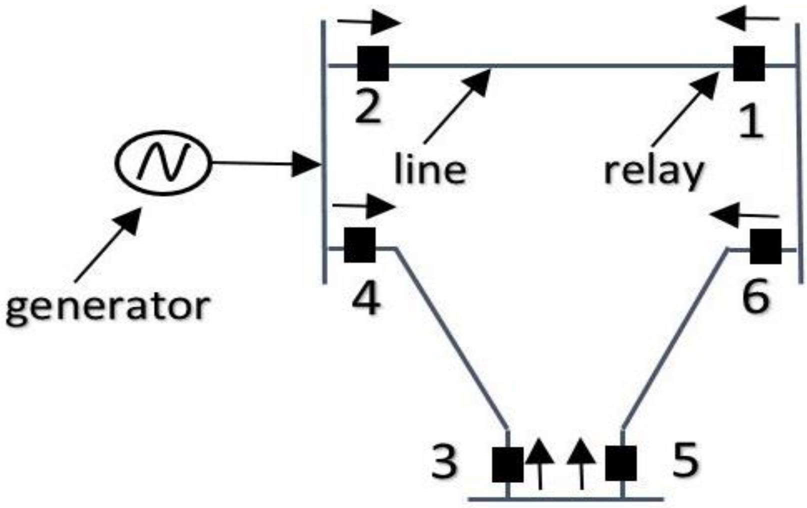


Fig 8.1. IEEE 3-bus system

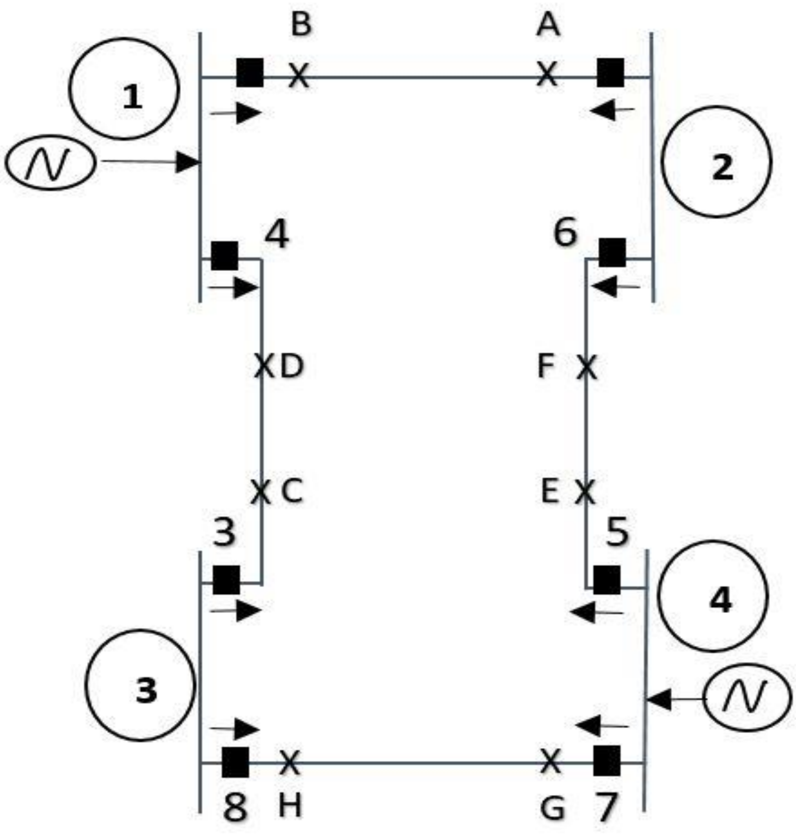


Fig 8.2. IEEE 4-bus system

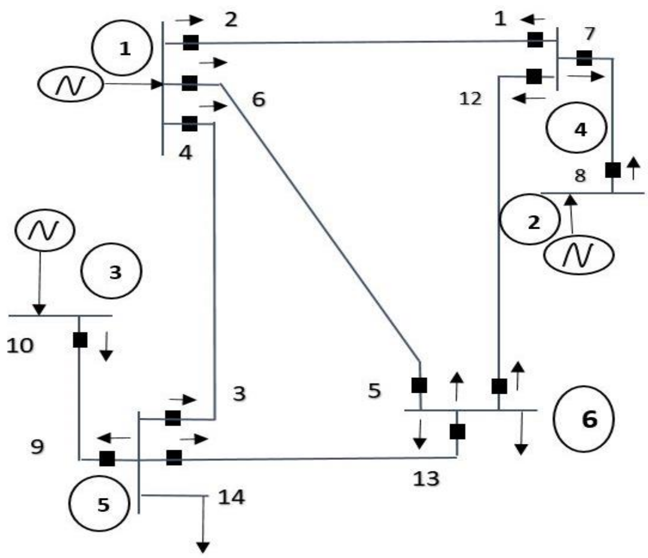


Fig 8.3. IEEE 6-bus system

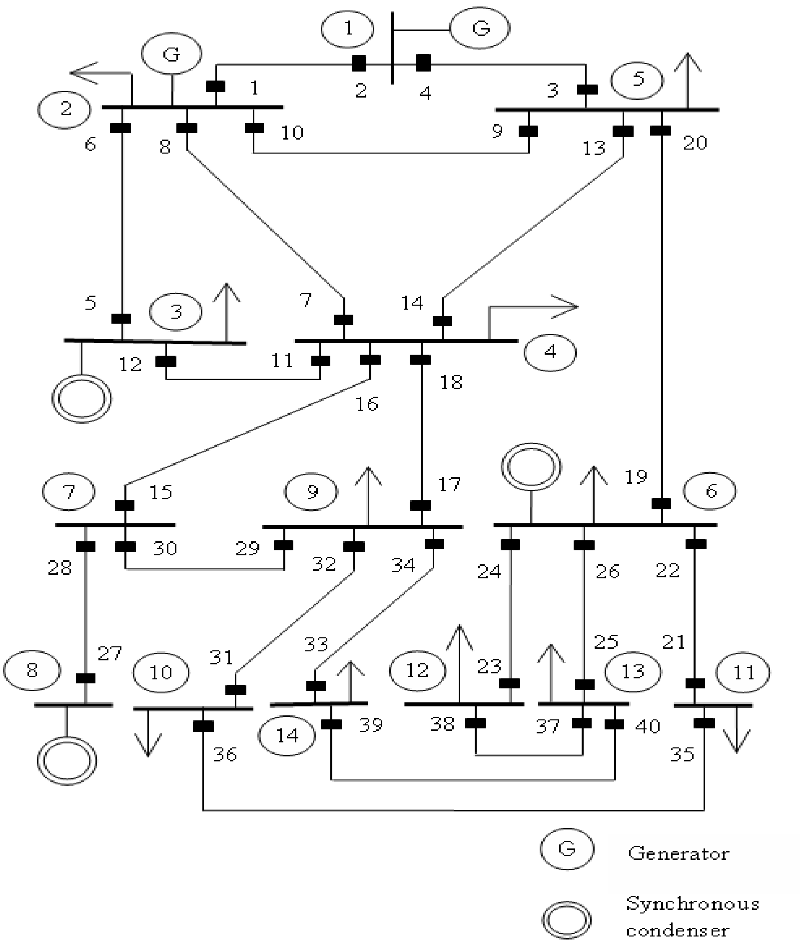


Fig 8.4. IEEE 14-bus system

Chapter 9

Conclusions and Future scope

This chapter forms the concluding part of this Thesis and also proposes some suggestions towards which the present work can be further continued. It consists of two sections; Section 9.1 brings out the overall conclusions of the research carried out in this Thesis and in Section 9.2, suggestions regarding the future research directions and possible extensions of the work presented in the Thesis are made.

9.1. Conclusions

The primary goal of the Thesis is to improve the performance of classical versions of Grey Wolf Optimizer (GWO) and Sine Cosine Algorithm (SCA) which are recently developed nature inspired optimization algorithms. The second aim of the work is to implement the proposed modified variants of GWO and SCA on real-life problems. For this two problems are selected from the field of electrical engineering and the other is from the field of image processing. The underlying focus of improving the performance of GWO and SCA is to improve the communication system among the wolves/candidate solutions for better exploration and exploitation of the provided search space. The Thesis proposes two variants of GWO and two variants of SCA for unconstrained and constrained optimization problems and for solving the real-life problems.

Chapter 1 introduces the field of Nature Inspired Optimization and presents the conceptual details, modeling and computational steps of classical GWO and SCA. Then the recent developments in designing modified GWO and SCA variants, its applications in different areas of studies and related literature are discussed. At the end of this chapter, the organization of the Thesis is discussed in brief. It is concluded that a lot of scope of research exists in this area of nature inspired optimization.

Chapter 2 introduces an improved variant of the classical GWO called RW-GWO which is based on enhancing the leading search ability of grey wolf pack, In the RW-GWO, cauchy distributed random walk is used to update the leading wolves and the greedy selection mechanism is applied between two consecutive iterations corresponding to each grey wolf. The cauchy random walk strategy maintains the exploration and exploitation of search space and helps the leaders to escape from the situation of stagnation at local optima. The greedy selection mechanism maintains the balance between exploration and exploitations and avoids the wolves to diverge from discovered promising search regions. The performance of the proposed RW-GWO algorithm is tested, analyzed and compared with classical GWO on unconstrained benchmark optimization problems given in IEEE CEC 2014. The test set consists of variety of benchmark problems with varying difficulty levels. It contains 3 unimodal problems, 13 multimodal problems, 6 hybrid problems and 8 composite problems. The comparison between the classical GWO and the RW-GWO is performed on the basis of various criteria such as minimum, median, mean, maximum and standard deviation of absolute error values in the objective function. Statistical significance of the obtained results from the RW-GWO is tested using Wilcoxon signed rank test and the convergence rate is analyzed through convergence curves. The computational complexity of the RW-GWO is also evaluated and compared with the classical GWO. Overall comparison of the results demonstrates the better performance of the RW-GWO as compared to the classical GWO. It is concluded that the RW-GWO algorithm is a better optimizer than the classical GWO for different category of unconstrained optimization problems.

Chapter 3 introduces another variant of classical GWO called mGWO which is developed by improving the search mechanism of grey wolf pack through a modified encircling and hunting mechanism, proposing a new search strategy to integrate the personal best guidance, and greedy selection. The modified position update mechanism increases the exploration and exploitation of available promising search regions and enhances the collaborative strength of the grey wolf pack. The greedy selection maintains the balance between exploration and exploitation, and prevents from skipping available promising regions of the search space. The performance of the proposed mGWO is tested, analyzed and compared with classical GWO on the same benchmark set given in IEEE CEC 2014 and used in Chapter 2 of the Thesis. Statistical significance of the obtained results from the mGWO is tested using Wilcoxon signed rank test and the convergence rate is analyzed through convergence curves. The computational complexity of the mGWO is also evaluated and compared with the classical GWO. Overall comparison of the results demonstrates the better performance of the mGWO as compared to the classical GWO. Later in this chapter, comparison between the mGWO and the RW-GWO (presented in Chapter 2) is also performed based on various criteria such as minimum, median, mean, maximum and standard deviation of absolute error values in the objective function, statistical analysis through Wilcoxon signed rank test and convergence behavior analysis.

From the detailed comparative analysis the following conclusions can be made:

1. In terms of worst time complexity calculated through big$-O$ notation, all the algorithms classical GWO, RW-GWO and mGWO are same.

2. On unimodal problems, the RW-GWO performs better than the classical GWO and mGWO.
3. On multimodal and hybrid problems, the mGWO algorithm is more successful as compared to the classical GWO and RW-GWO.
4. On the set of composite problems, the RW-GWO algorithm is more successful as compared to the classical GWO and mGWO.
5. The convergence rate is better in RW-GWO as compared to classical GWO and mGWO.

Chapter 4 introduces a modified variant of classical SCA called m-SCA which is based on the concept of opposite-based learning and modified position update mechanism. In the m-SCA, opposition-based learning is used to generate the opposite candidate solutions so that the stagnation at local optima can be avoided. The small jumping rate is chosen to sustain the balance between exploration and exploitation in the opposition-based learning. The search equation of classical SCA is modified to reduce the inefficient diversity of the population and to maintain the balance between exploration and exploitation during the search. The performance of the proposed m-SCA is tested, analyzed and compared with classical SCA on the same benchmark problems which are given in IEEE CEC 2014 and used in previous chapters. The comparison between the classical SCA and the m-SCA is performed based on various criteria such as minimum, median, mean, maximum and standard deviation of absolute error values obtained in the objective function. Statistical significance of the obtained results from the m-SCA is tested using Wilcoxon signed rank test and the convergence rate is analyzed through convergence curves. The computational complexity of the m-SCA is also evaluated and compared with the classical SCA. Overall comparison of the results demonstrates the better performance of the m-SCA as compared to the classical SCA. It is concluded that the m-SCA is a better optimizer than the classical SCA for different category of unconstrained optimization problems.

Chapter 5 introduces another variant of classical SCA called ISCA which enhances the performance of the classical SCA based on the crossover operator and modified position update mechanism. The search equation of the classical SCA is modified with the help of personal best state of candidate solutions and by adding the social component. In the ISCA, a greedy selection mechanism is also applied for each candidate solution between two consecutive iterations to avoid the divergence of candidate solutions from discovered promising search regions. The performance of the proposed ISCA is tested, analyzed and compared with classical SCA on the same benchmark problems which are given in IEEE CEC 2014 and used in previous chapters. The comparison between the classical SCA and the ISCA is performed based on various criteria such as minimum, median, mean, maximum and standard deviation of absolute error values obtained in the objective function. Statistical significance of the obtained results from the ISCA is tested using Wilcoxon signed rank test and the convergence rate is analyzed through convergence curves. The computational complexity of the ISCA is also evaluated and compared with the classical SCA. Overall comparison of the results demonstrates the better performance of the ISCA as compared to the classical SCA. Later on in this chapter, the comparison between the ISCA and the m-SCA (presented in Chapter 4) is also performed based on various criteria such as minimum, median, mean, maximum and standard deviation of absolute error values in objective function, statistical analysis through Wilcoxon signed rank test and convergence behavior analysis.

From the detailed comparative analysis the following conclusions can be made:

1. In terms of worst time complexity calculated through big$-O$ notation, all the algorithms classical SCA, m-SCA and ISCA are same.
2. On unimodal problems, the ISCA performs better than the classical SCA and m-SCA.
3. On multimodal and hybrid problems, the ISCA is more successful as compared to the classical SCA and m-SCA.
4. On the set of composite problems, both the algorithms m-SCA and ISCA are very competitive to each other and outperform classical SCA.
5. The convergence rate is better in ISCA as compared to classical SCA and m-SCA.

Thus, in the Thesis, four modified variants are proposed namely RW-GWO (presented in Chapter 2), mGWO (presented in Chapter 3), m-SCA (presented in Chapter 4) and ISCA (presented in Chapter 5). The comparison among these variants through statistical test can be performed to choose the best performer for different category of optimization problems. Hence, the Wilcoxon signed rank test is conducted between the classical GWO, classical SCA and their proposed variants on IEEE CEC 2014 benchmark problems and the obtained conclusions are reported in Table 9.1. In this table, the comparison of two algorithms A and B (represented as A/B in the table) is shown in the format 'a/b/c' where 'a' represents the number of problems in which the first algorithm (A) performs better than the second algorithm (B). Similarly, 'b' represents the number of problems in which the second algorithm (B) performs better than the first algorithm (A). 'c' represents the number of problems in which both the algorithms A and B are statistically same.

The statistical comparison concludes that the RW-GWO algorithm performed better than all other algorithms for unimodal and composite benchmark problems and for multimodal and hybrid problems, the mGWO algorithm outperformed other variants of GWO and SCA and their classical versions.

Table 9.1. Comparison between classical GWO, classical SCA and their proposed variants

Comparison	Dimension	unimodal	multimodal	hybrid	composite
RW-GWO/GWO	**10**	2/0/1	10/0/3	5/1/0	5/0/3
	30	3/0/0	12/0/1	4/0/2	7/1/0
mGWO/GWO	**10**	2/0/1	13/0/0	6/0/0	7/0/1
	30	3/0/0	13/0/0	6/0/0	7/1/0
RW-GWO/mGWO	**10**	1/1/1	2/8/3	1/5/0	3/2/3
	30	2/1/0	4/7/2	0/5/1	5/2/1
m-SCA/SCA	**10**	3/0/0	13/0/0	6/0/0	8/0/0
	30	3/0/0	13/0/0	6/0/0	8/0/0
ISCA/SCA	**10**	3/0/0	13/0/0	6/0/0	8/0/0
	30	3/0/0	13/0/0	6/0/0	8/0/0
m-SCA/ISCA	**10**	0/3/0	1/11/1	1/4/1	7/1/0
	30	1/2/0	2/11/0	1/5/0	2/6/0
GWO/SCA	**10**	3/0/0	12/0/1	5/0/1	6/0/2
	30	3/0/0	11/0/2	6/0/0	8/0/0
GWO/m-SCA	**10**	1/1/1	3/6/4	0/2/4	2/6/0
	30	0/2/1	6/4/3	0/2/4	2/4/2
GWO/ISCA	**10**	1/2/0	0/12/1	3/2/1	2/6/0
	30	0/3/0	0/10/3	0/4/2	0/7/1
RW-GWO/SCA	**10**	3/0/0	13/0/0	6/0/0	7/0/1
	30	3/0/0	13/0/0	6/0/0	8/0/0
mGWO/SCA	**10**	3/0/0	13/0/0	6/0/0	7/0/1
	30	3/0/0	13/0/0	6/0/0	8/0/0
RW-GWO/m-SCA	**10**	3/0/0	10/1/2	3/2/1	4/4/0
	30	3/0/0	12/1/0	4/1/1	7/1/0
RW-GWO/ISCA	**10**	3/0/0	6/7/0	2/2/2	5/3/0
	30	3/0/0	8/3/2	1/3/2	7/1/0
mGWO/m-SCA	**10**	3/0/0	11/1/1	6/0/0	4/3/1
	30	3/0/0	13/0/0	6/0/0	7/1/0
mGWO/ISCA	**10**	3/0/0	11/1/1	4/2/0	4/1/3
	30	3/0/0	11/0/2	5/0/1	7/0/1

The performance ordering based on the above performed analysis corresponding to different category of problems is as follows:

Unimodal: RW-GWO>mGWO>ISCA>m-SCA>GWO>SCA

Multimodal: mGWO>RW-GWO>ISCA>m-SCA>GWO>SCA

Hybrid: mGWO>ISCA>RW-GWO>m-SCA>GWO>SCA

Composite: RW-GWO>mGWO>m-SCA>ISCA>GWO>SCA

Chapter 6 evaluates the performances of classical GWO, RW-GWO, mGWO, classical SCA, m-SCA and ISCA on constrained benchmark problems given in IEEE CEC 2006. In this benchmark set, a simple constraint handling technique based on constraint violation is used to handle the constraints. The comparison between the algorithms is performed through various criteria such as minimum, median, mean, maximum and standard deviation of objective function values. The statistical analysis between the results is conducted with the help of Wilcoxon rank sum test which demonstrates the better search ability of the mGWO as compared to classical GWO and RW-GWO. Similarly, the ISCA performs significantly better than classical SCA and m-SCA as a constrained optimizer. It is concluded from the results that the proposed mGWO algorithm is better constrained optimizer than classical GWO, RW-GWO, classical SCA, m-SCA and ISCA. The performance ordering for the constrained problems based on statistical test and the number of problems on which algorithms provides feasible solution is as follows:

mGWO> RW-GWO> GWO> ISCA> m-SCA> SCA

Chapter 7 implements the classical GWO, RW-GWO, mGWO, classical SCA, m-SCA and ISCA to solve an unconstrained nonlinear optimization problem arising in the field of image processing. The objective of this problem is to determine the optimal thresholds for image segmentation in grey images. Nine benchmark images are used for experimentation and several statistical measures are used for the comparison. The analysis of results ensures that the proposed variants RW-GWO and mGWO of classical GWO perform better than classical GWO, classical SCA, m-SCA and ISCA. The mean and median value of the objective function defined by the Otsu method demonstrate the better efficiency and reliability of the proposed RW-GWO algorithm as compared to other proposed variants of GWO and SCA and their classical versions. It is concluded from all the results and analysis that the proposed RW-GWO is more suitable to find the optimal thresholds for grey images as compared to the classical GWO, mGWO, classical SCA, m-SCA and ISCA.

Chapter 8 employs the classical GWO, RW-GWO, mGWO, classical SCA, m-SCA and ISCA to solve a nonlinear constrained optimization problem arising in the field of electrical engineering. The objective of this problem is to determine the optimal setting for the proper coordination of overcurrent relays. The IEEE 3, 4, 6, and 14-bus systems are used to evaluate the performance of these algorithms on relay coordination problem. The comparative analysis of the results on these bus systems demonstrates the better search efficiency of the RW-GWO algorithm as compared to other proposed variants and classical versions of GWO and SCA to solve the relay coordination problem.

Chapter 9 depicts the overall conclusions of each chapter about the performance of proposed variants of classical GWO and classical SCA. It also outlines the limitations and scope of the proposed variants of GWO and SCA and suggests the best performer algorithm from these variants to solve global optimization problems. Finally, the chapter is closed with some future research directions.

9.2. Future Research

Research is an iterative and continuous procedure. The work presented in the Thesis is not an exception. There could be several research directions in which this could be expanded.

- Instead of applying the cauchy random walk for the leading hunters in GWO, other local and global search methods can also be employed to provide a better guidance for the wolf pack.
- The other constraint handling mechanisms can be merged into proposed modified variants of GWO and SCA to solve the constraint optimization problems.
- The proposed variants can also be implemented on large-scale optimization problems.
- The multi-objective versions of the proposed algorithms in the Thesis can also be developed to deal the multi-criteria optimization.
- The proposed variants can be applied to solve other real-life application problems.

Appendices

Appendix A

Unconstrained Test Problems

These problems are taken from (Liang, Qu and Suganthan, 2013).

Table A1. The brief description of IEEE CEC 2014 benchmark functions

No.	Type	Name	Optimum
F1	Unimodal Functions	Rotated high conditioned elliptic function	100
F2		Rotated bent cigar function	200
F3		Rotated discus function	300
F4	Simple Multimodal Functions	Shifted and rotated Rosenbrock's function	400
F5		Shifted and rotated Ackley's function	500
F6		Shifted and rotated Weierstrass function	600
F7		Shifted and rotated Griewank's function	700
F8		Shifted Rastrigin's function	800
F9		Six Hump Camel Back	900
F10		Shifted and rotated Rastrigin's Function	1000
F11		Shifted and rotated Schwefel's Function	1100
F12		Shifted and rotated Katsuura Function	1200
F13		Shifted and rotated HappyCat Function	1300
F14		Shifted and rotated HGBat Function	1400
F15		Shifted and rotated Expanded Griewank's plus Rosenbrock's Function	1500
F16		Shifted and rotated Expanded Scaffer's F6 Function	1600
F17	Hybrid Functions	Hybrid function 1 (N = 3)	1700
F18		Hybrid function 2 (N = 3)	1800
F19		Hybrid function 3 (N = 4)	1900
F20		Hybrid function 4 (N = 4)	2000
F21		Hybrid function 5 (N = 5)	2100
F22		Hybrid function 6 (N = 5)	2200
F23	Composite Functions	Composition function 1 (N = 5)	2300
F24		Composition function 1 (N = 3)	2400
F25		Composition function 1 (N = 3)	2500
F26		Composition function 1 (N = 5)	2600
F27		Composition function 1 (N = 5)	2700
F28		Composition function 1 (N = 5)	2800
F29		Composition function 1 (N = 3)	2900
F30		Composition function 1 (N = 3)	3000
Search Range: $[-100, 100]^D$			

A1. Definitions of the Basic Functions

A1.1. High Conditioned Elliptic Function

$$f_1(x) = \sum_{i=1}^{D} (10^6)^{\left(\frac{i-1}{D-1}\right)} x_i^2$$

A1.2. Bent Cigar Function

$$f_2(x) = x_1^2 + 10^6 \sum_{i=2}^{D} x_i^2$$

A1.3. Discus Function

$$f_3(x) = 10^6 x_1^2 + \sum_{i=2}^{D} x_i^2$$

A1.4. Rosenbrock's Function

$$f_4(x) = \sum_{i=1}^{D-1} (100(x_i^2 - x_{i+1})^2 + (x_i - 1)^2)$$

A1.5. Ackley's Function

$$f_5(x) = -20exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - exp\left(\frac{1}{D}\sum_{i=}^{D} \cos(2\pi x_i\right) + 20 + e$$

A1.6. Weierstrass Function

$$f_6(x) = \sum_{i=1}^{D}\left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_i + 0.5))]\right) - D\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)]$$

$$a = 0.5, b = 3, k_{\max} = 20$$

A1.7. Griewank's Function

$$f_7(x) = \sum_{i=1}^{D} \frac{x_i^2}{4000} - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$$

A1.8. Rastrigin's Function

$$f_8(x) = \sum_{i=1}^{D} (x_i^2 - 10\cos(2\pi x_i) + 10)$$

A1.9. Modified Schwefel's Function

$$f_9(x) = 418.9829 \times D - \sum_{i=1}^{D} g(z_i), \quad z_i = x_i + 4.209687462275036e + 002$$

$$g(z_i) = \begin{cases} z_i \sin\left(|z_i|^{\frac{1}{2}}\right) & if\ |z_i| \le 500 \\ (500 - mod(z_i, 500))sin\left(\sqrt{|500 - mod(z_i, 500)|}\right) - \frac{(z_i - 500)^2}{10^4 D} & if\ z_i > 500 \\ (mod(|z_i|, 500) - 500)sin\left(\sqrt{|mod(|z_i|, 500) - 500|}\right) - \frac{(z_i + 500)^2}{10^4 D} & if\ z_i < -500 \end{cases}$$

A1.10. Katsuura Function

$$f_{10}(x) = \frac{10}{D^2} \prod_{i=1}^{D} \left(1 + i \sum_{j=1}^{32} \frac{|2^j x_i - round(2^j x_i|}{2^j}\right)^{\frac{10}{D^{1.2}}} - \frac{10}{D^2}$$

A1.11. HappyCat Function

$$f_{11}(x) = \left|\sum_{i=1}^{D} x_i^2 - D\right|^{1.4} + \frac{(0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)}{D} + 0.5$$

A1.12. HGBat Function

$$f_{12}(x) = \left|\left(\sum_{i=1}^{D} x_i^2\right)^2 - \left(\sum_{i=1}^{D} x_i\right)^2\right|^{1/2} + \frac{(0.5 \sum_{i=1}^{D} x_i^2 + \sum_{i=1}^{D} x_i)}{D} + 0.5$$

A1.13. Expanded Griewank's plus Rosenbrock's Function

$$f_{13}(x) = f_7(f_4(x_1, x_2)) + f_7(f_4(x_2, x_3)) + \cdots + f_7(f_4(x_{D-1}, x_D)) + f_7(f_4(x_D, x_1))$$

A1.14. Expanded Scaffer's F6 Function

Scaffer's F6 Function: $g(x, y) = 0.5 + \frac{\left(\sin^2\left(\sqrt{x^2+y^2}\right) - 0.5\right)}{\left(1 + 0.001(x^2+y^2)\right)^2}$

$$f_{14}(x) = g(x_1, x_2) + g(x_2, x_3) + \cdots + g(x_{D-1}, x_D) + g(x_D, x_1)$$

Appendix B

Constrained Test Problems

These problems are taken from (Liang et al., 2006).

Problem g01

Minimize $f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=5}^{13} x_i$

$$
\begin{aligned}
g_1(x) &= (2x_1 + 2x_2 + x_{10} + x_{11}) - 10 \le 0,\\
g_2(x) &= (2x_1 + 2x_3 + x_{10} + x_{12}) - 10 \le 0,\\
g_3(x) &= (2x_2 + 2x_3 + x_{11} + x_{12}) - 10 \le 0,\\
g_4(x) &= -8x_1 + x_{10} \le 0,\\
g_5(x) &= -8x_2 + x_{11} \le 0,\\
g_6(x) &= -8x_3 + x_{12} \le 0,\\
g_7(x) &= -2x_4 - x_5 + x_{10} \le 0,\\
g_8(x) &= -2x_6 - x_7 + x_{11} \le 0,\\
g_9(x) &= -2x_8 - x_9 + x_{12} \le 0,\\
&0 \le x_i \le 1,\ i = 1,\cdots,9,\\
&0 \le x_i \le 100,\ i = 10,11,12,\\
&0 \le x_{13} \le 1.
\end{aligned}
$$

This problem has global minima at $x^* = (1,1,\cdots\cdots 1,3,3,3,1)$ with $f_{min} = -15$.

Problem g02

Minimize $f(x) = -\left|\dfrac{\sum_{i=1}^{n}\cos^4(x_i) - 2\prod_{i=1}^{n}\cos^2(x_i)}{\sqrt{\sum_{i=1}^{n} i x_i^2}}\right|$

Subject to:

$$g_1(x) = 0.75 - \prod_{i=1}^{n} x_i \le 0$$

$$g_2(x) = \sum_{i=1}^{n} x_i - 7.5n \le 0$$

where $n = 20$ and $0 \le x_i \le 10$ $(i = 1,\ldots\ldots,n)$. The global minimum $x^* =$

(3.16246061572185, 3.12833142812967, 3.09479212988791, 3.06145059523469,
3.02792915885555, 2.99382606701730, 2.95866871765285, 2.92184227312450,
0.49482511456933, 0.48835711005490, 0.48231642711865, 0.47664475092742,
0.47129550835493, 0.46623099264167, 0.46142004984199, 0.45683664767217,
0.45245876903267, 0.44826762241853, 0.44424700958760, 0.44038285956317)

with $f_{min} = 0.80361910412559$.

Problem g03

Minimize $f(x) = -(\sqrt{n})^n \prod_{i=1}^{n} x_i$

Subject to: $h_1(x) = \sum_{i=1}^{n} x_i^2 - 1 = 0$

where $n = 10$ and $0 \le x_i \le 1$ $(i = 1, 2, . . . , n)$, for each variable.

The global minimum is at $x^* =$

(0.31624357647283069, 0.316243577414338339, 0.316243578012345927, 0.316243575664017895, 0.316243578205526066, 0.31624357738855069, 0.316243575472949512, 0.316243577164883938, 0.316243578155920302, 0.316243576147374916) with $f_{min} = -1.00050010001$.

Problem g04

Minimize $f(x) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141$

Subject to:

$$g_1(x) = 85.334407 + 0.0056858x_2x_5 + 0.0006262x_1x_4 - 0.0022053x_3x_5 - 92 \le 0,$$
$$g_2(x) = -85.334407 - 0.0056858x_2x_5 - 0.0006262x_1x_4 + 0.0022053x_3x_5 \le 0,$$
$$g_3(x) = 80.51249 + 0.0071317x_2x_5 + 0.0029955x_1x_2 + 0.0021813{x^2}_3 - 110 \le 0,$$
$$g_4(x) = -80.51249 - 0.0071317x_2x_5 - 0.0029955x_1x_2 - 0.0021813{x^2}_3 + 90 \le 0,$$
$$g_5(x) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_3 + 0.0019085x_3x_4 - 25 \le 0,$$
$$g_6(x) = -9.300961 - 0.0047026x_3x_5 - 0.0012547x_1x_3 - 0.0019085x_3x_4 + 20 \le 0,$$

where $78 \le x_1 \le 102$, $33 \le x_2 \le 45$ and $27 \le x_i \le 45$ $(i = 3, 4, 5)$. The optimum solution is

$x^* = (78, 33, 29.995256025682, 45, 36.775812905788)$ where $f_{min} = -30665.539$.

Problem 05

Minimize $f(x) = 3x_1 + 0.000001x_1^3 + 2x_2 + (0.000002/3)x_2^3$

Subject to:

$$g_1(x) = -x_4 + x_3 - 0.55 \le 0,$$
$$g_2(x) = -x_3 + x_4 - 0.55 \le 0,$$
$$h_3(x) = 1000\, sin(-x_3 - 0.25) + 1000\, sin(-x_4 - 0.25) + 894.8 - x_1 = 0,$$
$$h_4(x) = 1000\, sin(x_3 - 0.25) + 1000\, sin(x_3 - x_4 - 0.25) + 894.8 - x_2 = 0,$$
$$h_5(x) = 1000\, sin(x_4 - 0.25) + 1000\, sin(x_4 - x_3 - 0.25) + 1294.8 = 0,$$

where $0 \le x_1 \le 1200$, $0 \le x_2 \le 1200, -0.55 \le x_3 \le 0.55$ and $-0.55 \le x_4 \le 0.55$. The best known solution $x^* = (679.9453, 1026.067, 0.1188764, -0.3962336)$ where $f_{min} = 5126.49671$.

Problem g06

Minimize $f(x) = (x_1 - 10)^3 + (x_2 - 20)^3$

Subject to:

$$g_1(x) = -(x_1 - 5)^2 - (x_2 - 5)^2 + 100 \le 0,$$
$$g_2(x) = (x_1 - 6)^2 + (x_2 - 5)^2 - 82.81 \le 0,$$

where $13 \le x_1 \le 100$ and $0 \le x_2 \le 100.$ The optimum solution is $x^* = (14.095, 0.84296)$ where $f_{min} = -6961.81388$.

Problem g07

Minimize
$$f(x) = x_1^2 + x_2^2 + x_1x_2 - 14x_1 - 16x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

Subject to:

$$g_1(x) = -105 + 4x_1 + 5x_2 - 3x_7 + 9x_8 \le 0,$$
$$g_2(x) = 10x_1 - 8x_2 - 17x_7 + 2x_8 \le 0,$$
$$g_3(x) = -8x_1 + 2x_2 + 5x_9 - 2x_{10} - 12 \le 0,$$
$$g_4(x) = 3(x_1 - 2)^2 + 4(x_2 - 3)^2 + 2x^2{}_3 - 7x_4 - 120 \le 0,$$
$$g_5(x) = 5x^2{}_1 + 8x_2 + (x_3 - 6)^2 - 2x_4 - 40 \le 0,$$
$$g_6(x) = x^2{}_1 + 2(x_2 - 2)^2 - 2x_1x_2 + 14x_5 - 6x_6 \le 0,$$
$$g_7(x) = 0.5(x_1 - 8)^2 + 2(x_2 - 4)^2 + 3x^2{}_5 - x_6 - 30 \le 0,$$
$$g_8(x) = -3x_1 + 6x_2 + 12(x_9 - 8)^2 - 7x_{10} \le 0,$$

where $-10 \le x_i \le 10 \ (i = 1, \ldots, 10)$. The optimum solution is

$x^* = (2.171996, 2.363683, 8.773926, 5.095984, 0.9906548, 1.430574, 1.321644,$

$9.828726, 8.280092, 8.375927)$ where $f_{min} = 24.3062091$.

Problem g08

Minimize $f(x) = -\dfrac{\sin^3(2\pi x_1)\sin(2\pi x_2)}{x_1^3(x_1 + x_2)}$

subject to:

$$g_1(x) = x^2{}_1 - x_2 + 1 \le 0,$$
$$g_2(x) = 1 - x_1 + (x_2 - 4)^2 \le 0,$$

where $0 \le x_1 \le 10$ and $0 \le x_2 \le 10.$ The optimum is located at

$x^* = (1.22797135260752599, 4.24537336612274885)$ where $f_{min} = -0.0958250414180359.$

Problem g09

Minimize $f(x) = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7$

Subject to:

$$g_1(x) = -127 + 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 \le 0,$$
$$g_2(x) = -282 + 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 \le 0,$$
$$g_3(x) = -196 + 23x_1 + x_2^2 + 6x_6^2 - 8x_7 \le 0,$$
$$g_4(x) = 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7 \le 0,$$

where $-10 \le x_i \le 10$ for $(i = 1, \ldots, 7)$. The optimum solution is

$x^* = (2.330499351474051 74, 1.951372368471145 92, -0.477541399510615805,$

$4.36572624923625874, -0.624486959100388983, 1.03813099410962173, 1.5942266780671519)$

with $f_{min} = 680.630057374402$.

Problem g10

Minimize $f(x) = x_1 + x_2 + x_3$

Subject to:

$$g_1(x) = -1 + 0.0025(x_4 + x_6) \le 0,$$
$$g_2(x) = -1 + 0.0025(x_5 + x_7 - x_4) \le 0,$$
$$g_3(x) = -1 + 0.01(x_8 - x_5) \le 0,$$
$$g_4(x) = -x_1x_6 + 833.33252x_4 + 100x_1 - 83333.333 \le 0,$$
$$g_5(x) = -x_2x_7 + 1250x_5 + x_2x_4 - 1250x_4 \le 0,$$
$$g_6(x) = -x_3x_8 + 1250000\, x_3x_5 - 2500x_5 \le 0,$$

where $100 \le x_1 \le 10000$, $1000 \le x_i \le 10000$ $(i = 2,3)$ and $10 \le x_i \le 1000$ $(i = 4, \ldots, 8)$. The optimum solution is at

$x^* = (579.306685017979589, 1359.97067807935605, 5109.97065743133317,$

$182.01769963061534, 295.601173702746792, 217.982300369384632, 286.41652592786852,$
$395.601173702746735)$ with $f_{min} = 7049.24802052867$.

Problem g11

Minimize $f(x) = x_1^2 + (x_2 - 1)^2$

Subject to:

$$h_1(x) = x_2 - x_1^2 = 0,$$

The bounds on the variables are: $-1 \le x_i \le 1,\ i = 1,2$. This problem has global minima at

$x^* = (-0.707036070037170616, 0.500000004333606807)$ with $f_{min} = 0.7499$.

Problem g12

Minimize $f(x) = -(100-(x_1-5)^2-(x_2-5)^2-(x_3-5)^2)/100$

Subject to:

$$g(x) = (x_1-p)^2+(x_2-q)^2+(x_3-r)^2-0.0625 \le 0,$$

where $0 \le x_i \le 10 (i = 1,2,3)$ and $p,q,r = 1,2,...,9$. The feasible region of the search space consists of 9^3 disjoint spheres. A point (x_1, x_2, x_3) is feasible if and only if there exist p,q,r such that the above inequality holds. The optimum is located at $x^* = (5,5,5)$ where $f_{min} = -1$. The solution lies within the feasible region.

Problem g13

Minimize $f(x) = e^{(x_1x_2x_3x_4x_5)}$

Subject to:

$$h_1(x) = x_1^2+x_2^2+x_3^2+x_4^2+x_5^2-10=0,$$
$$h_2(x) = x_2x_3-5x_4x_5=0,$$
$$h_3(x) = x_1^3+x_2^3+1=0,$$

where $-2.3 \le x_i \le 2.3$ $(i = 1,2)$ and $-3.2 \le x_i \le 3.2$ $(i = 3,4,5)$. The optimum solution is

$x^* = (-1.71714224003, 1.59572124049468, 1.8272502406271,$

$-0.763659881912867, -0.76365986736498)$ with $f_{min} = 0.053941514041898.$

Problem g14

Minimize $f(x) = \sum_{i=1}^{10} x_i \left(c_i + \ln \frac{x_i}{\sum_{j=1}^{10} x_j} \right)$

Subject to:

$$h_1(x) = x_1+2x_2+2x_3+x_6+x_{10}-2=0,$$
$$h_2(x) = x_4+2x_5+x_6+x_7-1=0,$$
$$h_3(x) = x_3+x_7+x_8+2x_9+x_{10}-1=0,$$

where the bounds are $0 \le x_i \le 10$ $(i = 1, ..., 10)$ and $c_1 = -6.089, c_2 = -17.164,$

$c_3 = -34.054, c_4 = -5.914, c_5 = -24.721, c_6 = -14.986, c_7 = -24.1, c_8 = -10.708, c_9 = -26.662,$

$c_{10} = -22.179$. The best known solution is at $x^* = (0.0406684113216282,$

$0.147721240492452, 0.783205732104114, 0.00141433931889084, 0.485293636780388,$

$0.000693183051556082, 0.0274052040687766, 0.0179509660214818, 0.0373268186859717,$

$0.0968844604336845)$ with $f_{min} = -47.7648884594915.$

Problem g15

Minimize $f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$

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 the bounds are $0 \le x_i \le 10$ $(i = 1, 2, 3)$. The best known solution is at

$x^* = (3.51212812611795133,\ 0.216987510429556135,\ 3.55217854929179921)$

with $f_{min} = 961.715022289961$.

Problem g16

Minimize
$$f(x) = -0.0000005843y_{17} + 0.000117y_{14} + 0.1365 + 0.00002358y_{13} + 0.000001502y_{16} + 0.0321y_{12} + 0.004324y_5 + \frac{c_{15}}{c_{16}} + 37.48\frac{y_2}{c_{12}}$$

Subject to:

$$g_1(x) = \frac{0.28}{0.72}y_5 - y_4 \le 0,$$
$$g_2(x) = x_3 - 1.5x_2 \le 0,$$
$$g_3(x) = 3496\ \frac{y_2}{c_{12}} - 21 \le 0$$
$$g_4(x) = 110.6 + y_1 - \frac{62212}{c_{17}} \le 0$$
$$g_5(x) = y_1 - 213.1 \ge 0,$$
$$g_6(x) = 405.23 - y_1 \ge 0,$$
$$g_7(x) = y_2 - 17.505 \ge 0,$$
$$g_8(x) = 1053.6667 - y_2 \ge 0,$$
$$g_9(x) = y_3 - 11.275 \ge 0,$$
$$g_{10}(x) = 35.03 - y_3 \ge 0,$$
$$g_{11}(x) = y_4 - 214.228 \ge 0,$$
$$g_{12}(x) = 665.585 - y_4 \ge 0,$$
$$g_{13}(x) = y_5 - 7.458 \ge 0,$$
$$g_{14}(x) = 584.463 - y_5 \ge 0,$$
$$g_{15}(x) = y_6 - 0.961 \ge 0,$$
$$g_{16}(x) = 265.916 - y_6 \ge 0,$$
$$g_{17}(x) = y_7 - 1.612 \ge 0,$$
$$g_{18}(x) = 7.046 - y_7 \ge 0,$$

$$g_{19}(x) = y_8 - 0.146 \geq 0,$$
$$g_{20}(x) = 0.222 - y_8 \geq 0,$$
$$g_{21}(x) = y_9 - 107.99 \geq 0,$$
$$g_{22}(x) = 273.366 - y_9 \geq 0,$$
$$g_{23}(x) = y_{10} - 922.693 \geq 0,$$
$$g_{24}(x) = 1286.105 - y_{10} \geq 0,$$
$$g_{25}(x) = y_{11} - 926.832 \geq 0,$$
$$g_{26}(x) = 1444.046 - y_{11} \geq 0,$$
$$g_{27}(x) = y_{12} - 18.766 \geq 0,$$
$$g_{28}(x) = 537.141 - y_{12} \geq 0,$$
$$g_{29}(x) = y_{13} - 1072.163 \geq 0,$$
$$g_{30}(x) = 3247.039 - y_{13} \geq 0,$$
$$g_{31}(x) = y_{14} - 8961.448 \geq 0,$$
$$g_{32}(x) = 26844.086 - y_{14} \geq 0,$$
$$g_{33}(x) = y_{15} - 0.063 \geq 0,$$
$$g_{34}(x) = 0.386 - y_{15} \geq 0,$$
$$g_{35}(x) = y_{16} - 71084.33 \geq 0,$$
$$g_{36}(x) = 140000 - y_{16} \geq 0,$$
$$g_{37}(x) = y_{17} - 2802713 \geq 0,$$
$$g_{38}(x) = 12146108 - y_{17} \geq 0,$$

where:

$$y_1 = x_2 + x_3 + 41.6,$$
$$c_1 = 0.024x_4 - 4.62,$$
$$y_2 = \frac{12.5}{c_1} + 12,$$
$$c_2 = 0.0003535x^2{}_1 + 0.5311x_1 + 0.08705y_2x_1,$$
$$c_3 = 0.052x_1 + 78 + 0.002377y_2x_1,$$
$$y_3 = \frac{c_2}{c_3},$$
$$y_4 = 19y_3,$$
$$c_4 = 0.04782(x_1 - y_3) + \frac{0.1956(x_1 - y_3)^2}{x_2} + 0.6376y_4 + 1.594y_3,$$
$$c_5 = 100x_2,$$
$$c_6 = x_1 - y_3 - y_4,$$
$$c_7 = 0.950 - \frac{c_4}{c_5},$$
$$y_5 = c_6c_7,$$
$$y_6 = x_1 - y_5 - y_4 - y_3,$$
$$c_8 = (y_5 + y_4)0.995,$$
$$y_7 = \frac{c_8}{y_1},$$
$$y_8 = \frac{c_8}{3798},$$

$$c_9 = y_7 - \frac{0.0663 y_7}{y_8} - 0.3153,$$
$$y_9 = \frac{96.82}{c_9} + 0.321 y_1,$$
$$y_{10} = 1.29 y_5 + 1.258 y_4 + 2.29 y_3 + 1.71 y_6,$$
$$y_{11} = 1.71 x_1 - 0.452 y_4 + 0.580 y_3,$$
$$c_{10} = \frac{12.3}{752.3}$$
$$c_{11} = (1.75 y_2)(0.995 x_1),$$
$$c_{12} = 0.995 y_{10} + 1998,$$
$$y_{12} = c_{10} x_1 + \frac{c_{11}}{c_{12}},$$
$$y_{13} = c_{12} - 1.75 y_2,$$
$$y_{14} = 3623 + 64.4 x_2 + 58.4 x_3 + \frac{143612}{y_9 + x_5},$$
$$c_{13} = 0.995 y_{10} + 60.8 x_2 + 48 x_4 - 0.1121 y_{14} - 5095,$$
$$y_{15} = \frac{y_{13}}{c_{13}},$$
$$y_{16} = 148000 - 331000 y_{15} + 40 y_{13} - 61 y_{15} y_{13},$$
$$c_{14} = 2324 y_{10} - 28740000 y_2,$$
$$y_{17} = 14130000 - 1328 y_{10} - 531 y_{11} + \frac{c_{14}}{c_{12}},$$
$$c_{15} = \frac{y_{13}}{y_{15}} - \frac{y_{13}}{0.52},$$
$$c_{16} = 1.104 - 0.72 y_{15},$$
$$c_{17} = y_9 + x_5,$$

and where the bounds are $704.4148 \le x_1 \le 906.3855$, $68.6 \le x_2 \le 288.88$, $0 \le x_3 \le 134.75$, $193 \le x_4 \le 287.0966$ and $25 \le x_5 \le 84.1988$. The best known solution is at $x^* = (705.174537070090537, 68.5999999999999943, 102.899999999999991, 282.324931593660324, 37.5841164258054832)$ with $f_{min} = -1.90515525853479$.

Problem g17

Minimize $f(x) = f(x_1) + f(x_2)$

where

$$f_1(x_1) = \begin{cases} 30x_1 & 0 \le x_1 < 300 \\ 31x_1 & 300 \le x_1 < 400 \end{cases}$$

$$f_2(x_2) = \begin{cases} 28x_2 & 0 \le x_2 < 100 \\ 29x_2 & 100 \le x_2 < 200 \\ 30x_2 & 200 \le x_2 < 1000 \end{cases}$$

Subject to:

$$h_1(x) = -x_1 + 300 - \frac{x_3 x_4}{131.078}\cos(1.48477 - x_6) + \frac{0.90798 x_3^2}{131.078}\cos(1.47588),$$
$$h_2(x) = -x_2 - \frac{x_3 x_4}{131.078}\cos(1.48477 + x_6) + \frac{0.90798 x_4^2}{131.078}\cos(1.47588),$$
$$h_3(x) = -x_5 - \frac{x_3 x_4}{131.078}\sin(1.48477 + x_6) + \frac{0.90798 x_4^2}{131.078}\sin(1.47588),$$
$$h_4(x) = 200 - \frac{x_3 x_4}{131.078}\sin(1.48477 - x_6) + \frac{0.90798 x_3^2}{131.078}\sin(1.47588),$$

where the bounds are $0 \le x_1 \le 400$, $0 \le x_2 \le 1000$, $340 \le x_3 \le 420$, $340 \le x_4 \le 420$, $-1000 \le x_5 \le 1000$ and $0 \le x_6 \le 0.5236$. The best known solution is at

$x^* = (201.784467214523659,\ 99.\ 9999999999999005,\ 383.071034852773266, 420,$

$-10.9076584514292652,\ 0.0731482312084287128)$ where $f(x) = 8853.53967480648$

Problem g18

Minimize $f(x) = -0.5(x_1x_4 - x_2x_3 + x_3x_9 - x_5x_9 + x_5x_8 - x_6x_7)$

Subject to:

$$g_1(x) = x^2{}_3 + x^2{}_4 - 1 \le 0,$$
$$g_2(x) = x^2{}_9 - 1 \le 0,$$
$$g_3(x) = x^2{}_5 + x^2{}_6 - 1 \le 0,$$
$$g_4(x) = x^2{}_1 + (x_2 - x_9)^2 - 1 \le 0,$$
$$g_5(x) = (x_1 - x_5)^2 + (x_2 - x_6)^2 - 1 \le 0,$$
$$g_6(x) = (x_1 - x_7)^2 + (x_2 - x_8)^2 - 1 \le 0,$$
$$g_7(x) = (x_3 - x_5)^2 + (x_4 - x_6)^2 - 1 \le 0,$$
$$g_8(x) = (x_3 - x_7)^2 + (x_4 - x_8)^2 - 1 \le 0,$$
$$g_9(x) = x^2{}_7 + (x_8 - x_9)^2 - 1 \le 0,$$
$$g_{10}(x) = x_2x_3 - x_1x_4 \le 0,$$
$$g_{11}(x) = -x_3x_9 \le 0,$$
$$g_{12}(x) = x_5x_9 \le 0,$$
$$g_{13}(x) = x_6x_7 - x_5x_8 \le 0,$$

where the bounds are $-10 \le x_i \le 10 (i = 1,\ldots,8)$ and $0 \le x_9 \le 20$. The best known -solution is at

$x^* = (-0.657776192427943163, -0.153418773482438542,\ 0.323413871675240938,$

$-0.946257611651304398, -0.657776194376798906, -0.753213434632691414,$

$0.323413874123576972, -0.346462947962331735, 0.599794662852175 42)$

where $f_{min} = -0.866025403784439.$

Problem g19

Minimize $f(x) = \sum_{j=1}^{5}\sum_{i=1}^{5} c_{ij} x_{(10+i)} x_{(10+j)} + 2\sum_{j=1}^{5} d_j x_{(10+j)}^3 - \sum_{i=1}^{10} b_i x_i$

Subject to:

$$g_j(x) = -2\sum_{i=1}^{5} c_{ij} x_{(10+i)} - 3d_j x_{(10+j)}^2 - e_j + \sum_{i=1}^{10} a_{ij} x_i \le 0, \quad j = 1,\ldots,5.$$

where $b = [-40, -2, -0.25, -4, -4, -1, -40, -60, 5, 1]$ and the remaining data is on Table B1. The bounds are $0 \le x_i \le 10 \ (i = 1, \ . \ . \ . \ , 15)$. The best known solution is at

$x^* = (1.66991341326291344e-17,\ 3.95378229282456509e-16,\ 3.94599045143233784,$
$1.06036597479721211e-16, 3.2831773458454161, 9.99999999999999822, 1.1282941467$
$1605333e-17, 1.2026194599794709e - 17,\ 2.50706276000769697e -15,\ 2.2462412298$ with
$7970677e-15,\ 0.370764847417013987, 0.278456024942955571, 0.523838487672241171,$
$0.388620152510322781,\ 0.298156764974678579)$

$f_{min} = 32.6555929502463.$

Table B1. Data set for test problem g19

j	1	2	3	4	5
e_j	-15	-27	-36	-18	-12
c_{1j}	30	-20	-10	32	-10
c_{2j}	-20	39	-6	-31	32
c_{3j}	-10	-6	10	-6	-10
c_{4j}	32	-31	-6	39	-20
c_{5j}	-10	32	-10	-20	30
d_j	4	8	10	6	2
a_{1j}	-16	2	0	1	0
a_{2j}	0	-2	0	0.4	2
a_{3j}	-3.5	0	2	0	0
a_{4j}	0	-2	0	-4	-1
a_{5j}	0	-9	-2	1	-2.8
a_{6j}	2	0	-4	0	0
a_{7j}	-1	-1	-1	-1	-1
a_{8j}	-1	-2	-3	-2	-1
a_{9j}	1	2	3	4	5
a_{10j}	1	1	1	1	1

Problem g20

Minimize $f(x) = \sum_{i=1}^{24} a_i x_i$

Subject to:

$$g_i(x) = \frac{(x_i + x_{(i+12)})}{\sum_{j=1}^{24} x_j + e_i} \le 0, \qquad i = 1,2,3$$
$$g_i(x) = \frac{(x_{(i+3)} + x_{(i+15)})}{\sum_{j=1}^{24} x_j + e_i} \le 0, \qquad i = 4,5,6$$
$$h_i(x) = \frac{x_{(i+12)}}{b_{(i+12)} \sum_{j=13}^{24} \frac{x_j}{b_j}} - \frac{c_i x_i}{40 b_i \sum_{j=1}^{12} \frac{x_j}{b_j}} = 0, \qquad i = 1,\ldots,12,$$
$$h_{13}(x) = \sum_{i=1}^{24} x_i - 1 = 0,$$
$$h_{14}(x) = \sum_{i=1}^{12} \frac{x_i}{d_i} + k \sum_{i=13}^{24} \frac{x_i}{b_i} - 1.671 = 0,$$

where $k = (0.7302)(530)\left(\frac{14.7}{40}\right)$ and the data set is detailed on Table B2. The bounds are $0 \le x_i \le 10 \ (i = 1, \ldots, 24)$. The best known solution is at

$x^* =$(1.28582343498528086 e-18, 4.83460302526130664 e-34, 0, 0, 6.30459929660781851 e-18, 7.57192526201145068 e-34, 5.03350698372840437 e-34, 9.28268079616618064 e-34, 0, 1.76723384525547359 e-17, 3.55686101822965701 e-34, 2.99413850083471346 e-34, 0.158143376337580827, 2.29601774161699833 e-19, 1.06106938611042947 e-18, 1.3196834431950 e-18, 0.530902525044209539, 0, 2.89148310257773535e - 18, 3.34892126180666159 e-18, 0, 0.310999974151577319, 5.41244666317833561 e-05, 4.84993165246959553 e-16).

This solution is a little infeasible and no feasible solution is found so far.

Problem g21

Minimize $f(x) = x_1$

Subject to:

$$g_1(x) = -x_1 + 35x_2^{0.6} + 35x_3^{0.6} \le 0,$$
$$h_1(x) = -300x_3 + 7500x_5 - 7500x_6 - 25x_4x_5 + 25x_4x_6 + x_3x_4 = 0,$$
$$h_2(x) = 100x_2 + 155.365x_4 + 2500x_7 - x_2x_4 - 25x_4x_7 - 15536.5 = 0,$$
$$h_3(x) = -x_5 + ln(-x_4 + 900) = 0,$$
$$h_4(x) = -x_6 + ln(x_4 + 300) = 0,$$
$$h_5(x) = -x_7 + ln(-2x_4 + 700) = 0,$$

where the bounds are $0 \le x_1 \le 1000$, $0 \le x_2, x_3 \le 40$, $100 \le x_4 \le 300$, $6.3 \le x_5 \le 6.7$, $5.9 \le x_6 \le 6.4$ and $4.5 \le x_7 \le 6.25$. The best known solution is at

$x^* =$(193.724510070034967, 5.56944131553368433e-27, 17.3191887294084914, 100.047897801386839, 6.68445185362377892, 5.99168428444264833, 6.21451648886070451)

where $f_{min} = 193.724510070035$.

Table B2. Data set for test problem g20

1	0.0693	44.094	123.7	31.244	0.1
2	0.0577	58.12	31.7	36.12	0.3
3	0.05	58.12	45.7	34.784	0.4
4	0.2	137.4	14.7	92.7	0.3
5	0.26	120.9	84.7	82.7	0.6
6	0.55	170.9	27.7	91.6	0.3
7	0.06	62.501	49.7	56.708	
8	0.1	84.94	7.1	82.7	
9	0.12	133.425	2.1	80.8	
10	0.18	82.507	17.7	64.517	
11	0.1	46.07	0.85	49.4	
12	0.09	60.097	0.64	49.1	
13	0.0693	44.094			
14	0.0577	58.12			
15	0.05	58.12			
16	0.2	137.4			
17	0.26	120.9			
18	0.55	170.9			
19	0.06	62.501			
20	0.1	84.94			
21	0.12	133.425			
22	0.18	82.507			
23	0.1	46.07			
24	0.09	60.097			

Problem g22

Minimize $f(x) = x_1$

Subject to:

$$
\begin{aligned}
g_1(x) &= -x_1 + x_2^{0.6} + x_3^{0.6} + x_4^{0.6} \le 0,\\
h_1(x) &= x_5 - 100000 x_8 + 1\times 10^7 = 0,\\
h_2(x) &= x_6 + 100000 x_8 - 100000 x_9 = 0,\\
h_3(x) &= x_7 + 100000 x_9 - 5\times 10^7 = 0,\\
h_4(x) &= x_5 + 100000 x_{10} - 3.3\times 10^7 = 0,\\
h_5(x) &= x_6 + 100000 x_{11} - 4.4\times 10^7 = 0,\\
h_6(x) &= x_7 + 100000 x_{12} - 6.6\times 10^7 = 0,\\
h_7(x) &= x_5 - 120 x_2 x_{13} = 0,\\
h_8(x) &= x_6 - 80 x_3 x_{14} = 0,\\
h_9(x) &= x_7 - 40 x_4 x_{15} = 0,\\
h_{10}(x) &= x_8 - x_{11} + x_{16} = 0,
\end{aligned}
$$

$h_{11}(x) = x_9 - x_{12} + x_{17} = 0,$
$h_{12}(x) = -x_{18} + ln(x_{10} - 100) = 0,$
$h_{13}(x) = -x_{19} + ln(-x_8 + 300) = 0,$
$h_{14}(x) = -x_{20} + ln(x_{16}) = 0,$
$h_{15}(x) = -x_{21} + ln(-x_9 + 400) = 0,$
$h_{16}(x) = -x_{22} + ln(x_{17}) = 0,$
$h_{17}(x) = -x_8 - x_{10} + x_{13}x_{18} - x_{13}x_{19} + 400 = 0,$
$h_{18}(x) = x_8 - x_9 - x_{11} + x_{14}x_{20} - x_{14}x_{21} + 400 = 0,$
$h_{19}(x) = x_9 - x_{12} - 4.60517x_{15} + x_{15}x_{22} + 100 = 0,$

where the bounds are $0 \le x_1 \le 20000,\ 0 \le x_2, x_3, x_4 \le 1\times10^6, 0 \le x_5, x_6, x_7 \le 4\times10^7,$ $100 \le x_8 \le 299.99, 100 \le x_9 \le 399.99, 100.01 \le x_{10} \le 300,\ 100 \le x_{11} \le 400, 100 \le x_{12} \le 600,$ $0 \le x_{13}, x_{14}, x_{15} \le 500,\ 0.01 \le x_{16} \le 300,\ 0.01 \le x_{17} \le 400,\ -4.7 \le x_{18},\ x_{19},\ x_{20},\ x_{21},\ x_{22} \le 6.25.$ The best known solution is at $x^* =$(236.430975504001054, 135.82847151732463, 204.818152544824585, 6446.54654059436416, 3007540.83940215595, 4074188.65771341929, 32918270.5028952882, 130.075408394314167, 170.817294970528621, 299.924591605478554, 399.258113423595205, 330.817294971142758, 184.51831230897065, 248.64670239647424, 127.658546694545862, 269.182627528746707, 160.000016724090955, 5.29788288102680571, 5.13529735903945728, 5.59531526444068827, 5.43444479314453499, 5.07517453535834395) where $f_{min} = 236.430975504001.$

Problem g23

Minimize $f(x) = -9x_5 - 15x_8 + 6x_1 + 16x_2 + 10(x_6 + x_7)$

Subject to:

$g_1(x) = x_9x_3 + 0.02x_6 - 0.025x_5 \le 0,$
$g_2(x) = x_9x_4 + 0.02x_7 - 0.015x_8 \le 0,$
$h_1(x) = x_1 + x_2 - x_3 - x_4 = 0,$
$h_2(x) = 0.03x_1 + 0.01x_2 - x_9(x_3 + x_4) = 0,$
$h_3(x) = x_3 + x_6 - x_5 = 0,$
$h_4(x) = x_4 + x_7 - x_8 = 0,$

where the bounds are $0 \le x_1,\ x_2,\ x_6 \le 300,\ 0 \le x_3,\ x_5,\ x_7 \le 100,\ 0 \le x_4,\ x_8 \le 200$ and $0.01 \le x_9 \le 0.03$. The best known solution is at $x^* =$(0.00510000000000259465, 99.9947000000000514, 9.01920162996045897e - 18, 99.9999000000000535, 0.000100000000027086086, 2.75700683389584542e -14, 99.9999999999999574, 2000.0100000100000100008) where $f_{min} =$-400.055099999999584.

Problem g24

Minimize $f(x) = -x_1 - x_2$

Subject to:

$$g_1(x) = -2x_1^4 + 8x_1^3 - 8x_1^2 + x_2 - 2 \le 0,$$
$$g_2(x) = -4x_1^4 + 32x_1^3 - 88x_1^2 + 96x_1 + x_2 - 36 \le 0,$$

The bounds on the variables are $0 \le x_1 \le 3,\ 0 \le x_2 \le 4$. This problem has one global minima at

$x^* =$ (2.32952019747762, 3.17849307411774) with $f_{min} =$ -5.50801327159536.

Appendix C

Data Set Corresponding to Various Bus-systems for Relay Coordination Problem

Table C1. The value of the constants i_f^j, $CT_{primary\ rating}^j$, i_f^k and $CT_{primary\ rating}^k$ for 3-bus system

$T_{pr\_cl\_in}^j$			$T_{pr\_far\_bus}^k$		
TS^j	i_f^j	$CT_{primary\ rating}^j$	TS^k	i_f^k	$CT_{primary\ rating}^k$
TS^1	9.460	2.06	TS^1	14.08	2.06
TS^2	26.910	2.06	TS^2	100.63	2.06
TS^3	8.810	2.23	TS^3	12.07	2.23
TS^4	37.680	2.23	TS^4	136.23	2.23
TS^5	17.930	0.80	TS^5	25.90	0.80
TS^6	14.350	0.80	TS^6	19.20	0.80

Table C2. The value of the constants i_f^j, $CT_{primary\ rating}^j$, i_f^k and $CT_{primary\ rating}^k$ corresponding to selectivity constraints for 3-bus system

T_{backup}^j			$T_{primary}^k$		
Relay	i_f^j	$CT_{primary\ rating}^j$	Relay	i_f^k	$CT_{primary\ rating}^k$
5	14.08	2.06	1	14.08	2.06
6	12.07	2.06	3	12.07	2.23
4	25.90	2.23	5	25.90	0.80
2	14.35	2.06	6	14.35	2.06
5	9.46	0.80	1	9.46	2.06
6	8.81	0.80	3	8.81	2.23
4	19.20	2.06	6	19.20	0.80
2	17.93	2.23	5	17.93	0.80

Table C3. The value of the constants i_f^j, $CT_{primary\ rating}^j$, i_f^k and $CT_{primary\ rating}^k$ for 4-bus system

$T_{pr\_cl\_in}^j$			$T_{pr\_far\_bus}^k$		
TS^j	i_f^j	$CT_{primary\ rating}^j$	TS^k	i_f^k	$CT_{primary\ rating}^k$
TS^1	20.32	0.4800	TS^1	12.48	0.4800
TS^2	88.85	0.4800	TS^2	23.75	0.4800
TS^3	13.61	1.1789	TS^3	10.38	1.1789
TS^4	116.81	1.1789	TS^4	31.92	1.1789
TS^5	116.79	1.5259	TS^5	31.92	1.5259
TS^6	16.67	1.5259	TS^6	12.07	1.5259
TS^7	71.70	1.2018	TS^7	18.91	1.2018
TS^8	19.27	1.2018	TS^8	11.00	1.2018

Table C4. The value of the constants i_f^j, $CT_{primary\ rating}^j$, i_f^k and $CT_{primary\ rating}^k$ corresponding to selectivity constraints for 4-bus system

T_{backup}^j			$T_{primary}^k$		
Relay	i_f^j	$CT_{primary\ rating}^j$	**Relay**	i_f^k	$CT_{primary\ rating}^k$
5	20.32	1.5259	**1**	20.32	0.4800
5	12.48	1.5259	**1**	12.48	0.4800
7	13.61	1.2018	**3**	13.61	1.1789
7	10.38	1.2018	**3**	10.38	1.1789
1	1.16	0.4800	**4**	116.81	1.1789
2	12.07	0.4800	**6**	12.07	1.5259
2	16.67	0.4800	**6**	16.67	1.5259
4	11.00	1.1789	**8**	11.00	1.2018
4	19.27	1.1789	**8**	19.27	1.2018

Table C5. The value of the constants i_f^j, $CT_{primary\ rating}^j$, i_f^k and $CT_{primary\ rating}^k$ for 6-bus system

$T_{pr\_cl\_in}^j$			$T_{pr\_far\_bus}^k$		
TS^j	i_f^j	$CT_{primary\ rating}^j$	TS^k	i_f^k	$CT_{primary\ rating}^k$
TS^1	2.5311	0.2585	TS^1	5.9495	0.2585
TS^2	2.7376	0.2585	TS^2	5.3752	0.2585
TS^3	2.9723	0.4863	TS^3	6.6641	0.4863
TS^4	4.1477	0.4863	TS^4	4.5897	0.4863
TS^5	1.9545	0.7138	TS^5	6.2345	0.7138
TS^6	2.7678	0.7138	TS^6	4.2573	0.7138
TS^7	3.8423	1.7460	TS^7	6.3694	1.7460
TS^8	5.6180	1.7460	TS^8	4.1783	1.7460
TS^9	4.6538	1.0424	TS^9	3.8700	1.0424
TS^{10}	3.5261	1.0424	TS^{10}	5.2696	1.0424
TS^{11}	2.5840	0.7729	TS^{11}	6.1144	0.7729
TS^{12}	3.8006	0.7729	TS^{12}	3.9005	0.7729
TS^{13}	2.4143	0.5879	TS^{13}	2.9011	0.5879
TS^{14}	5.3541	0.5879	TS^{14}	4.3350	0.5879

Table C6. The value of the constants i_f^j, $CT_{primary\,rating}^j$, i_f^k and $CT_{primary\,rating}^k$ corresponding to selectivity constraints for 6-bus system

$\boldsymbol{T_{backup}^j}$			$\boldsymbol{T_{primary}^k}$		
Relay	$\boldsymbol{i_f^j}$	$\boldsymbol{CT_{primary\,rating}^j}$	**Relay**	$\boldsymbol{i_f^k}$	$\boldsymbol{CT_{primary\,rating}^k}$
8	4.0909	1.7460	**1**	5.3752	0.2585
8	2.9323	1.7460	**1**	2.5311	0.2585
11	1.2886	0.7729	**1**	5.3752	0.2585
3	0.6213	0.4863	**2**	2.7376	0.2585
3	1.6658	0.4863	**2**	5.9495	0.2585
10	2.5610	1.0424	**3**	2.9723	0.4863
10	3.0923	1.0424	**3**	4.5897	0.4863
13	1.4995	0.5879	**3**	4.5897	0.4863
1	1.5243	0.2585	**4**	6.6641	0.4863
1	0.8869	0.2585	**4**	4.1477	0.4863
12	1.4549	0.7729	**5**	1.9545	0.7138
12	2.5444	0.7729	**5**	4.2573	0.7138
14	1.7142	0.5879	**5**	4.2573	0.7138
1	1.1231	0.2585	**6**	6.2345	0.7138
3	1.4658	0.4863	**6**	6.2345	0.7138
2	2.0355	0.2585	**7**	4.1783	1.7460
2	1.8718	0.2585	**7**	3.8423	1.7460
11	2.1436	0.7729	**7**	4.1783	1.7460
11	1.9712	0.7729	**7**	3.8423	1.7460
4	3.4386	0.4863	**9**	5.2696	1.0424
4	3.0368	0.4863	**9**	4.6538	1.0424
13	1.8321	0.5879	**9**	5.2696	1.0424
13	1.6180	0.5879	**9**	4.6538	1.0424
6	1.8138	0.7138	**11**	3.9005	0.7729
6	1.1099	0.7138	**11**	2.5840	0.7729
14	2.0871	0.5879	**11**	3.9005	0.7729
14	1.4744	0.5879	**11**	2.5840	0.7729
2	0.4734	0.2585	**12**	3.8006	0.7729
2	1.5432	0.2585	**12**	6.1144	0.7729
8	3.3286	1.7460	**12**	3.8006	0.7729
8	4.5736	1.7460	**12**	6.1144	0.7729
6	1.6085	0.7138	**13**	4.3350	0.5879
12	2.7269	0.7729	**13**	4.3350	0.5879
12	1.8360	0.7729	**13**	2.4143	0.5879
4	0.8757	0.4863	**14**	2.9011	0.5879
4	2.5823	0.4863	**14**	5.3541	0.5879
10	2.0260	1.0424	**14**	2.9011	0.5879
10	2.7784	1.0424	**14**	5.3541	0.5879

Table C7. The value of the constants i_f^j, $CT_{primary\ rating}^j$, i_f^k and $CT_{primary\ rating}^k$ for 14-bus system

$T_{pr\_cl\_in}^j$			$T_{pr\_far\_bus}^k$		
TS^j	i_f^j	$CT_{primary\ rating}^j$	TS^k	i_f^k	$CT_{primary\ rating}^k$
TS^1	9.2913	1.4883	TS^1	6.5448	1.4883
TS^2	1.2344	1.4883	TS^2	1.8339	1.4883
TS^3	6.0113	0.7143	TS^3	2.6433	0.7143
TS^4	1.9793	0.7143	TS^4	4.3575	0.7143
TS^5	6.4479	0.7040	TS^5	3.1281	0.7040
TS^6	3.1417	0.7040	TS^6	6.4878	0.7040
TS^7	6.2658	0.5364	TS^7	2.6728	0.5364
TS^8	2.8926	0.5364	TS^8	6.7856	0.5364
TS^9	5.9094	0.3989	TS^9	2.5043	0.3989
TS^{10}	2.9101	0.3989	TS^{10}	6.8739	0.3989
TS^{11}	6.1548	0.2389	TS^{11}	3.1983	0.2389
TS^{12}	3.4000	0.2389	TS^{12}	6.6622	0.2389
TS^{13}	5.1962	0.6129	TS^{13}	4.0312	0.6129
TS^{14}	5.1451	0.6129	TS^{14}	6.6726	0.6129
TS^{15}	4.1203	0.3375	TS^{15}	2.3424	0.3375
TS^{16}	3.2408	0.3375	TS^{16}	6.4102	0.3375
TS^{17}	3.2393	0.1642	TS^{17}	1.2685	0.1642
TS^{18}	1.5751	0.1642	TS^{18}	4.5904	0.1642
TS^{19}	5.6780	0.4462	TS^{19}	2.8500	0.4462
TS^{20}	2.8302	0.4462	TS^{20}	5.6145	0.4462
TS^{21}	1.7026	0.0806	TS^{21}	1.1769	0.0806
TS^{22}	3.4646	0.0806	TS^{22}	6.7384	0.0806
TS^{23}	1.4265	0.0772	TS^{23}	0.6528	0.0772
TS^{24}	2.8174	0.0772	TS^{24}	6.3513	0.0772
TS^{25}	1.6753	0.1818	TS^{25}	0.8932	0.1818
TS^{26}	4.2491	0.1818	TS^{26}	7.5912	0.1818
TS^{27}	5.3987	0.1748	TS^{27}	3.4374	0.1748
TS^{28}	2.8776	0.1748	TS^{28}	4.1301	0.1748
TS^{29}	3.0118	0.2719	TS^{29}	2.3332	0.2719
TS^{30}	3.6132	0.2719	TS^{30}	5.0393	0.2719
TS^{31}	1.6025	0.0588	TS^{31}	1.3932	0.0588
TS^{32}	3.7512	0.0588	TS^{32}	4.7782	0.0588
TS^{33}	1.3873	0.0947	TS^{33}	0.9942	0.0947
TS^{34}	2.3552	0.0947	TS^{34}	4.2483	0.0947
TS^{35}	2.5982	0.0433	TS^{35}	1.8567	0.0433
TS^{36}	2.1033	0.0433	TS^{36}	3.0181	0.0433
TS^{37}	3.5842	0.0179	TS^{37}	1.9815	0.0179
TS^{38}	1.2000	0.0179	TS^{38}	1.9971	0.0179
TS^{39}	1.7314	0.0596	TS^{39}	1.1341	0.0596
TS^{40}	1.8029	0.0596	TS^{40}	3.3438	0.0596

Table C8. The value of the constants i_f^j, $CT^j_{primary\,rating}$, i_f^k and $CT^k_{primary\,rating}$ corresponding to selectivity constraints for 14-bus system

T^j_{backup}			$T^k_{primary}$		
Relay	i_f^j	$CT^j_{primary\,rating}$	**Relay**	i_f^k	$CT^k_{primary\,rating}$
5	2.1554	0.7040	1	9.2913	1.4883
7	1.4652	0.5364	1	9.2913	1.4883
9	1.2524	0.3989	1	9.2913	1.4883
5	1.5929	0.7040	1	6.5448	1.4883
7	0.9171	0.5364	1	6.5448	1.4883
9	0.6463	0.3989	1	6.5448	1.4883
3	1.2344	0.7143	2	1.2344	1.4883
3	1.8339	0.7143	2	1.8339	1.4883
10	1.1509	0.3989	3	6.0113	0.7143
14	3.1662	0.6129	3	6.0113	0.7143
19	1.7088	0.4462	3	6.0113	0.7143
14	1.5165	0.6129	3	2.6433	0.7143
19	1.2078	0.4462	3	2.6433	0.7143
1	4.3575	1.4883	4	4.3575	0.7143
11	1.5283	0.2389	5	6.4479	0.7040
7	1.1184	0.5364	6	6.4878	0.7040
9	1.0336	0.3989	6	6.4878	0.7040
12	1.9806	0.2389	7	6.2658	0.5364
13	2.3154	0.6129	7	6.2658	0.5364
15	1.4418	0.3375	7	6.2658	0.5364
17	0.5423	0.1642	7	6.2658	0.5364
12	1.0887	0.2389	7	2.6728	0.5364
15	1.1197	0.3375	7	2.6728	0.5364
17	0.4140	0.1642	7	2.6728	0.5364
9	0.8095	0.3989	8	6.7856	0.5364
5	1.6993	0.7040	8	6.7856	0.5364
14	3.3518	0.6129	9	5.9094	0.3989
19	1.8535	0.4462	9	5.9094	0.3989
14	1.4259	0.6129	9	2.5043	0.3989
19	1.4814	0.4462	9	2.5043	0.3989
7	0.9609	0.5364	10	6.8739	0.3989
5	1.7512	0.7040	10	6.8739	0.3989
13	2.7200	0.6129	11	6.1548	0.2389
15	1.4654	0.3375	11	6.1548	0.2389
17	0.5528	0.1642	11	6.1548	0.2389
8	1.4184	0.5364	11	6.1548	0.2389
13	1.3332	0.6129	11	3.1983	0.2389
15	1.0109	0.3375	11	3.1983	0.2389
17	0.3791	0.1642	11	3.1983	0.2389
6	1.5526	0.7040	12	6.6622	0.2389
19	2.2251	0.4462	13	5.1962	0.6129
10	1.8327	0.3989	13	5.1962	0.6129
4	1.1499	0.7143	13	5.1962	0.6129
19	1.9086	0.4462	13	4.0312	0.6129

Relay	i_f^j	$CT_{primary\,rating}^j$	Relay	i_f^k	$CT_{primary\,rating}^k$
10	1.3082	0.3989	13	4.0312	0.6129
15	1.4640	0.3375	14	5.1451	0.6129
17	0.5283	0.1642	14	5.1451	0.6129
12	2.0902	0.2389	14	5.1451	0.6129
8	1.0740	0.5364	14	5.1451	0.6129
15	1.7474	0.3375	14	6.6726	0.6129
17	0.6535	0.1642	14	6.6726	0.6129
12	2.5475	0.2389	14	6.6726	0.6129
8	1.7383	0.5364	14	6.6726	0.6129
27	2.5389	0.1748	15	4.1203	0.3375
29	1.6000	0.2719	15	4.1203	0.3375
27	1.8665	0.1748	15	2.3424	0.3375
29	0.4991	0.2719	15	2.3424	0.3375
13	1.4492	0.6129	16	3.2408	0.3375
12	1.2568	0.2389	16	3.2408	0.3375
8	0.8968	0.5364	16	3.2408	0.3375
17	0.3181	0.1642	16	6.4102	0.3375
13	2.6267	0.6129	16	6.4102	0.3375
12	2.0146	0.2389	16	6.4102	0.3375
8	1.4608	0.5364	16	6.4102	0.3375
30	1.9335	0.2719	17	3.2393	0.1642
31	0.8180	0.0588	17	3.2393	0.1642
33	0.5290	0.0947	17	3.2393	0.1642
30	0.5860	0.2719	17	1.2685	0.1642
31	0.4261	0.0588	17	1.2685	0.1642
33	0.2756	0.0947	17	1.2685	0.1642
12	0.8171	0.2389	18	1.5751	0.1642
15	0.5615	0.3375	18	4.5904	0.1642
13	1.7162	0.6129	18	4.5904	0.1642
12	1.3473	0.2389	18	4.5904	0.1642
8	0.9739	0.5364	18	4.5904	0.1642
21	0.5760	0.0806	19	5.6780	0.4462
25	0.2974	0.1818	19	5.6780	0.4462
14	1.4757	0.6129	20	2.8302	0.4462
10	0.8328	0.3989	20	2.8302	0.4462
14	3.2590	0.6129	20	5.6145	0.4462
10	1.4484	0.3989	20	5.6145	0.4462
36	1.7026	0.0433	21	1.7026	0.0806
36	1.1769	0.0433	21	1.1769	0.0806
20	0.7589	0.4462	22	3.4646	0.0806
20	1.6087	0.4462	22	6.7384	0.0806
37	1.4265	0.0179	23	1.4265	0.0772
37	0.6528	0.0179	23	0.6528	0.0772
21	0.3086	0.0806	24	2.8174	0.0772
20	0.8810	0.4462	24	2.8174	0.0772
21	0.5996	0.0806	24	6.3513	0.0772
20	1.5316	0.4462	24	6.3513	0.0772
38	0.7848	0.0179	25	1.6753	0.1818
39	0.8925	0.0596	25	1.6753	0.1818

Relay	i_f^j	$CT_{primary\ rating}^j$	Relay	i_f^k	$CT_{primary\ rating}^k$
38	0.2091	0.0179	25	0.8932	0.1818
39	0.6851	0.0596	25	0.8932	0.1818
21	0.3319	0.0806	26	4.2491	0.1818
20	1.1625	0.4462	26	4.2491	0.1818
21	0.6907	0.0806	26	7.5912	0.1818
20	1.8301	0.4462	26	7.5912	0.1818
29	1.2730	0.2719	28	2.8776	0.1748
16	1.6111	0.3375	28	2.8776	0.1748
29	1.8270	0.2719	28	4.1301	0.1748
16	2.3124	0.3375	28	4.1301	0.1748
31	1.2257	0.0588	29	3.0118	0.2719
33	0.7928	0.0947	29	3.0118	0.2719
18	1.0273	0.1642	29	3.0118	0.2719
31	1.0031	0.0588	29	2.3332	0.2719
33	0.6488	0.0947	29	2.3332	0.2719
18	0.7063	0.1642	29	2.3332	0.2719
27	2.0912	0.1748	30	3.6132	0.2719
16	1.5226	0.3375	30	3.6132	0.2719
27	2.7262	0.1748	30	5.0393	0.2719
16	2.3145	0.3375	30	5.0393	0.2719
35	1.6025	0.0433	31	1.6025	0.0588
35	1.3932	0.0433	31	1.3932	0.0588
33	0.5585	0.0947	32	3.7512	0.0588
30	2.3926	0.2719	32	3.7512	0.0588
18	0.8254	0.1642	32	3.7512	0.0588
33	0.7640	0.0947	32	4.7782	0.0588
30	3.0042	0.2719	32	4.7782	0.0588
18	1.0475	0.1642	32	4.7782	0.0588
40	1.3873	0.0596	33	1.3873	0.0947
40	0.9942	0.0596	33	0.9942	0.0947
31	0.4315	0.0588	34	2.3552	0.0947
30	1.4469	0.2719	34	2.3552	0.0947
18	0.4890	0.1642	34	2.3552	0.0947
31	0.9445	0.0588	34	4.2483	0.0947
30	2.4759	0.2719	34	4.2483	0.0947
18	0.8600	0.1642	34	4.2483	0.0947
22	2.5982	0.0806	35	2.5982	0.0433
22	1.8567	0.0806	35	1.8567	0.0433
32	2.1033	0.0588	36	2.1033	0.0433
32	3.0181	0.0588	36	3.0181	0.0433
39	0.7780	0.0596	37	3.5842	0.0179
26	2.8071	0.1818	37	3.5842	0.0179
39	0.5065	0.0596	37	1.9815	0.0179
26	1.4752	0.1818	37	1.9815	0.0179
24	1.2000	0.0772	38	1.2000	0.0179
24	1.9971	0.0772	38	1.9971	0.0179
34	1.7314	0.0947	39	1.7314	0.0596
34	1.1341	0.0947	39	1.1341	0.0596
38	0.3683	0.0179	40	1.8029	0.0596

Relay	i_f^j	$CT_{primary\,rating}^j$	**Relay**	i_f^k	$CT_{primary\,rating}^k$
26	1.4391	0.1818	40	1.8029	0.0596
38	0.6831	0.0179	40	3.3438	0.0596
26	2.6692	0.1818	40	3.3438	0.0596

Bibliography

1. Himmelblau, D. M. (1972). Applied nonlinear programming. McGraw-Hill Companies.
2. Boyd, S., & Vandenberghe, L. (2004). Convex optimization. Cambridge University Press.
3. Rao, S. S. (2009). Engineering optimization: theory and practice. John Wiley & Sons.
4. Chandra, S., & Jayadeva, M. (2009). Numerical Optimization with Applications, Alpha Science International.
5. Mohan, C., & Deep, K. (2009). Optimization Techniques. New Age New Delhi.
6. Jha, P. C., & Hoda, M. N. (2011). Mathematical Modelling, Optimization and Their Applications. Narosa Publishing House.
7. Kapur, P. K., Pham, H., Gupta, A., & Jha, P. C. (2011). Software reliability assessment with OR applications. Springer.
8. Bertsekas, D. P. (2014). Constrained optimization and Lagrange multiplier methods. Academic Press.
9. Bertsekas, D. P., & Scientific, A. (2015). Convex optimization algorithms. Belmont: Athena Scientific.
10. Omran, M. G. (2004). Particle swarm optimization methods for pattern recognition and image processing (Doctoral dissertation, University of Pretoria).
11. Tripathi, R. K., Singh, Y. N., & Verma, N. K. (2013). Clustering algorithm for non-uniformly distributed nodes in wireless sensor network. Electronics Letters, 49(4), 299-300.
12. Singh, V., Dev, R., Dhar, N. K., Agrawal, P., & Verma, N. K. (2018). Adaptive type-2 fuzzy approach for filtering salt and pepper noise in grayscale images. IEEE Transactions on Fuzzy Systems, 26(5), 3170-3176.
13. Sevakula, R. K., & Verma, N. K. (2017). Compounding general purpose membership functions for fuzzy support vector machine under noisy environment. IEEE Transactions on Fuzzy Systems, 25(6), 1446-1459.
14. Holland, J. H. (1992). Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence. MIT Press.
15. Koza, J. R. (1992). Genetic programming: on the programming of computers by means of natural selection (Vol. 1). MIT Press.
16. Storn, R., & Price, K. (1997). Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization, 11(4), 341-359.

17. Ali, M. M., & Zhu, W. X. (2013). A penalty function-based differential evolution algorithm for constrained global optimization. Computational Optimization and Applications, 54(3), 707-739.
18. Monakhov, O. G., & Grosbein, E. B. (1998). A parallel genetic algorithm for the graphs mapping problem. In: 1998 International Workshop on Applied Parallel Computing (pp. 379-384). Springer.
19. Monakhova, E. A., Monakhov, O. G., & Mukhoed, E. V. (1999). Genetic construction of optimal circulant network designs. In: 1999 Workshops on Applications of Evolutionary Computation (pp. 215-223). Springer.
20. Monakhov, O. G., & Monakhova, E. (2002). Using evolutionary algorithm for generation of dense families of circulant networks. In: 2002 Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No. 02TH8600) (Vol. 2, pp. 1854-1859). IEEE.
21. Rakshit, P., Konar, A., Bhowmik, P., Goswami, I., Das, S., Jain, L. C., & Nagar, A. K. (2013). Realization of an adaptive memetic algorithm using differential evolution and q-learning: a case study in multirobot path planning. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 43(4), 814-831.
22. Rakshit, P., Konar, A., Das, S., Jain, L. C., & Nagar, A. K. (2013). Uncertainty management in differential evolution induced multiobjective optimization in presence of measurement noise. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 44(7), 922-937.
23. Srinu, S., & Sabat, S. L. (2014). Multinode sensing with forward error correction and differential evolution algorithms for noisy cognitive radio networks. Computers & Electrical Engineering, 40(4), 1090-1100.
24. Bhardwaj, A., Tiwari, A., Varma, M. V., & Krishna, M. R. (2014). Classification of EEG signals using a novel genetic programming approach. In: 2014 Proceedings of the Companion Publication of the 2014 Annual Conference on Genetic and Evolutionary Computation (pp. 1297-1304). ACM.
25. Bhardwaj, A., Tiwari, A., Krishna, R., & Varma, V. (2016). A novel genetic programming approach for epileptic seizure detection. Computer Methods and Programs in Biomedicine, 124, 2-18.
26. Bhardwaj, H., Sakalle, A., Bhardwaj, A., & Tiwari, A. (2019). Classification of electroencephalogram signal for the detection of epilepsy using Innovative Genetic Programming. Expert Systems, 36(1), e12338.
27. Kennedy, J. (2010). Particle swarm optimization. Encyclopedia of Machine Learning, 760-766.

28. Ali, M. M., & Kaelo, P. (2008). Improved particle swarm algorithms for global optimization. Applied Mathematics and Computation, 196(2), 578-593.
29. Sabat, S. L., Ali, L., & Udgata, S. K. (2011). Integrated learning particle swarm optimizer for global optimization. Applied Soft Computing, 11(1), 574-584.
30. Maitra, M., & Chatterjee, A. (2008). A hybrid cooperative–comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. Expert Systems with Applications, 34(2), 1341-1350.
31. Sharma, K. D., Chatterjee, A., & Rakshit, A. (2009). A hybrid approach for design of stable adaptive fuzzy controllers employing Lyapunov theory and particle swarm optimization. IEEE Transactions on Fuzzy Systems, 17(2), 329-342.
32. Ali, L., Sabat, S. L., & Udgata, S. K. (2012). Particle swarm optimisation with stochastic ranking for constrained numerical and engineering benchmark problems. International Journal of Bio-Inspired Computation, 4(3), 155-166.
33. Nwankwor, E., Nagar, A. K., & Reid, D. C. (2013). Hybrid differential evolution and particle swarm optimization for optimal well placement. Computational Geosciences, 17(2), 249-268.
34. Pedro, J. O., Dangor, M., Dahunsi, O. A., & Ali, M. M. (2014). Intelligent feedback linearization control of nonlinear electrohydraulic suspension systems using particle swarm optimization. Applied Soft Computing, 24, 50-62.
35. Mandal, D., Chatterjee, A., & Maitra, M. (2014). Robust medical image segmentation using particle swarm optimization aided level set based global fitting energy active contour approach. Engineering Applications of Artificial Intelligence, 35, 199-214.
36. Sadollah, A., Choi, Y., Yoo, D. G., & Kim, J. H. (2015). Metaheuristic algorithms for approximate solution to ordinary differential equations of longitudinal fins having various profiles. Applied Soft Computing, 33, 360-379.
37. Sadollah, A., Eskandar, H., Yoo, D. G., & Kim, J. H. (2015). Approximate solving of nonlinear ordinary differential equations using least square weight function and metaheuristic algorithms. Engineering Applications of Artificial Intelligence, 40, 117-132.
38. Krishnanand, K. N., & Ghose, D. (2006). Glowworm swarm based optimization algorithm for multimodal functions with collective robotics applications. Multiagent and Grid Systems, 2(3), 209-222.
39. Krishnanand, K. N., & Ghose, D. (2009). Glowworm swarm optimisation: a new method for optimising multi-modal functions. International Journal of Computational Intelligence Studies, 1(1), 93-119.

40. Karaboga, D. (2005). An idea based on honey bee swarm for numerical optimization (Vol. 200). Technical Report-tr06, Erciyes University, Engineering Faculty, Computer Engineering Department.
41. Sabat, S. L., Udgata, S. K., & Abraham, A. (2010). Artificial bee colony algorithm for small signal model parameter extraction of MESFET. Engineering Applications of Artificial Intelligence, 23(5), 689-694.
42. Biswas, S., Chatterjee, A., & Goswami, S. K. (2013). An artificial bee colony-least square algorithm for solving harmonic estimation problems. Applied Soft Computing, 13(5), 2343-2355.
43. Bansal, J. C., Gopal, A., & Nagar, A. K. (2018a). Analysing Convergence, Consistency, and Trajectory of Artificial Bee Colony Algorithm. IEEE Access, 6, 73593-73602.
44. Bansal, J. C., Gopal, A., & Nagar, A. K. (2018b). Stability analysis of artificial bee colony optimization algorithm. Swarm and Evolutionary Computation, 41, 9-19.
45. Mehrabian, A. R., & Lucas, C. (2006). A novel numerical optimization algorithm inspired from weed colonization. Ecological Informatics, 1(4), 355-366.
46. 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.
47. Naidu, Y. R., & Ojha, A. K. (2015). A hybrid version of invasive weed optimization with quadratic approximation. Soft Computing, 19(12), 3581-3598.
48. Prabha, D. R., & Jayabarathi, T. (2016). Optimal placement and sizing of multiple distributed generating units in distribution networks by invasive weed optimization algorithm. Ain Shams Engineering Journal, 7(2), 683-694.
49. Naidu, Y. R., & Ojha, A. K. (2018). A space transformational invasive weed optimization for solving fixed-point problems. Applied Intelligence, 48(4), 942-952.
50. Naidu, Y. R., & Ojha, A. K. (2018). Solving multiobjective optimization problems using hybrid cooperative invasive weed optimization with multiple populations. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(6), 821-832.
51. Bansal, J. C., Sharma, H., Jadon, S. S., & Clerc, M. (2014). Spider monkey optimization algorithm for numerical optimization. Memetic Computing, 6(1), 31-47.
52. Dorigo, M., Birattari, M., & Stutzle, T. (2006). Ant colony optimization. IEEE Computational Intelligence Magazine, 1(4), 28-39.
53. Passino, K. M. (2002). Biomimicry of bacterial foraging for distributed optimization and control. IEEE Control Systems, 22(3), 52-67.

54. Rashedi, E., Nezamabadi-Pour, H., & Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information Sciences, 179(13), 2232-2248.
55. Formato, R. A. (2008). Central force optimization: A new nature inspired computational framework for multidimensional search and optimization. In: 2008 Nature Inspired Cooperative Strategies for Optimization (NICSO) (pp. 221-238). Springer.
56. Formato, R. A. (2009). Central force optimization: A new deterministic gradient-like optimization metaheuristic. OPSEARCH, 46(1), 25-51.
57. Formato, R. A. (2010). Parameter-free deterministic global search with simplified central force optimization. In: 2010 International Conference on Intelligent Computing (pp. 309-318). Springer.
58. Geem, Z. W., Kim, J. H., & Loganathan, G. V. (2001). A new heuristic optimization algorithm: harmony search. Simulation, 76(2), 60-68.
59. Yoo, D. G., Kim, J. H., & Geem, Z. W. (2014). Overview of Harmony Search algorithm and its applications in Civil Engineering. Evolutionary Intelligence, 7(1), 3-16.
60. Sadollah, A., Sayyaadi, H., Yoo, D. G., Lee, H. M., & Kim, J. H. (2018). Mine blast harmony search: a new hybrid optimization method for improving exploration and exploitation capabilities. Applied Soft Computing, 68, 548-564.
61. Shah-Hosseini, H. (2009). Optimization with the nature-inspired intelligent water drops algorithm. In Evolutionary computation. InTechOpen.
62. Mirjalili, S. (2015a). The ant lion optimizer. Advances in Engineering Software, 83, 80-98.
63. Tan, Y., & Zhu, Y. (2010). Fireworks algorithm for optimization. In: 2010 International Conference in Swarm Intelligence (pp. 355-364). Springer.
64. Rao, R. V., Savsani, V. J., & Vakharia, D. P. (2011). Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design, 43(3), 303-315.
65. Jaddi, N. S., Alvankarian, J., & Abdullah, S. (2017). Kidney-inspired algorithm for optimization problems. Communications in Nonlinear Science and Numerical Simulation, 42, 358-369.
66. Mirjalili, S. (2015b). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 89, 228-249.
67. Ali, M. M., & Törn, A. (2004). Population set-based global optimization algorithms: some modifications and numerical studies. Computers & Operations Research, 31(10), 1703-1725.
68. Simon, D. (2013). Evolutionary optimization algorithms. John Wiley & Sons.

69. Yang, X., S., (2014). Nature-Inspired Optimization Algorithms. Elsevier, First Edition, p 263.
70. Yang, X. S., Chien, S. F., & Ting, T. O. (2015). Bio-inspired computation in telecommunications. Morgan Kaufmann. First Edition.
71. Kar, A. K. (2016). Bio inspired computing–A review of algorithms and scope of applications. Expert Systems with Applications, 59, 20-32.
72. Arora, J. (2017). Introduction to Optimum Design. Elsevier, Fourth Edition, pp 739–769.
73. Wolpert, D. H., and Macready, W. G. (1995). No free lunch theorems for search. Technical Report SFI-TR-95-02-010, Santa Fe Institute, New Mexico.
74. Wolpert, D. H., and Macready, W. G. (1997). No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation, 1(1), 67-82.
75. Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69, 46-61.
76. Muro, C., Escobedo, R., Spector, L., & Coppinger, R. P. (2011). Wolf-pack (Canis lupus) hunting strategies emerge from simple rules in computational simulations. Behavioural Processes, 88(3), 192-197.
77. Črepinšek, M., Liu, S. H., & Mernik, M. (2013). Exploration and exploitation in evolutionary algorithms: A survey. ACM Computing Surveys (CSUR), 45(3), 35.
78. Mittal, N., Singh, U., & Sohi, B. S. (2016). Modified grey wolf optimizer for global engineering optimization. Applied Computational Intelligence and Soft Computing, 2016, 8.
79. Long, W., Liang, X., Cai, S., Jiao, J., & Zhang, W. (2017). A modified augmented Lagrangian with improved grey wolf optimization to constrained optimization problems. Neural Computing and Applications, 28(1), 421-438.
80. Malik, M. R. S., Mohideen, E. R., & Ali, L. (2015). Weighted distance grey wolf optimizer for global optimization problems. In: 2015 International Conference on Computational Intelligence and Computing Research (ICCIC), IEEE, pp. 1-6.
81. Heidari, A. A., & Pahlavani, P. (2017). An efficient modified grey wolf optimizer with Lévy flight for optimization tasks. Applied Soft Computing, 60, 115-134.
82. Amirsadri, S., Mousavirad, S. J., & Ebrahimpour-Komleh, H. (2018). A Levy flight-based grey wolf optimizer combined with back-propagation algorithm for neural network training. Neural Computing and Applications, 30(12), 3707-3720.
83. Yang, B., Zhang, X., Yu, T., Shu, H., & Fang, Z. (2017). Grouped grey wolf optimizer for maximum power point tracking of doubly-fed induction generator based wind turbine. Energy Conversion and Management, 133, 427-443.

84. Rodríguez, L., Castillo, O., & Soria, J. (2016). Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic. In: 2016 IEEE Congress on Evolutionary Computation (CEC) (pp. 3116-3123). IEEE.
85. Rodríguez, L., Castillo, O., & Soria, J. (2017). A study of parameters of the grey wolf optimizer algorithm for dynamic adaptation with fuzzy logic. In: 2017 Nature-Inspired Design of Hybrid Intelligent Systems (pp. 371-390). Springer.
86. Emary, E., Zawbaa, H. M., & Grosan, C. (2018). Experienced gray wolf optimization through reinforcement learning and neural networks. IEEE Transactions on Neural Networks and Learning Systems, 29(3), 681-694.
87. Kaveh, A., & Zakian, P. (2018). Improved GWO algorithm for optimal design of truss structures. Engineering with Computers, 34(4), 685-707.
88. Long, W., Jiao, J., Liang, X., & Tang, M. (2018). Inspired grey wolf optimizer for solving large-scale function optimization problems. Applied Mathematical Modelling, 60, 112-126.
89. Kumar, V., & Kumar, D. (2017). An astrophysics-inspired Grey wolf algorithm for numerical optimization and its application to engineering design problems. Advances in Engineering Software, 112, 231-254.
90. Lu, C., Gao, L., & Yi, J. (2018). Grey wolf optimizer with cellular topological structure. Expert Systems with Applications, 107, 89-114.
91. Long, W., Jiao, J., Liang, X., & Tang, M. (2018). An exploration-enhanced grey wolf optimizer to solve high-dimensional numerical optimization. Engineering Applications of Artificial Intelligence, 68, 63-80.
92. Khandelwal, A., Bhargava, A., Sharma, A., & Sharma, H. (2018). Modified grey wolf optimization algorithm for transmission network expansion planning problem. Arabian Journal for Science and Engineering, 43(6), 2899-2908.
93. Rodríguez, L., Castillo, O., Soria, J., Melin, P., Valdez, F., Gonzalez, C. I., & Soto, J. (2017). A fuzzy hierarchical operator in the grey wolf optimizer algorithm. Applied Soft Computing, 57, 315-328.
94. Saremi, S., Mirjalili, S. Z., & Mirjalili, S. M. (2015). Evolutionary population dynamics and grey wolf optimizer. Neural Computing and Applications, 26(5), 1257-1263.
95. Jayabarathi, T., Raghunathan, T., Adarsh, B. R., & Suganthan, P. N. (2016). Economic dispatch using hybrid grey wolf optimizer. Energy, 111, 630-641.
96. Sanjay, R., Jayabarathi, T., Raghunathan, T., Ramesh, V., & Mithulananthan, N. (2017). Optimal allocation of distributed generation using hybrid grey wolf optimizer. IEEE Access, 5, 14807-14818.

97. Pradhan, M., Roy, P. K., & Pal, T. (2017). Oppositional based grey wolf optimization algorithm for economic dispatch problem of power system. Ain Shams Engineering Journal, 9(4), 2015-2025.
98. Zhang, S., Luo, Q., & Zhou, Y. (2017). Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method. International Journal of Computational Intelligence and Applications, 16(2), 1750012.
99. Oliveira, J., Oliveira, P. M., Boaventura-Cunha, J., & Pinho, T. (2017). Chaos-based grey wolf optimizer for higher order sliding mode position control of a robotic manipulator. Nonlinear Dynamics, 90(2), 1353-1362.
100. Kohli, M., & Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of Computational Design and Engineering, 5(4), 458-472.
101. Teng, Z. J., Lv, J. L., & Guo, L. W. (2018). An improved hybrid grey wolf optimization algorithm. Soft Computing, 1-15. DOI: https://doi.org/10.1007/s00500-018-3310-y
102. Tripathi, A. K., Sharma, K., & Bala, M. (2018). A novel clustering method using enhanced grey wolf optimizer and mapreduce. Big Data Research, 14, 93-100.
103. Saxena, A., Kumar, R., & Das, S. (2019). β-Chaotic map enabled Grey Wolf Optimizer. Applied Soft Computing, 75, 84-105.
104. Al-Betar, M. A., Awadallah, M. A., Faris, H., Aljarah, I., & Hammouri, A. I. (2018). Natural selection methods for grey wolf optimizer. Expert Systems with Applications, 113, 481-498.
105. Long, W., Wu, T., Cai, S., Liang, X., Jiao, J., & Xu, M. (2019). A Novel Grey Wolf Optimizer Algorithm with Refraction Learning. IEEE Access, 7, 57805-57819.
106. Heidari, A. A., Abbaspour, R. A., & Chen, H. (2019). Efficient boosted grey wolf optimizers for global search and kernel extreme learning machine training. Applied Soft Computing, 81, 105521. DOI: https://doi.org/10.1016/j.asoc.2019.105521
107. Tawhid, M. A., & Ali, A. F. (2017). A hybrid grey wolf optimizer and genetic algorithm for minimizing potential energy function. Memetic Computing, 9(4), 347-359.
108. Gu, Q., Li, X., & Jiang, S. (2019). Hybrid Genetic Grey Wolf Algorithm for Large-Scale Global Optimization. Complexity, 2019. DOI: https://doi.org/10.1155/2019/2653512.
109. Zhu, A., Xu, C., Li, Z., Wu, J., & Liu, Z. (2015). Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC. Journal of Systems Engineering and Electronics, 26(2), 317-328.
110. Debnath, M. K., Mallick, R. K., & Sahu, B. K. (2017). Application of hybrid differential evolution–grey wolf optimization algorithm for automatic generation control of a multi-source interconnected power system using optimal fuzzy–PID controller. Electric Power Components and Systems, 45(19), 2104-2117.

111. Tawhid, M. A., & Ibrahim, A. M. (2019). A hybridization of grey wolf optimizer and differential evolution for solving nonlinear systems. Evolving Systems, 1-23. DOI: https://doi.org/10.1007/s12530-019-09291-8
112. Kamboj, V. K. (2016). A novel hybrid PSO–GWO approach for unit commitment problem. Neural Computing and Applications, 27(6), 1643-1655.
113. Singh, N., & Singh, S. B. (2017). Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance. Journal of Applied Mathematics, 2017. DOI: https://doi.org/10.1155/2017/2030489
114. Singh, N., & Singh, S. B. (2017). A novel hybrid GWO-SCA approach for optimization problems. Engineering Science and Technology, An International Journal, 20(6), 1586-1601.
115. Zhang, X., Kang, Q., Cheng, J., & Wang, X. (2018). A novel hybrid algorithm based on Biogeography-Based Optimization and Grey Wolf Optimizer. Applied Soft Computing, 67, 197-214.
116. Daniel, E., Anitha, J., & Gnanaraj, J. (2017). Optimum laplacian wavelet mask based medical image using hybrid cuckoo search–grey wolf optimization algorithm. Knowledge-Based Systems, 131, 58-69.
117. Ibrahim, R. A., Elaziz, M. A., & Lu, S. (2018). Chaotic opposition-based grey-wolf optimization algorithm based on differential evolution and disruption operator for global optimization. Expert Systems with Applications, 108, 1-27.
118. Gaidhane, P. J., & Nigam, M. J. (2018). A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems. Journal of Computational Science, 27, 284-302.
119. Barraza, J., Rodríguez, L., Castillo, O., Melin, P., & Valdez, F. (2018). A new hybridization approach between the fireworks algorithm and grey wolf optimizer algorithm. Journal of Optimization, 2018. DOI: https://doi.org/10.1155/2018/6495362
120. Rodríguez, L., Castillo, O., García, M., & Soria, J. (2019). Firefly Algorithm and Grey Wolf Optimizer for Constrained Real-Parameter Optimization. In: 2019 International Fuzzy Systems Association World Congress (pp. 531-541). Springer.
121. Luo, Q., Zhang, S., Li, Z., & Zhou, Y. (2015). A novel complex-valued encoding grey wolf optimization algorithm. Algorithms, 9(1), 4.
122. Mirjalili, S., Saremi, S., Mirjalili, S. M., & Coelho, L. D. S. (2016). Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Systems with Applications, 47, 106-119.

123. Siddavaatam, P., & Sedaghat, R. (2019). Grey Wolf Optimizer Driven design space exploration: A novel framework for multi-objective trade-off in architectural synthesis. Swarm and Evolutionary Computation, 49, 44-61.
124. Emary, E., Zawbaa, H. M., & Hassanien, A. E. (2016). Binary grey wolf optimization approaches for feature selection. Neurocomputing, 172, 371-381.
125. Mirjalili, S. (2015c). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Applied Intelligence, 43(1), 150-161.
126. Song, X., Tang, L., Zhao, S., Zhang, X., Li, L., Huang, J., & Cai, W. (2015). Grey Wolf Optimizer for parameter estimation in surface waves. Soil Dynamics and Earthquake Engineering, 75, 147-157.
127. Komaki, G. M., & Kayvanfar, V. (2015). Grey Wolf Optimizer algorithm for the two-stage assembly flow shop scheduling problem with release time. Journal of Computational Science, 8, 109-120.
128. Madadi, A., & Motlagh, M. M. (2014). Optimal control of DC motor using grey wolf optimizer algorithm. Technical Journal of Engineering and Applied Science, 4(4), 373-379.
129. El-Fergany, A. A., & Hasanien, H. M. (2015). Single and multi-objective optimal power flow using grey wolf optimizer and differential evolution algorithms. Electric Power Components and Systems, 43(13), 1548-1559.
130. Muangkote, N., Sunat, K., & Chiewchanwattana, S. (2014). An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets. In: 2014 International Computer Science and Engineering Conference (ICSEC) (pp. 209-214). IEEE.
131. Kamboj, V. K., Bath, S. K., & Dhillon, J. S. (2016). Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer. Neural Computing and Applications, 27(5), 1301-1316.
132. Sultana, U., Khairuddin, A. B., Mokhtar, A. S., Zareen, N., & Sultana, B. (2016). Grey wolf optimizer based placement and sizing of multiple distributed generation in the distribution system. Energy, 111, 525-536.
133. Emary, E., Yamany, W., Hassanien, A. E., & Snasel, V. (2015). Multi-objective gray-wolf optimization for attribute reduction. Procedia Computer Science, 65, 623-632.
134. Emary, E., Zawbaa, H. M., Grosan, C., & Hassenian, A. E. (2015). Feature subset selection approach by gray-wolf optimization. In: 2015 Afro-European Conference for Industrial Advancement (pp. 1-13). Springer.
135. Zhang, S., Zhou, Y., Li, Z., & Pan, W. (2016). Grey wolf optimizer for unmanned combat aerial vehicle path planning. Advances in Engineering Software, 99, 121-136.

136. Lu, C., Gao, L., Li, X., & Xiao, S. (2017). A hybrid multi-objective grey wolf optimizer for dynamic scheduling in a real-world welding industry. Engineering Applications of Artificial Intelligence, 57, 61-79.
137. Khairuzzaman, A. K. M., & Chaudhury, S. (2017). Multilevel thresholding using grey wolf optimizer for image segmentation. Expert Systems with Applications, 86, 64-76.
138. Guha, D., Roy, P. K., & Banerjee, S. (2016). Load frequency control of interconnected power system using grey wolf optimization. Swarm and Evolutionary Computation, 27, 97-115.
139. Li, S. Y., Wang, S. M., Wang, P. F., Su, X. L., Zhang, X. S., & Dong, Z. H. (2018). An improved grey wolf optimizer algorithm for the inversion of geoelectrical data. Acta Geophysica, 66(4), 607-621.
140. Zhang, S., & Zhou, Y. (2017). Template matching using grey wolf optimizer with lateral inhibition. Optik-International Journal for Light and Electron Optics, 130, 1229-1243.
141. Medjahed, S. A., Saadi, T. A., Benyettou, A., & Ouali, M. (2016). Gray Wolf Optimizer for hyperspectral band selection. Applied Soft Computing, 40, 178-186.
142. Rameshkumar, J., Ganesan, S., Subramanian, S., & Abirami, M. (2016). Short-term unit consignment solution using real-coded grey wolf algorithm. Australian Journal of Electrical and Electronics Engineering, 13(1), 54-66.
143. Sulaiman, M. H., Mustaffa, Z., Mohamed, M. R., & Aliman, O. (2015). Using the gray wolf optimizer for solving optimal reactive power dispatch problem. Applied Soft Computing, 32, 286-292.
144. Mirjalili, S., Aljarah, I., Mafarja, M., Heidari, A. A., & Faris, H. (2020). Grey Wolf Optimizer: Theory, Literature Review, and Application in Computational Fluid Dynamics Problems. In Nature-Inspired Optimizers (pp. 87-105). Springer.
145. Mirjalili, S. (2016). SCA: a sine cosine algorithm for solving optimization problems. Knowledge-Based Systems, 96, 120-133.
146. Sindhu, R., Ngadiran, R., Yacob, Y. M., Zahri, N. A. H., & Hariharan, M. (2017). Sine–cosine algorithm for feature selection with elitism strategy and new updating mechanism. Neural Computing and Applications, 28(10), 2947-2958.
147. Meshkat, M., & Parhizgar, M. (2017). A novel weighted update position mechanism to improve the performance of sine cosine algorithm. In: 2017 5th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS) (pp. 166-171). IEEE.
148. Long, W., Wu, T., Liang, X., & Xu, S. (2019). Solving high-dimensional global optimization problems using an improved sine cosine algorithm. Expert Systems with Applications, 123, 108-126.

149. Rizk-Allah, R. M. (2018). Hybridizing sine cosine algorithm with multi-orthogonal search strategy for engineering design problems. Journal of Computational Design and Engineering, 5(2), 249-273.
150. Elaziz, M. A., Oliva, D., & Xiong, S. (2017). An improved opposition-based sine cosine algorithm for global optimization. Expert Systems with Applications, 90, 484-500.
151. Bairathi, D., & Gopalani, D. (2017). Opposition-based sine cosine algorithm (OSCA) for training feed-forward neural networks. In: 2017 13th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) (pp. 438-444). IEEE.
152. Turgut, O. E. (2017). Thermal and economical optimization of a shell and tube evaporator using hybrid backtracking search—sine–cosine algorithm. Arabian Journal for Science and Engineering, 42(5), 2105-2123.
153. Li, N., Li, G., & Deng, Z. (2017). An improved sine cosine algorithm based on levy flight. In: 2017 Ninth International Conference on Digital Image Processing (ICDIP 2017) (Vol. 10420, p. 104204R). International Society for Optics and Photonics.
154. Qu, C., Zeng, Z., Dai, J., Yi, Z., & He, W. (2018). A modified sine-cosine algorithm based on neighborhood search and greedy levy mutation. Computational Intelligence and Neuroscience, 2018. DOI: https://doi.org/10.1155/2018/4231647.
155. Guo, W., Wang, Y., Zhao, F., & Dai, F. (2019). Riesz fractional derivative Elite-guided sine cosine algorithm. Applied Soft Computing, 81, 105481.
156. Liu, S., Feng, Z. K., Niu, W. J., Zhang, H. R., & Song, Z. G. (2019). Peak Operation Problem Solving for Hydropower Reservoirs by Elite-Guide Sine Cosine Algorithm with Gaussian Local Search and Random Mutation. Energies, 12(11), 2189.
157. Cheng, J., & Duan, Z. (2019). Cloud model based sine cosine algorithm for solving optimization problems. Evolutionary Intelligence, 1-12. DOI: https://doi.org/10.1007/s12065-019-00251-4
158. Nenavath, H., & Jatoth, R. K. (2018). Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking. Applied Soft Computing, 62, 1019-1043.
159. Nenavath, H., Jatoth, R. K., & Das, S. (2018). A synergy of the sine-cosine algorithm and particle swarm optimizer for improved global optimization and object tracking. Swarm and Evolutionary Computation, 43, 1-30.
160. Issa, M., Hassanien, A. E., Oliva, D., Helmi, A., Ziedan, I., & Alzohairy, A. (2018). ASCA-PSO: Adaptive sine cosine optimization algorithm integrated with particle swarm for pairwise local sequence alignment. Expert Systems with Applications, 99, 56-70.

161. Nenavath, H., & Jatoth, R. K. Hybrid SCA–TLBO: a novel optimization algorithm for global optimization and visual tracking. Neural Computing and Applications, 1-30. DOI: https://doi.org/10.1007/s00521-018-3376-6

162. Li, C., Luo, Z., Song, Z., Yang, F., Fan, J., & Liu, P. X. (2019). An enhanced brain storm sine cosine algorithm for global optimization problems. IEEE Access, 7, 28211-28229.

163. Tawhid, M. A., & Savsani, V. (2019). Multi-objective sine-cosine algorithm (MO-SCA) for multi-objective engineering design problems. Neural Computing and Applications, 31(2), 915-929.

164. Reddy, K. S., Panwar, L. K., Panigrahi, B. K., & Kumar, R. (2018). A New Binary Variant of Sine–Cosine Algorithm: Development and Application to Solve Profit-Based Unit Commitment Problem. Arabian Journal for Science and Engineering, 43(8), 4041-4056.

165. Hafez, A. I., Zawbaa, H. M., Emary, E., & Hassanien, A. E. (2016). Sine cosine optimization algorithm for feature selection. In: 2016 International Symposium on Innovations in Intelligent Systems and Applications (INISTA) (pp. 1-5). IEEE.

166. Wang, J., Yang, W., Du, P., & Niu, T. (2018). A novel hybrid forecasting system of wind speed based on a newly developed multi-objective sine cosine algorithm. Energy Conversion and Management, 163, 134-150.

167. Sahlol, A. T., Ewees, A. A., Hemdan, A. M., & Hassanien, A. E. (2016). Training feedforward neural networks using Sine-Cosine algorithm to improve the prediction of liver enzymes on fish farmed on nano-selenite. In: 2016 12th International Computer Engineering Conference (ICENCO) (pp. 35-40). IEEE.

168. Elfattah, M. A., Abuelenin, S., Hassanien, A. E., & Pan, J. S. (2016). Handwritten arabic manuscript image binarization using sine cosine optimization algorithm. In: 2016 International Conference on Genetic and Evolutionary Computing (pp. 273-280). Springer.

169. Attia, A. F., El Sehiemy, R. A., & Hasanien, H. M. (2018). Optimal power flow solution in power systems using a novel Sine-Cosine algorithm. International Journal of Electrical Power & Energy Systems, 99, 331-343.

170. Das, S., Bhattacharya, A., & Chakraborty, A. K. (2018). Solution of short-term hydrothermal scheduling using sine cosine algorithm. Soft Computing, 22(19), 6409-6427.

171. Kumar, V., & Kumar, D. (2017). Data clustering using sine cosine algorithm: Data clustering using SCA. In Handbook of Research on Machine Learning Innovations and Trends (pp. 715-726). IGI Global.

172. Oliva, D., Hinojosa, S., Elaziz, M. A., & Ortega-Sánchez, N. (2018). Context based image segmentation using antlion optimization and sine cosine algorithm. Multimedia Tools and Applications, 77(19), 25761-25797.

173. Banerjee, A., & Nabi, M. (2017). Re-entry trajectory optimization for space shuttle using Sine-Cosine algorithm. In: 2017 8th International Conference on Recent Advances in Space Technologies (RAST) (pp. 73-77). IEEE.

174. Algabalawy, M. A., Abdelaziz, A. Y., Mekhamer, S. F., & Aleem, S. H. A. (2018). Considerations on optimal design of hybrid power generation systems using whale and sine cosine optimization algorithms. Journal of Electrical Systems and Information Technology, 5(3), 312-325.

175. Rizk-Allah, R. M., & El-Sehiemy, R. A. (2018). A novel sine cosine approach for single and multiobjective emission/economic load dispatch problem. In: 2018 International Conference on Innovative Trends in Computer Engineering (ITCE) (pp. 271-277). IEEE.

176. Singh, V. P. (2017). Sine cosine algorithm based reduction of higher order continuous systems. In: 2017 International Conference on Intelligent Sustainable Systems (ICISS) (pp. 649-653). IEEE.

177. Majhi, S. K. (2018). An efficient feed foreword network model with sine cosine algorithm for breast cancer classification. International Journal of System Dynamics Applications (IJSDA), 7(2), 1-14.

178. Issa, M., Hassanien, A. E., Helmi, A., Ziedan, I., & Alzohairy, A. (2018). Pairwise global sequence alignment using Sine-Cosine optimization algorithm. In: 2018 International Conference on Advanced Machine Learning Technologies and Applications (pp. 102-111). Springer.

179. Fatlawi, A., Vahedian, A., & Bachache, N. K. (2018). Optimal Camera Placement Using Sine-Cosine Algorithm. In: 2018 8th International Conference on Computer and Knowledge Engineering (ICCKE) (pp. 115-119). IEEE.

180. Nayak, N., Mishra, S., Sharma, D., & Sahu, B. K. (2019). Application of modified sine cosine algorithm to optimally design PID/fuzzy-PID controllers to deal with AGC issues in deregulated power system. IET Generation, Transmission & Distribution. DOI: 10.1049/iet-gtd.2018.6489

181. Mishra, S., Gupta, S., & Yadav, A. (2018). Design and Application of Controller Based on Sine-Cosine Algorithm for Load Frequency Control of Power System. In: 2018 International Conference on Intelligent Systems Design and Applications (pp. 301-311). Springer, Cham.

182. Mirjalili, S. M., Mirjalili, S. Z., Saremi, S., & Mirjalili, S. (2020). Sine Cosine Algorithm: Theory, Literature Review, and Application in Designing Bend Photonic Crystal Waveguides. In: 2020 Nature-Inspired Optimizers (pp. 201-217). Springer.

183. Feller, W. (2008). An introduction to probability theory and its applications (Vol. 2). John Wiley & Sons. Second Edition, Vol II.

184. Yang, X. S. (2010). Nature-inspired metaheuristic algorithms. Luniver Press. Second Edition.
185. Liang, J. J., Qu, B. Y., & Suganthan, P. N. (2013). Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization. Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.
186. Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1(1), 3-18.
187. Tizhoosh, H. R. (2005). Opposition-based learning: a new scheme for machine intelligence. In: 2005 International Conference on Computational Intelligence for Modelling, Control and Automation and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'06) (Vol. 1, pp. 695-701). IEEE.
188. Ventresca, M., & Tizhoosh, H. R. (2008). Numerical condition of feedforward networks with opposite transfer functions. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence) (pp. 3233-3240). IEEE.
189. Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition versus randomness in soft computing techniques. Applied Soft Computing, 8(2), 906-918.
190. Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. Swarm and Evolutionary Computation, 39, 1-23.
191. Deb, K. (2000). An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering, 186(2-4), 311-338.
192. Wang, Y., & Cai, Z. (2009). A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems. Frontiers of Computer Science in China, 3(1), 38-52
193. Liang, J. J., Runarsson, T. P., Mezura-Montes, E., Clerc, M., Suganthan, P. N., 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), 8-31.
194. Bazi, Y., Bruzzone, L., & Melgani, F. (2007). Image thresholding based on the EM algorithm and the generalized Gaussian distribution. Pattern Recognition, 40(2), 619-634.
195. Otsu, N. (1979). A threshold selection method from gray-level histograms. IEEE Transactions on Systems, Man, and Cybernetics, 9(1), 62-66.
196. Kapur, J. N., Sahoo, P. K., & Wong, A. K. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. Computer Vision, Graphics, and Image Processing, 29(3), 273-285.

197. Li, C. H., & Tam, P. K. S. (1998). An iterative algorithm for minimum cross entropy thresholding. Pattern Recognition Letters, 19(8), 771-776.
198. Verma, N. K., Roy, A., & Vasikarla, S. (2010). Medical image segmentation using improved mountain clustering technique version-2. In: 2010 Seventh International Conference on Information Technology: New Generations (pp. 156-161). IEEE.
199. Horng, M. H. (2011). Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Systems with Applications, 38(11), 13785-13791.
200. 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.
201. Sathya, P. D., & Kayalvizhi, R. (2011). Optimal multilevel thresholding using bacterial foraging algorithm. Expert Systems with Applications, 38(12), 15549-15564.
202. Mlakar, U., Potočnik, B., & Brest, J. (2016). A hybrid differential evolution for optimal multilevel image thresholding. Expert Systems with Applications, 65, 221-232.
203. El Aziz, M. A., Ewees, A. A., & Hassanien, A. E. (2017). Whale optimization algorithm and moth-flame optimization for multilevel thresholding image segmentation. Expert Systems with Applications, 83, 242-256.
204. Damborg, M. J., Ramaswami, R., Venkata, S. S., & Postforoosh, J. M. (1984). Computer aided transmission protection system design part I: Alcorithms. IEEE Transactions on Power Apparatus and Systems, (1), 51-59.
205. Abyaneh, H. A., Al-Dabbagh, M., Karegar, H. K., Sadeghi, S. H. H., & Khan, R. J. (2003). A new optimal approach for coordination of overcurrent relays in interconnected power systems. IEEE Transactions on Power Delivery, 18(2), 430-435.
206. Fuller, J. F., Fuchs, E. F., & Roesler, D. J. (1988). Influence of harmonics on power distribution system protection. IEEE Transactions on Power Delivery, 3(2), 549-557.
207. Bedekar, P. P., Bhide, S. R., & Kale, V. S. (2011). Determining optimum TMS and PS of overcurrent relays using linear programming technique. In: 2011 8th Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand-Conference 2011 (pp. 700-703). IEEE.
208. Uthitsunthorn, D., & Kulworawanichpong, T. (2010). Optimal overcurrent relay coordination using genetic algorithms. In: 2010 International Conference on Advances in Energy Engineering (pp. 162-165). IEEE.
209. Singh, D. K., & Gupta, S. (2012). Use of genetic algorithms (GA) for optimal coordination of directional over current relays. In: 2012 Students Conference on Engineering and Systems (pp. 1-5). IEEE.

210. Deep, K., & Singh, D. (2016). Optimization of Directional Overcurrent Relay Times Using C-SOMGA. In: 2016 Self-Organizing Migrating Algorithm (pp. 167-186). Springer.
211. Thangaraj, R., Pant, M., & Abraham, A. (2010). New mutation schemes for differential evolution algorithm and their application to the optimization of directional over-current relay settings. Applied Mathematics and Computation, 216(2), 532-544.
212. Thangaraj, R., Pant, M., & Deep, K. (2010). Optimal coordination of over-current relays using modified differential evolution algorithms. Engineering Applications of Artificial Intelligence, 23(5), 820-829.
213. Chelliah, T. R., Thangaraj, R., Allamsetty, S., & Pant, M. (2014). Coordination of directional overcurrent relays using opposition based chaotic differential evolution algorithm. International Journal of Electrical Power & Energy Systems, 55, 341-350.
214. Zeineldin, H., El-Saadany, E. F., & Salama, M. A. (2005). Optimal coordination of directional overcurrent relay coordination. In: 2005 IEEE Power Engineering Society General Meeting, 2005 (pp. 1101-1106). IEEE.
215. Zeineldin, H. H., El-Saadany, E. F., & Salama, M. M. A. (2006). Optimal coordination of overcurrent relays using a modified particle swarm optimization. Electric Power Systems Research, 76(11), 988-995.
216. Mansour, M. M., Mekhamer, S. F., & El-Kharbawe, N. (2007). A modified particle swarm optimizer for the coordination of directional overcurrent relays. IEEE Transactions on Power Delivery, 22(3), 1400-1410.
217. Bansal, J. C., & Deep, K. (2008). Optimization of directional overcurrent relay times by particle swarm optimization. In: 2008 IEEE Swarm Intelligence Symposium (pp. 1-7). IEEE.
218. Deep, K., & Bansal, J. C. (2009). Optimization of directional overcurrent relay times using Laplace Crossover Particle Swarm Optimization (LXPSO). In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC) (pp. 288-293). IEEE.
219. Yang, M. T., & Liu, A. (2013). Applying hybrid PSO to optimize directional overcurrent relay coordination in variable network topologies. Journal of Applied Mathematics, 2013. DOI: http://dx.doi.org/10.1155/2013/879078
220. Birla, D., Maheshwari, R. P., Gupta, H. O., Deep, K., & Thakur, M. (2006). Application of random search technique in directional overcurrent relay coordination. International Journal of Emerging Electric Power Systems, 7(1).
221. Amraee, T. (2012). Coordination of directional overcurrent relays using seeker algorithm. IEEE Transactions on Power Delivery, 27(3), 1415-1422.
222. Thakur M. (2007). New real coded genetic algorithms for global optimization Ph.D. Thesis. India: Department of Mathematics, Indian Institute of Technology Roorkee.

223. Benmouyal, G., Meisinger, M., Burnworth, J., Elmore, W. A., Freirich, K., Kotos, P. A., & De Sa, J. P. (1999). IEEE standard inverse-time characteristic equations for overcurrent relays. IEEE Transactions on Power Delivery, 14(3), 868-872.

224. 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(11-12), 1245-1287.