EXTENDED GREY WOLF OPTIMIZATION ALGORITHM AND ITS APPLICATION FOR THE SYNTHESIS OF NON-UNIFORMLY SPACED ANTENNA ARRAYS

A thesis submitted in Partial Fulfilment of the Requirement for the Award of the Degree of

MASTER OF ENGINEERING

in

Electronics and Communication Engineering

Submitted by

SAKSHI SHARMA

Roll No. 801561026

Under Supervision of

Dr. URVINDER SINGH

Assistant Professor, ECED

Thapar University

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA, PUNJAB

JULY, 2017

DECLARATION

I Sakshi Sharma hereby declare that the work presented in this thesis entitled "EXTENDED GREY WOLF OPTIMIZATION ALGORITHM AND ITS APPLICATION FOR THE SYNTHESIS OF NON-UNIFORMLY SPACED ANTENNA ARRAYS" is partial fulfilment of the requirement for the award of degree of Master of Engineering submitted at Electronics and Communication Department, Thapar University, Patiala is an authentic record of work carried out under supervision of Dr. Urvinder Singh (Assistant Professor, Department of ECE, Thapar University) from 2015 to 2017. The matter presented in this has not been submitted either in part or full to any other university or institute for the award of any other degree.

Date..17-07-2017

Sakshi Sharma

Sakshi Sharma

801561026

It is certified that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date: 17/7/17

Dr. Urvinder Singh

Assistant Professor

ECED

ACKNOWLEDGEMENT

It is my proud privilege to acknowledge and extend my gratitude to several persons who helped me directly or indirectly in completion of this report. I would like to express sincerest gratitude to my esteemed mentor **Dr. Urvinder Singh**, Assistant Professor, Electronics and Communication Engineering Department, Thapar University, Patiala, for his regular devotion and valuable guidance throughout the completion of the dissertation. It has been a matter of privilege to get the opportunity to learn great insights of in-depth knowledge under his mentorship.

I would like to thank **Dr. Alpana Aggarwal,** Head, Electronics and Communication Engineering Department, Thapar University, for her earnest efforts to inspire the students to perform efficiently to meet the research objectives.

I express gratitude to **Dr. Amit Mishra,** Assistant Professor, ECE Program Coordinator and **Dr. Hem Dutt Joshi,** Assistant Professor, P.G. Coordinator Electronics and Communication Engineering Department, Thapar University for inspired motivation to give the best of the potential at each step.

I would also like to thank my friends **Ms. Komalpreet** and **Ms. Deepika** who have more or less contributed to the preparation of this report. I will be always indebted to them. Last but not the least, I would like to thank my parents for their years of unyielding love and encourage. They have always wanted the best for me and I admire their determination and sacrifice.

The study has indeed helped me to explore knowledge and avenues related to my topic and I am sure it will help me in my future.

Sakshi Sharma

ABSTRACT

Grey wolf optimization (GWO) is a recently developed nature-inspired global optimization method which mimics the social behaviour and hunting mechanism of grey wolves. The algorithm is very competitive and has been applied to various fields of research, but has poor exploration capability and suffers from local optima stagnation. So, in order to improve the explorative ability of GWO, an extended version of grey wolf optimization (GWO-E) algorithm is presented. This newly proposed algorithm consists of two modifications. Firstly, it is able to explore new areas in the search space because of diverse positions assigned to the leaders. This helps in increasing the exploration and avoids local optima stagnation problem. Secondly, an opposition based learning method has been used in the initial half of iterations to provide diversity among the search agents. This algorithm has been tested on nineteen standard benchmarking functions to prove its effectiveness over other state-of-the-art algorithms. Experimental results show that the GWO-E algorithm performs better than grey wolf optimization (GWO), bat algorithm (BA), bat flower pollinator (BFP), differential evolution (DE), firefly algorithm (FA) and flower pollination algorithm (FPA). Statistical testing of GWO-E has been done to prove its significance over other compared algorithm.

GWO-E is also tested on real world application of design of the non-uniformly spaced linear antenna array (NUSLA). Designing of antenna array is classical electromagnetic problem. This problem is highly complex and non-linear in nature. Hence, nature-inspired meta-heuristic algorithms are preferred for its synthesis. In this thesis, the NUSLA is designed with an aim to reduce the side lobe level (SLL) and placement of null in desired direction by controlling the inter-element spacing. Performance of GWO-E is evaluated by considering the several case studies of NUSLA that exists in literature and the results are compared with the results of other popular meta-heuristic algorithms. These results show the better performance of the proposed algorithm.

TABLE OF CONTENTS

LISTS OF TABLES

LISTS OF FIGURES

LIST OF ABBREVIATIONS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
BA	Bat Algorithm
BBBC	Big Bang Big Crunch
BFP	Bat Flower Pollination
BH	Black Hole
CAA	Circular Antenna Array
CS	Cuckoo Search
CSO	Cat Swarm Optimization
DE	Differential Evolution
EA	Evolutionary Algorithm
EP	Evolutionary Programming
EGWO	Enhanced Grey Wolf Optimization
E-GWO	Extended Grey Wolf Optimization
ES	Evolution Strategy
FA	Firefly Algorithm
FNBW	First Null Beam Width
FPA	Flower Pollination Algorithm
GA	Genetic Algorithm
GP	Genetic Programming
GSA	Gravitational Search Algorithm
GWO	Grey Wolf Optimization
HPBW	Half Power beam Width
HSA	Harmony Search Algorithm
LAA	Linear Antenna Array
NIA	Nature Inspired Algorithms
NUSLA	Non Uniformly Spaced Linear Array
PSO	Particle Swarm Optimization
SA	Simulated Annealing
SI	Swarm Intelligence
SLL	Side Lobe Level

CHAPTER 1
INTRODUCTION

1.1 OPTIMIZATION

Optimization is the act of achieving best possible results of the problem with given constraints. This process is being used in day to day planning, business decisions, in engineering problems and many more. The basic goal of optimization for all these problems is to minimize the efforts or maximizing the benefits. These efforts or benefits are usually presented as function of some design variables.

In engineering problems, optimization deals with the design and maintenance of the system. One of the factor which affects the design of the system is cost related with it. The optimum design can be obtained when the cost associated with it is lowest as compared with other possible designs. The design choices are also limited by several constraints such as: labor constraint, material constraint, environmental constraint and so on. Hence, design is also circumscribed by the constrained boundaries along with the cost. Thus, overall both the cost function and constrained function can be formulated as mathematical function in order to achieve optimum design. Hence, optimization of this function is a process of finding the condition that will either maximize it or minimize the function's value, depending upon the requirement of the problem. This type of the problem can be solved by using any optimization method. Initially, the optimization methods were developed by Newton [1], Lagrange [2], Cauchy [3] and Euler [4]. These optimization techniques make use of differential calculus for solving the different problems, but the scope of these algorithms are limited to continuous and differential functions only, hence are not suitable for today's practical problems. The study of these classical techniques form a basis for developing new meta-heuristic techniques which are suitable for today's real world example. Thus, over a decade a lot of meta-heuristic optimization algorithms have been designed and their efficiency has been proved for various real time application. Some of these algorithms are simulated annealing (SA) [5], genetic algorithm (GA) [6], differential evolution (DE) [7], ant colony optimization (ACO) [8], artificial bee colony algorithm (ABC) [9], particle swarm algorithm (PSO) [10], flower pollination algorithm (FPA) [11], grey wolf algorithms (GWO) [12], cuckoo search (CS) [13], bat algorithm (BA) [14] and so on. Among all these algorithms, GWO [12] is newly developed algorithm which mimics the social behavior of grey wolves and it performs well on various application as compare to other algorithms. The detail study of this algorithm is discussed in this thesis in the Chapter 2 and 3 and this algorithm has been used for the synthesis of antenna array.

1.2 TERMINOLOGY OF OPTIMIZATION

Optimization is the act of achieving the best possible solution of the problem relative to the set of some predefined constraints. The goal of optimization is to maximize the desired factors and to minimize the undesired ones. These factors are expressed in the form of mathematical function and the maxima and minima of function can be defined as

Maxima and minima of function: For a function *f(x)* shown in Figure 1.1, the point x^* shows the smallest value of the function and hence it corresponds to the minima of function. This same point is equivalent to the maxima of the negative of the function. As the maximum of the function can be found by calculating the minima of the negative of same function hence, in general, the optimization is taken as minimization process.

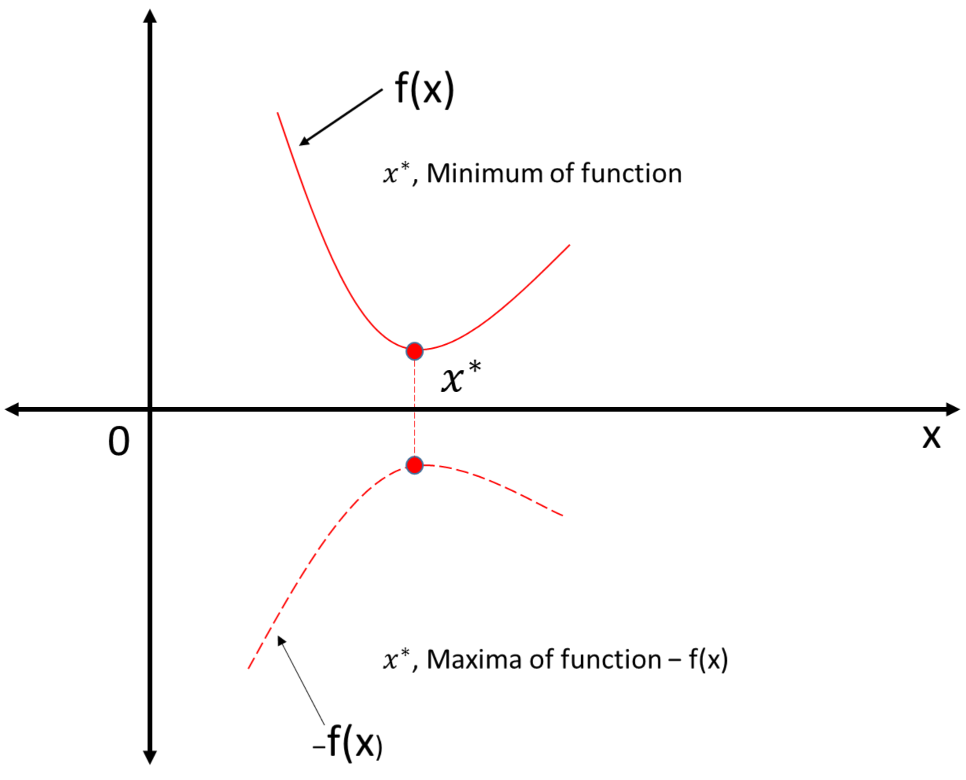


Figure 1.1 Minima and Maxima of a function

Types of minima

- Local minima: It is the smallest functional value in some feasible neighborhood such that it is not the lowest value in the entire set. It is also known as ‘relative minima’.
- Global minima: It is lowest value of the function over all feasible points. It is also known as ‘absolute minima’.

  These two classes of minima are shown in Figure 1.2.

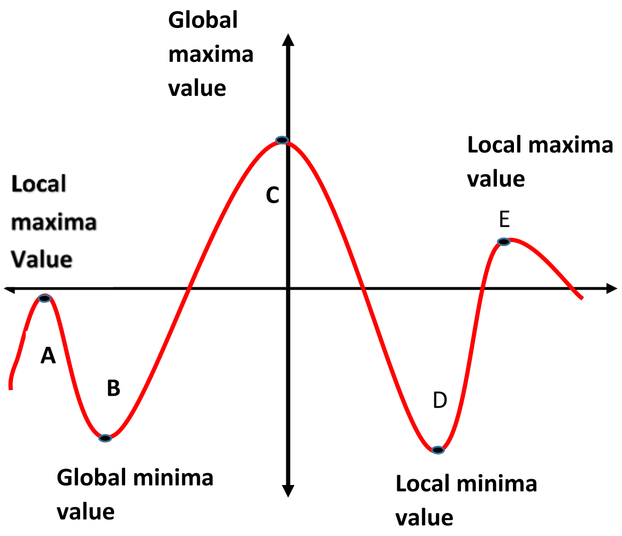


Figure 1.2 Local and Global minima.

1.3 COMPONENTS OF OPTIMIZATION PROBLEM

- *Objective function*: This is the main component of the optimization model and is a mathematical function which either needs to be minimized or maximized. For example, while designing any system/product the objective function can be: benefits of the product, cost related with the product, efforts required while designing the product. For any optimization problem, there can be one or more than one objective function related to it.
- *Decision variables:* These variables can take any value from the defined feasible range of the problem and the value of the objective function is controlled by these variables.
- *Constraints:* Constraints are basically the set of conditions which must be satisfied while designing a optimizing a problem. A collection of constraints puts some limit on the range of the decision variables i.e. they allow them to take on certain values only.

Thus the aim of optimization problem is either to maximize or minimize the objective function by finding the values of decision variables which satisfies the constraints. The mathematical form of this optimization problem can be defined as:

$$\text{Minimization of } F(x)\ ;\ x = [x_1, x_2, x_3, x_4 \ldots \ldots \ldots x_p] \qquad (1.1)$$

where $F(x)$ is objective function and x is decision variable. Constraints related with this problem are:

$$s_j(x) \leq 0\ ;\ \text{for } j = 1 \text{ to } m \qquad (1.2)$$

$$h_k(x) = 0\ ;\ \text{for } k = 1 \text{ to } n \qquad (1.3)$$

$$\text{and } x_{iL} \le x_i \le x_{iU} \text{ ; for } i = 1 \text{ to } p$$

where $s_j(x)$ and $h_k(x)$ are inequality and equality constraints respectively and p is number of decision variables.

1.4 CLASSIFICATION OF OPTIMIZATION PROBLEMS

Optimization problems can be categorized based on the type and number of function, variables and constraints. These classifications are briefly discussed as below:

1.4.1. Based on the existence of constraints

- *Constrained Optimization:* The problems which are subject to one or more constraints.
- *Unconstrained optimization*: These class of problems are not subjected to any type of constraint.

1.4.2. Based on the physical structure of problem

- *Optimal control problem:* This type of problem consist of mathematical programming problems containing number of stages and every next stage develop from previous stage in a given manner. This type of problem is controlled by two type of variables i.e. design and state variable.
- *Non-optimal control problem:* The problem which are not optimal, known as non-optimal problems.

1.4.3. Based on the nature of equation involved

- *Linear programming problem*: In this type of problem, the objective function and constraints of the problem are linear function of design variables.
- *Non-linear programing model:* If either the objective function or constraints is non-linear, then the problem is known as non-linear programming model.
- *Geometric programing model:* For this type of problem the objective function and constraints are expressed as polynomial of X. Application of geometric programming are electric circuit design, finance and statistics.
- *Quadratic programming model*: In this problem, the objective is to find the solution of quadratic objective function of several variables, subjected to the linear constraints.

1.4.4. Based on the value of decision variables

- *Integer programing model*: In this class of problem, the objective function is minimized or maximized by selecting only integer values of decision variables.
- *Real-value programing model:* For this class of problem, the decision variables take the real value from the allowed set.

1.4.5. Based on the deterministic nature of variables

Under this classification, problem can be of two type i.e. *deterministic and stochastic problem.* If all the design variables are deterministic in nature, then the problem is known as deterministic problem. The problem is stochastic in nature, if some or all the design variable are probabilistic in nature. For example, determining the life span of any structure is stochastic in nature because of the probabilistic value of load capacity and concrete strength.

1.4.6. Based on the separability of the functions

The optimization problems can be classified as *separable and non-separable* based on separability of objective function and constraints. A function is said to separable if it can be expressed as sum of *n*- individual functions and the optimization problem with such type of objective function are known as separable programming problems. If the objective function is non-separable then the problem with such type of problem is known as non- separable problems.

1.4.7. Based on the number of objective function

The problem can be classified as *single- objective and multi- objective programming model* based on the number of objective function required for a given problem. For a single-objective problem there is only one objective function which needs to minimize. In a multi-objective function, problem consist of more than one objective function and task is to minimize every function simultaneously.

1.5 OPTIMIZATION ALGORITHMS: (WAY TO SOLVE THE OPTIMIZATION PROBLEMS)

Optimization algorithms acts as tool for solving optimization problems and no single algorithm is best suited for every type of the problem [15]. Thus, many algorithms have been developed for solving different types of problems. These algorithms can be classified into two main broad categories:

- Traditional algorithms
- Non-traditional algorithms

1.5.1. Traditional algorithms

These algorithms make use of the differential calculus for finding the optimum solution (either maxima or minima) of the problem and hence there applicability is limited to continuous and differential function only. These traditional algorithms provides a good way to know the properties of minima and maxima points of a function and how an algorithm works iteratively to find the optimum solution of a problem. Traditional algorithms can be used for optimization of single variable and multi-variable problem and are classified into two main categories:

- Direct method: This method uses the objective function value for detecting the optimum solution of the problem. The typical type of direct method include uniform search, golden section search, and lattice search.
- Gradient method: Gradient method uses the derivative operator (first and/or second derivative) for detecting the optimal solution. Some of the important method under this class are Newton-Raphson method, Steep descent (Cauchy's) method.

The solution provided by these algorithm are also not optimum because the final value depends on the initial chosen solution.

1.5.2. Non-traditional algorithms

The traditional algorithms can stuck at some local optimum value depending upon the non-linearity and complexity of objective function. Hence, these algorithms are not found as efficient for providing the feasible solution. Thus non-traditional algorithms have been developed and are being used now a days to optimize the real world problems. These algorithms are also known as *Meta-heuristic algorithm*. These algorithms works most of time and quality solution of the problem can be found in reasonable time. It works on tradeoff between random search and local search. Some of the algorithms under this category are: genetic algorithm (GA) [6], differential evolution (DE) [7], bat algorithm (BA) [14], particle swarm algorithm (PSO) [10], flower pollination algorithm (FPA) [11]. According to No free lunch theorem (NFL) [15], no single algorithm is best suited for every type of problem. Thus, new algorithms are keep on developing to meet the requirement of the problem and/or to reduce the complexity and computational time of an algorithm. The detail study of these Meta-heuristic algorithms have been covered in the next section.

1.6 DETAIL STUDY OF META-HEURISTIC ALGORITHMS

Meta-heuristic algorithms are very popular now a days. These algorithms have proven their worth and dominated the field of optimization from the past two decades. These algorithms have been adopted by scientists and applied to number of different fields because of four main reasons *i.e. simplicity, flexibility, local optima avoidance and derivative free mechanism* which can be explained as [12]

Meta-heuristic algorithms are simple in nature because they are motivated from simple concepts like: social behavior of animals, physical phenomenon, and evolutionary nature of animals. This simple process of meta-heuristic algorithms motivate the scientists to learn it and to apply it on different problems. Simplicity of these algorithms also motivates the learners to simulate some operators and different nature's concept with these algorithms in order to propose some new algorithms.

Flexibility of the meta-heuristic means it can be applied to different type of problems without any modification in the structure of the algorithm. The one and only point, applicant need to do is to represent his problem according to the meta-heuristic algorithm.

Traditional techniques are deterministic in nature and easily got stagnated in local optima value and hence fail to provide global solution. But due to the stochastic nature of meta-heuristic algorithms, they have superior ability to get rid of local optima stagnation.

The optimization process of meta-heuristic algorithms starts with random solution and move in search space according to some nature's rule and there is no need to use the derivative operators to move in search space. Thus the mechanism of meta-heuristic algorithm is derivative free.

1.6.1 Components of meta-heuristic algorithms

Meta-heuristic algorithms [16] have two main components which are: *exploration and exploitation, [17, 18] or also termed as diversification and intensification*. Exploration is the process of visiting new regions of the search space as broadly as possible. In an algorithm exploration can be assured with proper use of operators in order to randomly and globally search the space. The exploitation phase focus on local search i.e. exploiting the neighborhood of promising region obtained by exploration process. The exploitation ensures that the algorithm converges to optimum value and exploration helps in avoiding the local optima stagnation. A good balance between these two components is a tuff task but it is must for the proper working of an algorithm [19, 20].

1.6.2 Classification of meta-heuristic algorithms

1.6.2.1 *Single solution vs population based algorithms*

In single solution based algorithms, the process starts with single solution and it focus on improving and modifying this single candidate solution only. Simulated annealing and iterated local search are examples of this class of algorithm.

In population based algorithms, the optimization process starts with the set of solutions (population) and this population is modified over the course of iterations for finding the optimal value.

1.6.2.2 *Nature inspired meta-heuristic algorithms*

These algorithms are inspired from nature and make use of nature's rule for finding the optimum solution. These algorithms are population based stochastic algorithms and hence their solution quality is very good when compared to conventional optimization techniques [22]. Another important feature of nature inspired optimization is that they don't require initial guess for solving any problem. These have been broadly classified into three main categories namely evolutionary algorithms (EA), physics based algorithms and swarm intelligence algorithms.

Evolutionary algorithms are usually inspired from the concept of evolution in nature. One of the most popular algorithm under this class is genetic algorithm (GA). This algorithm is based on the Darwinian theory of natural selection and was proposed by Holland [6]. Other algorithms under this class are differential evolution (DE) [7], genetic programming (GP) [23], evolutionary programming (EP) [24], evolution strategy (ES) [25, 26], moth flame optimization (MFO) [27], biogeography based optimization (BBO) [28].In EAs, initial solutions are generated randomly which are updated over subsequent generations by crossover and mutation. As per Darwinian concept, fittest solutions of the previous generation have more chance to participate in producing new solution in the next generation. So, best solution from the previous generation is used to generate new solutions in the next generation and over the course of iterations, initial random solution get optimized. A lot of work has been done in the past with evolutionary algorithms and brief summary of these algorithms is given in the Chapter 2.

Second category include Physics−based algorithm, which are different from the concept of the evolutionary algorithm. These algorithms mimic the rules of physics while optimizing initial solutions over the course of iterations. Some of the algorithms under this category are gravitational local search (GLS) [29], big bang big crunch (BBBC) [30], black hole (BH) [31] and water wave optimization [32]. Here general rules of physics such gravitational force, ray casting, weights and inertia force are used. These algorithms have adopted by many researchers to optimize their problems. A brief review of the work related with physic based algorithm is presented in Chapter 2.

Swarm intelligence (SI) based algorithm is third category of stochastic nature inspired algorithms, which was firstly proposed by Beni and Wang [33]. These algorithms are inspired from the collective behavior and social intelligence of swarms, flocks or herds [34]. PSO [10] is the most popular SI based algorithm in which new position is guided by personal best, global best and particle's current position. Other algorithms under this category include ant colony optimization (ACO) [8], bat algorithm (BA) [14], grey wolf optimization (GWO) [12], bat flower pollinator (BFP) [35]. This swarm based algorithm have number of advantages over the other classes of nature-inspired algorithm. Thus the detailed study of SI algorithm is presented in section 16.3.

1.6.3 Swarm Intelligence

In the recent years, the swarm intelligence has become a research interest of scientists of different field. The term SI was introduced and used by Gerardo Beni [33] in cellular robotic system. Later, SI was related with engineering field by Bonaneau *et al.* in 1999 [34] and is described as *"emergent collective intelligence of groups of simple agents"*. This collective intelligence of the group is helpful for solving the complex problem that is difficult to solve by single agent. This collective intelligence of the group comes from the social behavior of animals which is defined as a way of interaction between different members of a group while searching for food/prey.

SI works on two important properties of eusocial animals (who lives in colony) which are self-organization/ de-centralized system and division of labour and are explained as below:

- Self-organization is a mechanism, in which the structure at global level arises from the interaction of low-level component. This mechanism is spontaneous in nature and hence there is no need of control by any external agent. Bonabeau [34] have defined four different rules for self-organization: Positive feedback, Negative feedback, multiple interaction and fluctuations.
    i. Positive feedback: is the set of information from the output of system to the input in order to create the structure at global level. Positive feedback works in order to enhance the current results of the system. In term of algorithms, it can be explained by taking the example of artificial bee colony algorithm (ABC) [8]. In this algorithm, positive feedback is provided by bees in the form of dance in order to recruit more bees for the collection of food.
    ii. Negative feedback counterbalance the effect of positive feedback. In the ABC, the negative feedback is used to limit the competition or crowd among the bees for the same food source.
    iii. Fluctuation is a random walk of the swarm in order to generate a new solution. Randomness will promote the diversification and hence help the algorithm to get rid of local optima stagnation.
    iv. Multiple interactions allows the agents to learn from the results of their own activities and from other members of the group. Thus, multiple interaction

provides some form of communication. Pheromone markers in ACO is best example of this property.

- Division of labor: Inside a group there a various task to be performed. Generally, these tasks are divided among different members to be performed simultaneously and this phenomenon is known as division of labor. Swarm intelligence based algorithm works on this phenomenon and this simultaneous task provides better results as compared to sequential task.

These special features of swarm intelligence make them more suitable to play important role in engineering applications. These swarm intelligence is a boon to researchers over the traditional algorithms because of their effectiveness and robustness. These are effective because they can perform several task simultaneously that one robot cannot. Robustness can be explained in the sense that even if one member of the group fails, it can still provide the better result because of the collective behavior of group. Some of the advantages are of swarm approach are listed as below:

- Computational efficiency: The availability of multiple members, decreases the computational overhead and hence increase the efficiency.
- Reliability: Swarm approach performs consistently well because of group operation and de-centralized approach.
- Longevity: Swarm approach has long life period because of multiple agents outlive a single agent.
- Swarm approach has self-organizing ability.
- Swarm intelligence based approach require less hardware and hence cost is less.

From the above discussion it can be inferred that, SI is very interesting branch of the nature-inspired algorithms and also it has ability to produce the optimum results with high efficiency and good reliability. Thus in this thesis, work has been done with one of the swarm intelligence based algorithm i.e. grey wolf algorithm (GWO). This algorithm was proposed by Mirzalili *et al.* in 2014 [12] and it mimics the social behavior and hunting mechanism of grey wolves. In this algorithm, new solution is generated by evaluating the mean of the three best solutions from the entire population. The detail study of previously related work and working of GWO [12] is presented in Chapter 2 and 3 respectively. The brief review of work related with the other algorithms of this class of nature-inspired algorithm is also presented in the Chapter 2.

1.7 ANTENNA

Antenna is a transforming device that converts the electronic signal (propagating in RF transmitter/receiver) into electromagnetic signal (travels in free wave) and vice versa [36, 37]. In addition to transmitting and receiving waves, an antenna is also used to optimize the radiation energy in particular direction. Hence, antenna can also be used as directive device along with transmitting and receiving device. Thus to meet the specific need, antenna can take any form like, piece of conducting wire, a patch, an aperture, a lens, a reflector, an array of elements and so on. Antenna acts as one of the important component of the wireless communication system. In the communication system antenna is as important as the eyes and eye-glasses to the human body. A good and proper design of antenna can improve the overall system performance. For example in TV a broadband reception can be improved by using high performance antenna.

The performance of an antenna can be measured from the radiation pattern, which is the graphical representation of the radiation properties of the antenna with respect to space coordinates. These radiation properties include radiation intensity, power flux density, directivity and phase. A graph of received electric/ magnetic field at constant radius is called the amplitude *field pattern* and a graph of variation of power density along a constant radius is known as *power pattern*. These field and power pattern are normalized with respect to their maximum value. Power pattern is plotted in logarithmic scale (or in dB), as this scale can emphasize the low power value (known as minor lobes) in more detail. As radiation pattern is important factor to evaluate the performance of an antenna, the brief view of radiation pattern and the terms related with it which is required to study the antenna characteristic is presented in the next topic.

1.7.1 RADIATION PATTERN

The radiation pattern is divided into different lobes known as major, minor, side and back lobe. A radiation lobe is a portion of radiation pattern which is bounded by weak radiation intensity. Figure 1.3 illustrate the three 3-d radiation pattern and Figure 1.4 shows the linear 2-d pattern. A major lobe encloses the direction of maximum radiation intensity and all the other lobes except the major lobe are classified as minor lobes. The lobes which are just next to major lobe are known as side lobes (Figure 1.3 and 1.4). These lobes are normally largest of minor lobes and it represents the radiation intensity in undesired direction and hence they should be reduced. For example, in case of radar systems, low side lobe levels are very important in order to avoid false target detection through these side lobes.

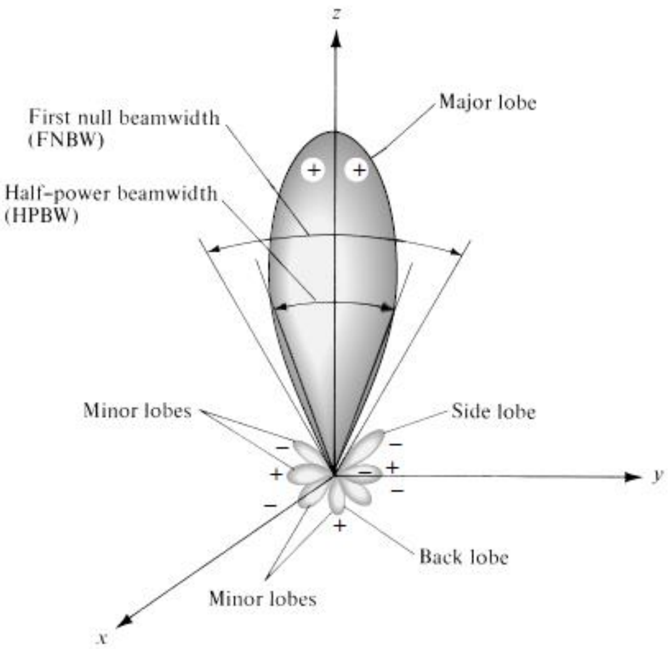


Figure 1.3 Three- dimensional radiation pattern [36]

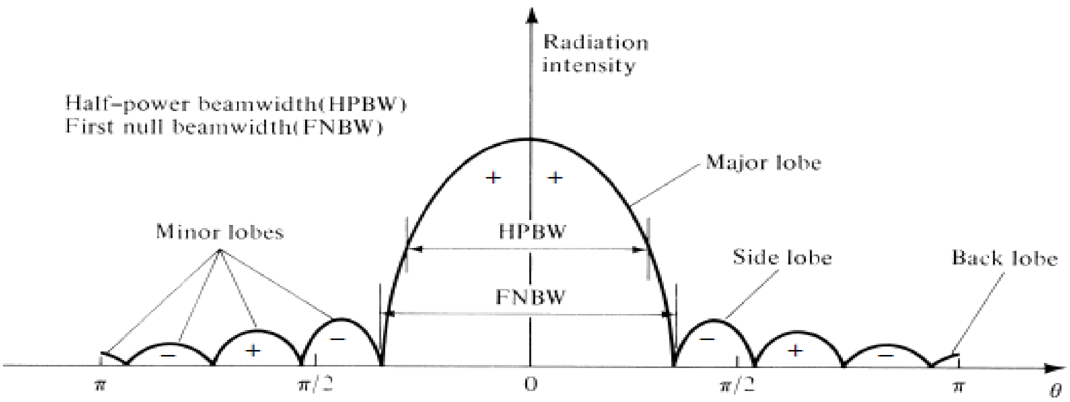


Figure 1.4 Two- dimensional view of radiation pattern [36]

A side lobe level of -20dB or less then this is generally desirable in practical applications. Another important parameter of radiation pattern is the beam width where first null beam width (FNBW) is the angular separation between the first null on the either side of the main lobe and the half power beam width (HPBW) is the angular separation between the half power (-3dB) point of the main lobe.

1.7.2 ANTENNA ARRAY

The antenna are used in all sort of wireless communication like radar communication, telecommunication and without antenna it is not possible to imagine these systems [38, 39]. For all these type of application, a single antenna need to be of very large antenna size so as to provide coverage to whole geographic area. But this large size antenna has disadvantage of high designing cost and it is difficult to shift this antenna from one place to another. Thus to overcome this problem, a large number of small antennas can be arranged in the space to form an antenna array with directive radiation pattern [36, 37]. The size of antenna array can be made extremely large in order to gather information from multiple point. Another point to note about the antenna is that, it provides low value of gain (directivity) and for many applications it is required to design antenna with high value of gain (i.e. directive characteristic). Again, this can be accomplished by using an assembly of radiating elements which is referred as antenna array. In most of the cases, the antenna array is formed with identical elements and the individual element can be of any form (wire, patch, apertures etc.). The total radiation pattern of an array is equal to the vector sum of the field radiated by the individual antennas. Thus, the arrangement of antennas in an array should be such that radiation intensity from the different elements adds up to provide the maximum radiation in particular direction and minimum in undesired direction. The geometrical arrangement of the antenna element in an array also affects its radiation pattern and based on the geometrical configuration of the elements array can be classified into: linear, circular and planar array.

Linear antenna array (LAA): This is the simplest and one of the most practical geometric configuration of antenna array in which elements are arranged in a line. This linear antenna array has gained attention of many researcher because of its geometrical simplicity.

Circular antenna array (CAA): Circular array is basically a 1-D linear array in which elements are arranged in the form of a circle instead of a line. All the elements are arranged in a circle of fixed radius. It provides a full 360 ° scan in horizontal direction.

Planar array: In this, antenna elements are arranged in a rectangular grid to form a planar/ rectangular array. In other words it is a set of linear antenna array placed in a parallel such that all the elements of every linear antenna array is equally spaced. Planar array has 360 ° view of the area, hence is widely used in communication, tracking radar, remote sensing and many others. These array have large aperture and can be used to provide a directional beam but it is a tuff task to synthesize these array because of their complexity and cost related with the synthesis is also high.

Random placement of the antenna elements in these array can cause problems like wastage of power, radiation pollution. Hence, the relevant performance and the desired shape of radiation pattern can be achieved by controlling any of the following parameters:

- By changing the geometrical configuration of an antenna array.
- The excitation amplitude of an individual element

- The phase of the excitation current provided to individual element
- The distance between the antenna elements
- The individual pattern.

Thus the sensible design of an array can be made to fulfill the desired objectives by controlling any of above parameters. This process is known as synthesis of antenna array and it can be done by using mathematical tool such as optimization algorithms. Synthesis of antenna array is difficult and non-linear problem. This process is difficult and non-linear. Nature inspired algorithms are preferred for the synthesis of these antenna array over conventional/ derivative method because the latter one suffers from the problem of local optima stagnation i.e. not able to provide the global optima value.

Generally, antenna array are designed with an aim to produce directive radiation pattern in specific direction and to minimize the side lobe level (SLL). In the array synthesis, this can be accomplished either by controlling the magnitude and phase of excitation current or by considering the distance between two consecutive elements [40, 41]. However, to provide the exact value of phase or magnitude of excitation current a complex electronic circuitry is required [38]. In many application, this type of complex control is not required so the desired direction of main beam can be controlled by varying the distance between the elements and keeping the constant value of excitation current. Such antenna arrays are known as non-uniform array. In the linear antenna array all the elements are arranged in straight line. A number of stochastic algorithms have been used to design the both uniform and non-uniform linear array. In this thesis the synthesis of non-uniform linear antenna array has been done by using the new variant GWO algorithm. The background of the synthesis if LAA is discussed in Chapter2 and its synthesis using new variant of GWO is presented in Chapter 4.

1.8 OBJECTIVES OF THESIS

- To design an enhanced grey wolf optimization (EGWO) algorithm with an improved performance to find the optimum value of the problem.
- To design an extended version of grey wolf optimization (GWO-E) with high convergence speed and better explorative ability.
- Implementing EGWO and GWO-E on standard benchmark functions in order to prove their effectivity over other state-of-the-art algorithms.
- Implementing extended version of grey wolf optimization (GWO-E) for the synthesis of linear antenna array with an aim to reduce the side lobe level and null positioning.

1.9 ORGANIZATION OF THESIS

The rest of the thesis is organized into following chapters and the pictorial representation of this is presented in Figure 1.5.

Chapter 2: It provides the literature survey required for achieving the objective. Firstly, brief literature review of some of the nature inspired algorithms and detailed study of previous work of GWO are presented. Secondly, it covers the detail of the qualitative work that exist in the literature for the synthesis of linear antenna array.

Chapter 3: This chapter, firstly illustrates the basic concept and working of the original algorithm (i.e. GWO). Then, the concept of the proposed approaches has been discussed.

Chapter 4: It provides the detail of the linear antenna array problem.

Chapter 5: In this chapter, the proposed approaches of the chapter 4 are applied on standard benchmark problems to test their performance. The related results are presented in the tables and their convergence plots are shown in figures. Then, linear antenna array is synthesized using one of the proposed approach and the results are compared with other work that exists in the literature.

Chapter 6: The conclusion and future scope of the thesis is presented in this chapter.

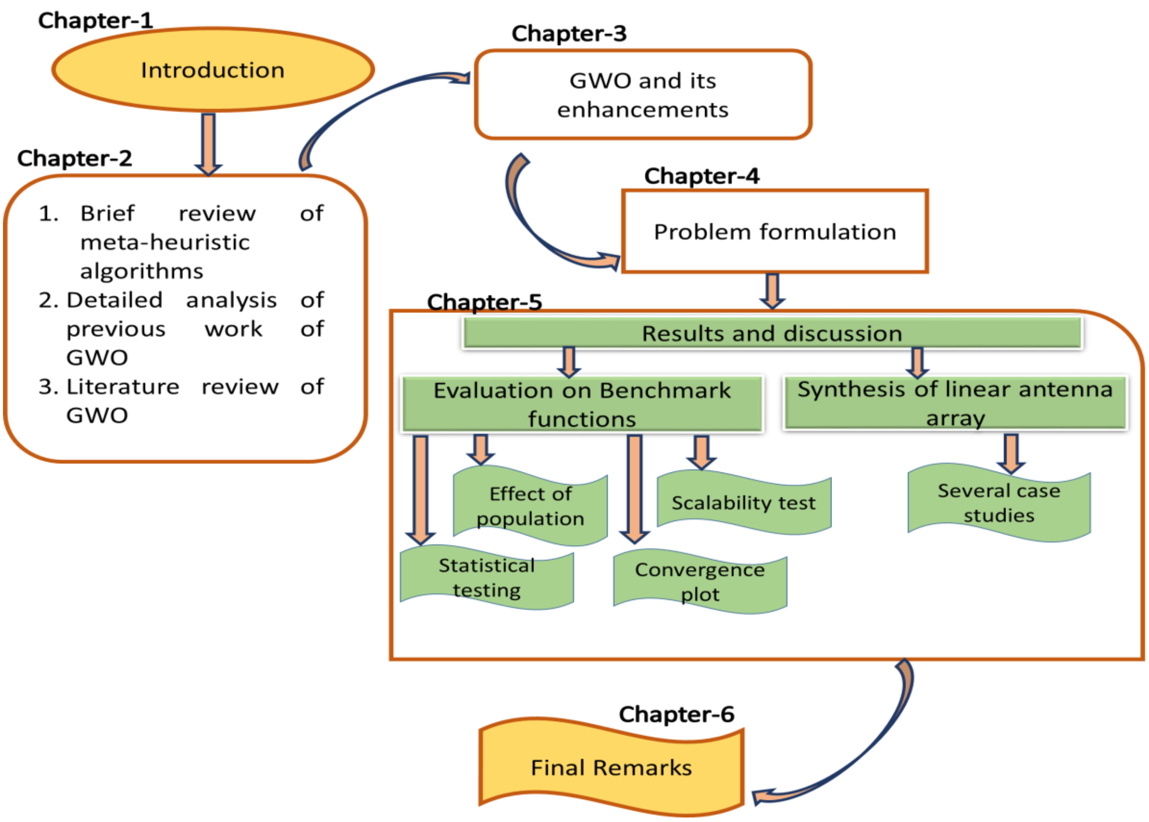


Figure 1.5 Thesis outline

CHAPTER 2
LITERATURE SURVEY

2.1 LITERATURE REVIEW OF NATURE INSPIRED ALGORITHMS

Evolutionary algorithms are stochastic population based algorithm which are inspired from the concept of evolution in nature. Genetic algorithm is one of the most popular algorithm under this class, which was proposed by Holland [6].This algorithm is based on the Darwinian theory of natural selection. In this algorithm, fittest population of the current generation undergo genetic operation (crossover and mutation) to form next generation population. The Fittest population from the current generation is selected using some selection strategies like: roulette wheel selection, ranking selection, tournament selection and so on. The two random solutions from the fittest population are mated through the process of crossover, in which next generation is inherited from either of the parent. In addition to this, some offspring undergo mutation process, in which their randomly selected one gene (dimension of population/chromosome) changes by some random amount. After several generation, initial solution gets optimized and hence the algorithm converges to the best solution. The use of GA for engineering applications was found by Goldberg [42]. After that, GA have been used to solve the various real world problems like scheduling problem [43], neural network problem [44], antenna design problem [45], pattern classification problems [46] and clustering problem [47]. This algorithm has proven to be flexible, robust and efficient for vast complex spaces but it takes too much time to solve the problem and also lacks in diversity. The speed of GA can be enhanced by using GA in parallel structure [48]. Another example of EA is flower pollination algorithm (FPA). This algorithm was proposed by Yang [11] and is inspired from the pollination process of the flowers. In this algorithm, the initial random generated population gets optimized over the course of iteration either by global pollination or local pollination (as inspired from the cross-pollination and self-pollination of the flowers). The switching between the global and local pollination is controlled by switching probability. There are many researchers who have adopted this algorithm and used it for real world problem. FPA was used with wireless and sensor network global optimization [49], retinal vessal segmentation [50], minimization of weight of truss structure [51],fractal image compression [52], economic and emission dispatch problem [53], planar graph coloring problem [54], synthesis of circular array antenna [55]. Binary version of FPA has used for the feature selection problem [56].

Physics−based algorithm, mimic the rules of physics while optimizing initial solutions over the course of iterations. Black hole [31] is one of the physics based algorithm which is inspired from the black hole phenomenon. It starts by initializing a random population for the problem under test and after certain iterations, the best particle is generated which acts as the black hole. This best particle will attract all other particles from the population around it. The particle which gets too closer to the black hole will be swallowed by it and new random particle will be generated at new position in the search space. This algorithm has been used to solve many real world problems like medical data classification [57], image grey level enhancement [58], big data analysis [59], optimal power flow [60]. A binary version of the black hole algorithm has been proposed by Alvaro *et al.* [61] for solving the set covering problem [61]. BH algorithm has also used in the real time application of cloud computing for the prediction of work load on the cloud [62]. Black hole algorithm has also modified by some researchers to enhance its performance, like the concept of inertia weight has been embedded in the black hole algorithm for enhancing its global search capability [63]. The another algorithm of physics based algorithm is big bang big crunch algorithm and is based on big bang and big crunch theories. This algorithm use the principle of big bang theory to scatter the search agents from one point in the space. This is done in order to explore the search agents and to the best solution. Then the search agents gathers at best point achieved so far by following the rule of big crunch theory. The gravitational search algorithm (GSA) [64] mimics gravitational force to optimize the solutions. In this algorithm search agents are collection of masses and these masses attract each other based on newton gravity. Heavy mass corresponds to the best solution of the problem and it has the bigger attractive forces. Thus, over the course of iteration, the heavier mass (global solution) attracts the other masses by a gravitational force which is proportional to their distance. These algorithms mimics the physics based rule and hence it includes the complex equation. Thus, researchers found it as difficult to understand and to adapt these techniques as per their requirement.

Swarm intelligence based algorithms are most popular algorithms from the last some years and is inspired from the collective behavior and social intelligence of swarms, flocks or herds [34]. Some of the most popular SI algorithms are Particle swarm optimization (PSO) [10], ant colony optimization (ACO) [8], artificial bee colony (ABC) [9], cuckoo Search, bat algorithm (BA) [14], flower pollination algorithm (FPA) [11], and grey wolf algorithms (GWO) [12]. In this section, the concept of some of the algorithms and their related work have been discussed. Particle swarm optimization was introduced by Kennedy and Eberhart and it mimics the social behavior of birds flocking. It uses a population of different particles (candidate solution) in order to find the best solution of the problem. In the optimization process, each particle updates their position w.r.t. current position, current velocity, personal best and global best. The complete process of PSO includes: (1) population/particle initialization, (2) velocity updating,

(3) to update the particle position, (4) memory updating and then (5) checking of termination criteria. A series of works has been done on the development of PSO since it was introduced. A section brief review regarding the modified PSO and application of PSO to various domain is being discussed here.

The continuous model of PSO (CPS) was proposed by Emara and Fattah [65] in order to increase the performance of the algorithm. In this continuous model, the differential equations were used for updating the current position of the particles. Various modification to PSO was proposed by using different inertial weight strategies. The concept of inertial weight was proposed by Shi and Eberhart in [66]. In this they studied the effect of different values of weight on the performance of PSO and suggested that any constant value from the range [0.8 1.2] ensures the convergence of the algorithm to good solution. In [67], a random value of inertia weight from the range [0.5 1] is used to track the optimum value from the search space. Apart from this, time varying inertial weight strategies has been proposed by many researchers. PSO was embedded with a linear decreasing inertial weight, to control the extent of exploration and exploitation. Apart from this, exponential decreasing function, sigmoidal decreasing function, logarithmic decreasing function, oscillatory function has been used by different researchers to control the value of weight A PSO-DVM was developed by Shi in 2001[68] with dynamic range of velocity. In this, the value of V_{max} was made to decrease with time. In the initial iteration, the large scale of the velocity generates the particles at diversified location, hence promotes exploration. As the process continues, the scale is reduced in order to find the better value of solution (i.e. exploitation). Another modification to PSO is known as guaranteed convergence PSO (GCPSO) [69]. As in PSO, the particle updates its velocity according to the according to global best position and personal best position. For a particle at global best position (which is also a personal best position) the change in velocity will be zero and hence the particle stagnates. Thus, GCPSO was proposed to overcome this drawback of the PSO. In this, the global best particle updates its velocity with different principal, while the other particles follows the original updating principle. MPSO-TVAC [70] is the modified version of PSO and it consist of two improvements i.e. mutation operator and time varying acceleration coefficient (TVAC). In this method, the stagnated particle changes its location by mutating it with random particle in order to avoid the problem of stagnation. Also the time varying inertia weight has been used to control the global and local search. PSO with controllable random exploration velocity (PSO-CREV) is modified PSO structure. In this, original updating principle is modified with an additional decreasing coefficient and a random velocity. This improvement enhances the explorative ability and convergence rate of the PSO. This algorithm is popular over the evolutionary and physics based algorithm because it has few parameters to control and hence it has been applied to various real- world problems like Knapsack and feature selection problem [71], for neural network application [72], for image segmentation [73, 74], image superposition [75], solar photovoltaic system [76].

The other examples of SI is ant colony optimization (ACO) [8] which was proposed by Dorigo *et al.* in 2006. This algorithm follows the metaphor of social behavior of ants. In fact, social intelligence of ants to find the shortest path between the food and the nest is the main inspiration of ACO. While searching for food, ants move randomly in search space and leave pheromones trails in their searching path. These trails helps the other ants to find food and hence they stop wandering on other paths. With passage of time, the pheromones trails get evaporated unless more ants travel on same path to lay down more pheromone. This behavior also helps to find shortest distance between food and ants. These characteristics of ants are exploited and used for designing of ACO algorithm to solve complex engineering problems. ACO has been successfully applied for solving problems like scheduling problem [77], graph coloring problem [78], image segmentation problem [79] and routing problem [80]. ACO has proven its worth, by showing efficient and suitable results for benchmark functions and engineering tasks. According to No free lunch theorem (NFL), no algorithm is best suited for every type of the problem. Thus, there are many SI algorithms proposed so far which are inspired from social and hunting behavior of animals. Grey wolf algorithm is new to SI algorithms which is proposed by Mirjalili *et al.* [12]. This algorithm mimics the hunting mechanism and leadership hierarchy of grey wolves. In this algorithm, new solution is generated by evaluating the mean of the three best solutions from the entire population which is the main cause for its better exploitative capability. While testing it performance on standard benchmark functions, this algorithm gave better results as compared with other well-known nature inspired algorithms. Hence deriving a new algorithm from this by removing its drawbacks, will be beneficial for many researchers. Thus, in this thesis GWO algorithm is explored in detail and the related study of this algorithm is presented in chapter 2 and 3. The detailed analysis of previous work of GWO is discussed below. This analysis covers almost all the previous paper of GWO that exists in the literature.

2.2 QUALITATIVE ANALYSIS OF PREVIOUS WORK OF GWO

Since its inception, GWO is found to be a very effective algorithm and has proved its worth. This can be noted from the fact that it has been applied to a wide range of applications. This section details about the brief literature review regarding the modified GWO algorithms and the application of GWO to various domains. The various publications till date are analyzed according to two main criteria:

1. Benchmarking or applications used by the authors in their paper.
2. The way used by the authors to validate their results (comparative study/statistical testing or scalability testing).

2.2.1 GWO with Modifications

Article 1 [81]*:* In this article, authors have described the evolutionary population dynamics (EPD) [82] and used this concept with GWO. EPD is a process of removing worst agents from the population instead of evolving around the best agents. Here half of the population (worst population) was re-located around Alpha, Beta, and Delta at random positions with equal probability. The whole process was done to enhance the exploration rate and convergence rate of GWO algorithm.

Inference: EPD-GWO was evaluated on six unimodal and seven multimodal benchmark functions (standard benchmarks functions were used [83]). Authors claimed that EPD-GWO has potential of improving the exploration, exploitation and local optima avoidance over the GWO. But from the comparison table, it was found that the experimental results for both the algorithms are comparable. No such analysis was presented which shows the improvement in the algorithm. Also, the authors didn't present any qualitative or quantitative analysis with respect to other state-of-the-art algorithms.

Article 2 [84]*:* Dynamic adaptation of parameters of GWO with fuzzy logic was given in this paper. Authors discussed about the effect of different ranges of the adaptive parameters namely $\vec{a}$ and $\vec{C}$ on the performance of GWO. Adaptation of the parameter in the proposed range was done either by increasing or decreasing fuzzy rules. Further it was found that the performance of GWO was best when '*a'* follows the range [0.5, 2.5] with decreasing fuzzy rules and *'c'* follows the range [0, 0.5] with increasing fuzzy rules.

Inference: In this case, seven benchmarking functions were considered to study the effect of parameters on the performance of GWO. Along with this, Z-test were provided to study the effect of best chosen range of the parameters (*'a'* and '*c'*) on the GWO at dimension size of 30, 60,128. But they failed to analyze, which parameters have greatest effect on the algorithm. Also, adaptation of these parameters with fuzzy logic increases computational time.

Article 3 [85]: A binary version of GWO was provided in this article. This new version of GWO was used for feature selection task. Authors gave two different methods for binary version and used them for feature subset extraction with an aim to minimize the number of selected subsets and to maximize the accuracy in feature selection process. In the first method (bGWO1), binary steps for alpha, beta and delta were designed and then stochastic crossover was used to find the binary updated position of the grey wolf. In the second method (bGWO2), the binary solutions were generated by mapping with the sigmoidal function.

Inference: Results of binary version of GWO was compared with PSO [86] and GA [87] with three different initialization methods, namely random initialization, small initialization and large initialization. This approach was evaluated over 18 different datasets of the UCI repository. Statistical testing of bGWO1 and bGWO2 was performed to evaluate the significant difference with GA and PSO. It was found that the performance of BGWO was better than the other methods for these set of problems.

Article 4 [88]: In this paper, GWO was hybridized with pattern search (PS) algorithm for minimizing the risk of blackout in the smart grid power system management. As security in power system planning is a complex and multi-objective problem, it has many objective functions to optimize the settings of various control parameters. GWO-PS based algorithm was used to maximize the marginal stability, loading factor of the critical load and to minimize the fuel cost, power loss, and voltage deviation. In this method, PS was used as local search mechanism with the GWO. To avoid blackout in critical situation, best particle was simulated to change in local search depending upon the information of the PS. The concept of PS was hybridized with GWO to make proper balance between exploration and exploitation in the process.

Inference: To check the efficiency of the proposed approach, it was applied to IEEE 30-Bus test system and a comparison was made with other state-of- the-art algorithms. It was found that GWO-PS provides better security to power system under critical situations. But no work was done to prove its effectiveness over original GWO and also no experiment study was presented to claim the better balance between exploration and exploitation than GWO.

Article 5 [89]: Hybridization of GWO with DE was done by Zhu *et al*. and they used this algorithm to solve the problem of test scheduling for 3D stacked system on–chip (SOC). Because of the strong searching ability of DE, it was combined with GWO to avoid local optima stagnation and to enhance the exploration ability. Mutation, crossover and selection operator of DE were incorporated in GWO to find the position of alpha, beta and delta members.

Inference: GWO-DE was tested on 23 benchmarks functions and one NP-hard problem. Results were taken for 30 runs with population size of 30 and iteration size of 500. GWO-DE achieved better results as compared with GWO [89], PSO [89] and DE [89]. Exploration and exploitation of the proposed algorithm were tested by using multimodal and unimodal functions respectively. In this paper, author does not provide any statistical analysis of the work.

Article 6 [90]: A new version of original GWO, termed as grouped grey wolf optimization (GGWO) was presented in this paper. This approach was applied to optimize the parameters of proportional integral controller of Doubly-fed induction generator (DFIG) based wind turbine. In this algorithm, population of grey wolves was divided into two sets namely hunting group and random scout group. First group performs the hunting mechanism as described in GWO. Second group is responsible for random global search and it makes proper balance between exploration and exploitation. Three cases of DFIG based wind turbine have been undertaken to evaluate the performance of proposed approach and comparison was performed with respect to GA [91], PSO [92], GWO [90] and MFO [90].
Inference: The addition of random scout group with GWO results in enhanced exploration and deep exploitation. Thus, GGWO can effectively avoid the problem of local optima stagnation. Further statistical analysis was also presented to validate the results.

Article 7 [93]*:* In this article, authors presented an improved GWO (IGWO) in which the control parameter $\vec{A}$ was used to control the extent of exploitation and exploration. For exploitation, original GWO was employed whereas for enhancing the exploration, omega wolves were guided towards some randomly chosen agents. Performance of this algorithm was tested for training q-Gaussian radial basis functional-link nets (q-RBFLNs) neural networks and comparison was performed with respect to PSO-qRBFLN, GSA-qRBFLN, TLBO-qRBFLN, GWO-qRBFLN.
Inference: This IGWO was quite helpful for training the qRBFLNs. Experimental and statistical results were presented in the paper to claim the effectiveness of the proposed approach but the results obtained were quite comparable to original GWO.

Article 8 [94]: Chandra *et al.* used new modified GWO (MGWO) [95] for optimal web service selection with global constraint. In the MGWO, crossover operator was embedded with GWO. After updating the population of the grey wolves, crossover between two randomly selected agents was performed and fittest among them was passed to next generation. This modified approach was applied for optimal web service selection with global constraint and compared with GA [96] and GWO [94].
Inference: The crossover operator of GA allowed the wolves to share information with other group mates and helped them in increasing diversity. This increased diversity helped the algorithm in overcoming local optima problem of GWO. In this case, the performance of the

MGWO for web services was evaluated by computing the availability, response time, throughput and successability but the authors does not provided any statistical testing.

2.2.2 Direct Applications with GWO

Besides the modifications of GWO, there are many authors who have adopted the original grey wolf optimization and used it for solving real world problems. The detail of all these type of work is given as:

- GWO was used for automatic generation control in the multi-area ST-thermal power system [97]. In this, authors used GWO to optimize the PID controller and found that it performed better in terms of settling time and magnitude of oscillation.
- GWO has been used with fuzzy PID controller for automatic generation control in hydro-thermal power system by [98]. Another application in which GWO was employed for optimizing the parameters of PID controller is DC motor [99].
- GWO was employed for extracting the essential oil from Cleome coluteoides [100] for optimizing the values of parameters such as temperature, pressure, particle size and extraction time.
- Original GWO was used along with maximum power point design for photovoltaic cell [101], flow shop scheduling [102], surface wave analysis [103], unmanned combat aerial vehicle(UCAV) [104].
- Support vector machine (SVM) is widely used for the classification of the different classes but setting the values of SVM parameters is a big challenge. Thus, GWO was hybridized with SVM to optimize its parameters for making efficient classification algorithm [105].
- One of the application which make use of GWO is band selection for hyperspectral images [106]. These hyperspectral images consist of hundreds of bands. GWO was used for selecting relevant band without reducing the accuracy rate. This was done in order to reduce the dimensionality of the images.
- GWO has been used for training the multi-layer perceptron [107], to control the load frequency of interconnected power system [108], for combined heat and power dispatch problem for allocating the generation and outputs among different units [109], in distribution system for allocating multiple distribution generation so as to minimize the power losses and maximize the voltage profile [110], to deal with the problem of non-convex economic load dispatch [111], in optimal reactive power dispatch problem to set the best values of the control parameters in order to reduce the loss and deviation in voltage [112], for optimization of the photonic crystal filters [113], for designing the power system stabilizer [114], in forecasting model in order to predict the commodity prices [115], with

fuzzy control system in order to reduce the parametric sensitivity and computational cost [116].

Inference drawn:

This analysis of previous work of GWO, helps us to figure out its drawback. Hence, it can be inferred that the original GWO suffers from the problem of stagnation in local optima. This is because of the lack of diversity among search agents and poor exploration of GWO. Although steps have been taken by many authors in the past two years to enhance the performance of GWO but they failed to provide either proper explanation or statistical results of their work to show the improvement in their proposed work. This motivated us to provide new version of GWO in order to improve the performance of the original GWO. Detail of this modification is given in Chapter 4.

2.3 LINEAR ANTENNA ARRAYS

In the linear antenna array all the elements are arranged in straight line. Many stochastic algorithms have been used to design the both uniform and non-uniform linear array. In the uniform array, the meta-heuristic algorithms are used to find the exact value of magnitude/phase of the excitation current while keeping the uniform distance between the elements of array. The PSO [117, 41], ABC [118], GA [40] have been put in use to design uniform linear array. Designing of Non-uniform spaced linear array (NUSLA) also exist in literature. The first article regarding the concept of NUSLA was published in 1961 [119]. In this, designing was done by perturbation method and main focus was on the side lobe reduction. Hodjat and Hovanessian [120] developed an iterative method for side lobe reduction. Fourier transform has also been used for the designing NUSLA [121]. These methods are not so flexible for synthesizing any random radiation pattern.

Meta-heuristic algorithms provides flexibility to easily synthesize the radiation pattern of NUSLA. DE [122] and GA [123,124] have been used to reduce the SLL of NUSLA [123-125]. PSO has also been used in literature to synthesize the antenna array with reduced SLL and nulls in particular direction [126]. Simulated annealing (SA) [127] and harmony search (HS) [128] have been used for the synthesis of unequally spaced linear array. The use of ACO has been found in literature for designing the non- uniform antenna array [129]. FPA has been used by many researchers to optimize the linear antenna array by using different case studies [130, 131].Biogeography based optimization algorithm (BBO) has been used for designing of linear antenna array with an aim to reduce the SLL and null placement in desired direction [132, 133]. Various other algorithms like Tabu search (TS) [134], ant lion optimization (ALO) [135] and spider monkey optimization (SMO) [136] have been used for the designing of linear antenna array. The cuckoo search algorithm have been applied by Singh and Rattan [137] for the synthesis of linear antenna. GWO algorithms have used by Saxena and Kothari [138] to optimize the radiation pattern of linear antenna array. In this case several different cases were used to evaluate the performance of GWO. Other algorithms that exists in literature for the synthesis of LAA are Gravitational search algorithm (GSA) [139], Backtracking search optimization algorithm (BSA) [140], cat swarm optimization (CSO) [141].Besides these basic optimization algorithms, the use of their modified version have also been reported in literature. Oraizi and Fallahpour has solved the problem of NUSLA design by the combination of GA and conjugate gradient method (GA-CG) [142]. Multi-objective version of DE (MODE) [143] has been applied for the synthesis of linear antenna. Goudos *et al.* have used a comprehensive particle swarm optimization method for the synthesis of linear antenna array for null placement, SLL reduction and beam width control [144]. Cen *et al.* have used improved genetic algorithm for the synthesis of aperiodic linear antenna array [145]. The synthesis of LAA have also done by fitness adaptive differential evolution (FADE) [146] and enhanced ant lion optimization (e-ALO) [147], comprehensive learning particle swarm optimizer [148], invasive weed optimization (IWO) [149], modified invasive weed optimization [150], mean variance mapping method [151]. In this thesis GWO has been modified and used for the optimization of radiation pattern of linear antenna. The performance of the proposed approach is evaluated by comparing its results with those available in literature and results are shown in Chapter-5.

From the above discussion, it has observed that meta-heuristic algorithms helps in producing the cost-effective designs. There are various meta-heuristic algorithms that exist in literature which are different from each other based on their procedure of working. SI is one of the interesting branch of population based meta-heuristic algorithm as it is de-centralized process. These algorithms are computationally efficient and reliable because of availability of multiple members. These techniques have been applied to various real world problems and yields good results. These algorithms are found as very effective for solving various optimization problems but are not so efficient for solving every optimization problem [15]. Hence there is need to study these algorithms and to develop a valuable algorithm that can produce effective and better results for the real world optimization problems. Grey wolf optimization (GWO) algorithm is the newest of SI technique which is inspired from the social behavior of grey wolf. This algorithm has been used to solve various mathematical optimization problems and it has proved its worth by providing excellent results on standard benchmark functions and on real world applications. GWO algorithm does need any complex operators to solve the optimization problems and also it is computationally inexpensive in terms of memory required for an algorithm to run and has small value of runtime as compared with other algorithms. This make

GWO, a valuable algorithm that can obtain optimum results for design/optimization problem. But from the literature review of GWO, it can be concluded that but it suffers from local optima stagnation. This motivated us to do work with GWO in order to enhance its performance and to develop a new algorithm from it, which will be beneficial for researchers to optimize their own related work.

Also the antenna arrays are important parts of today's communication system and they are commonly used in mobile, radar and wireless communication. They can be designed to produce the desired characteristics in radiation pattern and this process is known as synthesis of array. Synthesis of linear antenna array is classical electromagnetic problem and it has been done by various existing nature inspired algorithms. The GWO [146] has also yielded good results in reducing the SLL and null placement. It prompted us to apply a modified version of GWO on this problem in order to check its performance on real world applications and to evaluate its performance by comparing it with the work that already exists in the literature.

CHAPTER 3
GWO AND ITS ENHANCEMENTS

This chapter, firstly covers the basics of the grey wolf optimization algorithm and then detail of the proposed approaches is presented. These are discussed as below:

3.1 BASIC OF GREY WOLF OPTIMIZATION

GWO [12] algorithm is inspired from the social behavior and hunting mechanism of the grey wolves. These animals live in groups and in each group, the population is divided into four levels. Top most position in a group is occupied by the alpha wolf followed by beta, then delta wolves and the lowest level is occupied by omega wolves. Alpha wolf is also known as dominant wolf because its order is followed by entire group. Beta wolf on the other hand, act as subordinate to the alpha wolf, while delta wolf acts as subordinate to beta and dominant over the omega wolves. The main phases of group hunting of grey wolf are as follows:

- Encircling
- Hunting
- Attacking the prey

3.1.1 *Social hierarchy*

While implementing GWO algorithm, there is need to model the social hierarchy of grey wolf. In every iteration, fittest solution among the entire set of population is consider as alpha (α) followed by beta (β) and delta (δ). Rest of the solutions come under the category of omega (ω) wolves. Top three fittest element will guide the omega wolves towards the position of prey.

3.1.2 *Encircling prey*

The first phase of hunting mechanism is encircling the prey. Here the position of grey wolf is updated inside the search space around the prey by using equations (3.1) and (3.2). Mathematical equations for this phase are given as:

$$\vec{D} = |\vec{C}.\overrightarrow{Xp}(t) - \vec{X}(t)| \quad (3.1)$$

$$\vec{X}(t+1) = \overrightarrow{X\mathrm{p}}(t) - \vec{A}.\vec{D} \quad (3.2)$$

where *t* denotes the current iteration, $\vec{A}$ and $\vec{C}$ are coefficient vectors, $\overrightarrow{Xp}$(t) is vector which denotes the position of prey and $\vec{X}$(t) is vector which denotes the position of grey wolf. The value of vectors $\overrightarrow{A}$ and $\vec{C}$ are calculated using equations (3.3) and (3.4)

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \quad (3.3)$$

$$\vec{C} = 2.\vec{r_2} \tag{3.4}$$

In this element of $\vec{a}$ are linearly decreased from 2 to 0 over the course of iteration, $\vec{r_1}$ and $\vec{r_2}$ are random vectors in the range of [0 1].

3.1.3 *Hunting*

The top three fittest members from the population of grey wolf i.e. alpha, beta and delta have good knowledge regarding the position of prey. Hence the positions of other remaining wolves (omega wolves) are changed with respect to the value of alpha, beta and delta (as dictated in Figure 3.1). The following equations are used in this regard.

$$D_\alpha = |\vec{C_1}.\vec{X_\alpha} - \vec{X}| \tag{3.5}$$

$$D_\beta = |\vec{C_2}.\vec{X_\beta} - \vec{X}| \tag{3.6}$$

$$D_\delta = |\vec{C_3}.\vec{X_\delta} - \vec{X}| \tag{3.7}$$

$$\vec{X_1} = \vec{X_\alpha} - \vec{A_1}.(\vec{D}_\alpha) \tag{3.8}$$

$$\vec{X_2} = \vec{X_\beta} - \vec{A_2}.(\vec{D}_\beta) \tag{3.9}$$

$$\vec{X_3} = \vec{X_\delta} - \vec{A_3}.(\vec{D}_\delta) \tag{3.10}$$

$$X(t+1) = \overrightarrow{(X_1} + \vec{X_2} + \overrightarrow{X_3)}./3 \tag{3.11}$$

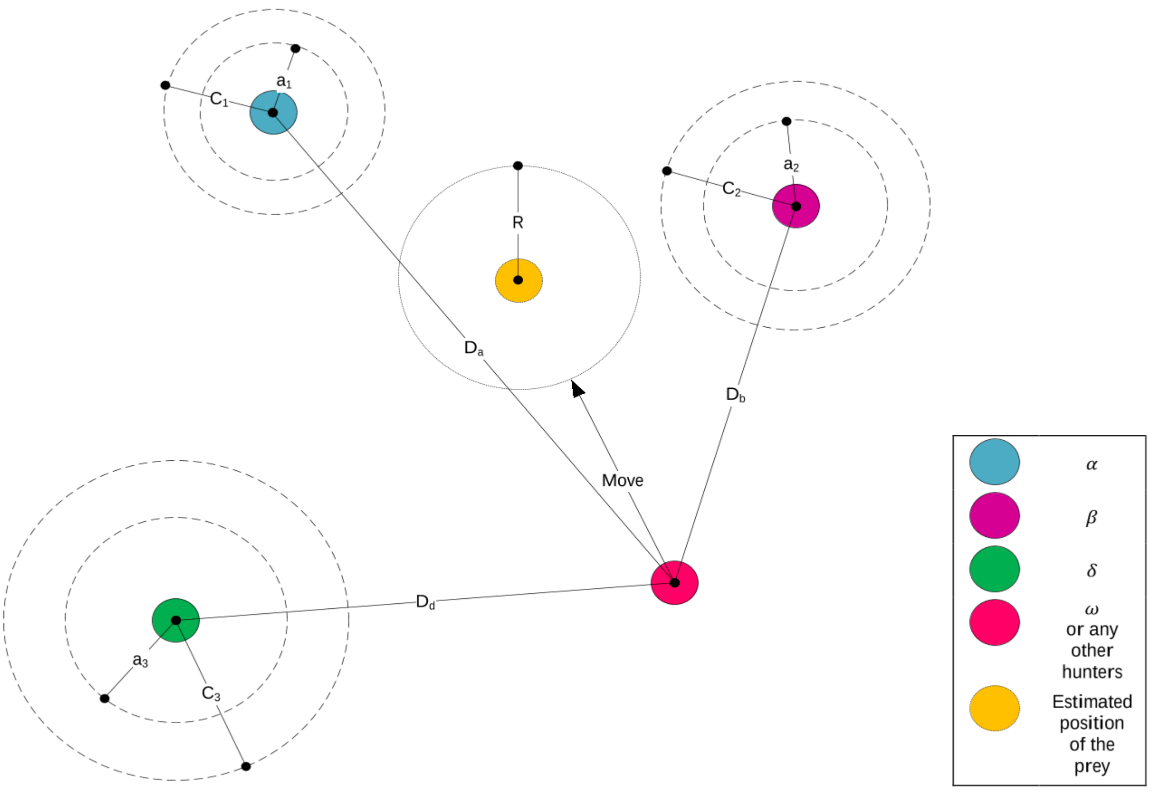


Figure 3.1 Updating Positions of omega (ω) wolf [12]

3.1.4 *Attacking the prey (Exploitation)*

During hunting mechanism, value of $\vec{a}$ reduces from 2 to 0 and thereby the value of $\vec{A}$ also decreases (as per the relation given in equation (3.3)). When the value of $\vec{A}$ lies between the range $[-1, 1]$, step size provided will be very small and hence next position of the agent will be very close to the prey's position. It means for this range of $\vec{A}$, wolves move towards the prey (as shown in Figure 3.2).

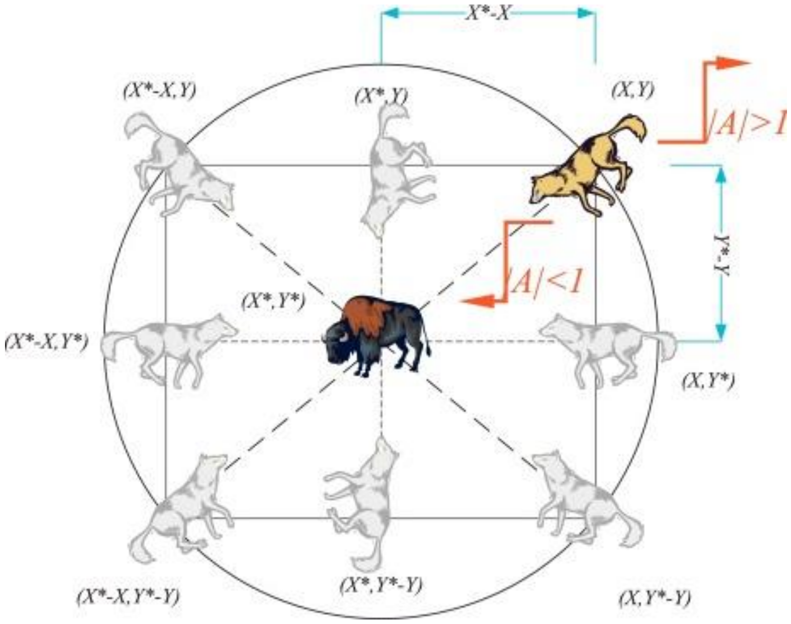


Figure 3.2 Behavior of wolves [12]

3.1.5 *Search for prey (Exploration)*

With the aim to find new solution (prey), grey wolves separate from each other and unites together to attack the prey and this is achieved in GWO by the use of parameter $\overrightarrow{A}$. When the value of $\vec{A}$ is less than -1 or greater than 1, it provides large value of step size and hence the new position of the search agents will be at large distance from the prey. Overall this range of $\overrightarrow{A}$ ($|\overrightarrow{A}| > 1$), make the wolves to move away from each other in order to search a new prey (as shown in Figure 3.2). There is another parameter $\vec{C}$ which helps in controlling the exploration of search agents and also avoids stagnation in local optima by providing random weight to the prey and thus enhancing the effect of prey for $\vec{C}>1$ and diminishing its effect for $\vec{C}<1$.

GWO starts by initializing random population of grey wolves. In every iteration, top three fittest members (alpha, beta and delta) are selected to guide the remaining wolves towards the position of prey. During the hunting mechanism value of $\vec{a}$ is linearly decreased from 2 to 0 and so the range of $\vec{A}$ decreases. When $|\vec{A}|<1$, it results in the exploitation of the search space and when $|\vec{A}|>1$, it enhances the exploration of the algorithm. The basic pseudo−code is shown in Figure 3.3.

```
GWO
Initialize the random population of grey wolf X_i (i= 1,2,.....n)
Initialize parameters a, A and C
Evaluate the fitness value of all search agents
X_α = fittest member of search space
X_β = second fittest member of search space
X_δ =third fittest member of search space
While (k<max value of iteration)
    For each member in search space
        Update their current position with the  equation no.(11)
    end for
    Update parameters a, A, C
    Evaluate the fitness value of all search agents
    Update the value of  X_α, X_β, X_δ
   k=k+1
end while
return X_α
```

Figure 3.3 Pseudo code of GWO

3.2 PROPOSED APPROACH-1

ENHANCED GREY WOLF OPTIMIZATION

GWO is SI based optimization technique, which mimics the hunting behavior and social leadership of grey wolves. Original GWO does not guaranteed us for exact approximation of global solution and also suffers from local optima stagnation. That's why to improve its performance, a new enhanced version of GWO algorithm has been proposed. The technique employs two modifications in the original GWO. Firstly, diversity among search agents has been increased with opposite based learning technique [152]. Secondly, value of parameter $\vec{a}$ oscillate in the range [2, 0] for 75% of iteration and attains some fixed value for remaining period.

- Opposition based learning.
- Oscillatory function [153].

Opposition based learning (OBL): The concept of the opposition based learning (OBL) technique was proposed by Tizhoosh [152] and is based on the theory of the simultaneous use of random number and its opposite number in order to estimate better position of the search agents. The opposite number can be evaluated by using following relation:

Consider a solution $\boldsymbol{X_i}$ in d−dimension search space as $\boldsymbol{X_i}$= $\{ x_{i,1}, x_{i,2}, x_{i,3},\dots x_{i,d}\}$. The opposite point of this solution ($X_i' = \{ x_{i.1}', x_{i,2}',\dots x_{i,n}' \}$) can be calculated as:

$$x'_{i.1} = (x_{max,j} + x_{min,j}) - x_i \qquad (3.12)$$

where i denotes the number of solution in search space, j denotes the dimensionality of the solution in the search space, x_{max} and x_{min} denotes the upper and lower bound.

In the basic algorithm of GWO, the initial population was generated using random numbers only. This random generated number can be very near to optimal value or can be very far away from the optimal value and hence affects the speed of algorithm to find the optimum value (which is known as computational time). If the initial random guess is located at far position from the optima value then the convergence time of an algorithm increases. This problem can be reduced by simultaneously checking for opposite of this number and choosing the best one i.e. one with better fitness value [154]. In this proposed version, concept of opposition has been applied in partial manner (i.e. population become opposite to the initially generated solution only in some dimensions). The partial opposition of the search agent is calculated if they follow the equation (3.13), where p is switching probability and this method is used in the both initialization and evolutionary phase (i.e. while generating the new population).

$$rand < p \qquad (3.13)$$

Oscillatory function: In E−GWO, oscillatory function has been used for the parameter *a* instead of linear decreasing function. In this function wave of global search (exploration) is followed by the wave of local search (exploitation) makes one cycle. This cycle is repeated over the course of iterations. By this method, there is periodically transition from the exploration phase to exploitation phase. This oscillating behavior of $\vec{a}$ will help in boosting exploration and exploitation and so, will help in finding global optima value. This behavior can be implemented by following equation:

$$\mathrm{a(t)} = \left(\frac{a_{min} + a_{max}}{2}\right) + \frac{a_{max} - a_{min}}{2}\cos\left(2\pi\frac{t}{T}\right) \qquad (3.14)$$

where a_{max} and a_{min} defines the oscillating range, t is iteration counter. T is period of oscillation and is given by

$$T = \frac{2S_1}{3 + 2k} \qquad (3.15)$$

where *k* can take any value from [1, 3, 5, 7]. In this approach we assigned *k=3*. More the value of *k,* more the algorithm goes under oscillation cycle. S_1 is number of iteration for which parameter $\vec{a}$ oscillates and for remaining iteration, parameter *a* is kept constant. In this proposed approach, parameter $\vec{a}$ oscillates between [2, 0] for first 75% of the iteration and then attains

0.02356 for the remaining iteration. This small value is assigned to $\vec{a}$ to emphasize exploitation in the remaining iterations. The pseudo code of EGWO is given in Figure 3.4.

```
EGWO
Initialize the random population of grey wolf X_i (i= 1,2,.....n)
Compute its fitness
for( every member in search space)
     performs the partial opposition if they follows Eq.(13)
     compute its fitness
     compare this fitness value with the original set of population and  choose the best
end for
Initialize parameters a, A,C
Evaluate the fitness value of all search agents
 X_α = fittest member of search space
 X_β = second fittest member of search space
 X_δ =third fittest member of search space
While(k<max value of iteration)
     for each member in search space
          Update their current position with the  equation no.(11)
     end for
    for( 75% of iteration)
          Update parameters a using equation no. (15)
    else
         use a as 0.02356
    end for
     Update parameter A, C
     Evaluate the fitness value of all search agents
     Update the value of  X_α, X_β, X_δ
     for( every member in search space)
          performs the partial opposition if they follows Eq.(13)
          compute its fitness
          compare this fitness value with the original set of population and
          choose the best and pass this to next iteration
      end for
     k=k+1
end while
return  X_α
```

Figure 3.4 Pseudo code of EGWO

3.3 PROPOSED APPROACH-2

EXTENDED GREY WOLF OPTIMIZATION (GWO-E)

As discussed above GWO is good in solving many optimization problems but it still has some limitations. Original GWO does not guarantee for exact approximation of the global solution because of its local optima stagnation. That’s why to improve its performance, a new extended version of GWO (GWO−E) algorithm has been proposed. This technique employs two modification in the original GWO. Firstly, when the value of $|\vec{A}|$ >1, switching to the different areas of the search space has been made by assigning different positions to the best members. This is done to get rid of local optima stagnation and to enhance exploration. Secondly, diversity has been brought among the search agents in the initial set of the iterations by using opposition based learning approach with an aim to find the global solution. A detailed discussion about the use of these modifications has been discussed in the subsequent sub-section.

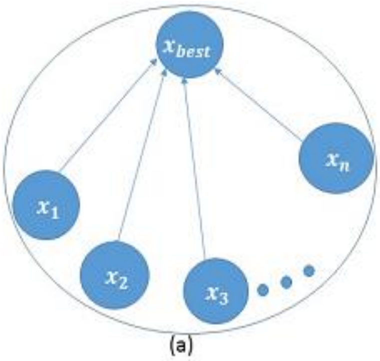
3.3.1 Modification 1

Why the condition of $|A|>1$? While approaching the prey value of $\vec{a}$ is reduced from 2 to 0 and thereby the range of $\vec{A}$ is also decreased. The fluctuating range of $\vec{A}$ lies in the interval of [−2a, 2a] as per the relation given in the equation (3.3). When the value of $\vec{A}$ lies between the range [−1, 1], the step size will be small. Hence, the next updated position of the agents will be around the prey's position as shown in the Figure 3.2 (a). It means that this range of $\vec{A}$ make the wolves to move toward the prey resulting in the exploitation of the search space. In order to search for the food (prey), grey wolves need to move away from each other. Mathematically it can be achieved with the help of parameter $\vec{A}$. When the value of $\vec{A}$ is less than −1 or greater than 1, the step size increases. Thus, the next position of the agent will be far away from the best (prey's) position as shown in Figure 3.2 (b). Thus, these values of $\vec{A}$ compel the wolves to diverge from each other resulting in exploration of search space.

As seen from the above discussion that the value of $\vec{A}$ controls the extent of exploration and exploitation in GWO. So, to enhance exploration ability of the search agents, new modifications have been proposed in the extended version of the GWO which depends on the value of the $\vec{A}$.

How new condition is imposed in GWO-E? In order to get rid of the problem of local optima stagnation, a new criterion has been used to find the positions of the alpha, beta and delta when $|\vec{A}|>1$. The whole population is randomly divided into three groups. The best value from first group is assigned to alpha score and its corresponding position is referred as alpha2 position. Similarly, best from second group is termed as beta2, best from third group is termed as delta2 and the remaining wolves update their position in the search space using equations from 3.5 to 3.11. In the exploitation phase ($|\vec{A}|<1$), alpha, beta and delta agents are the top three fittest members from the population and remaining wolves follow them to exploit the prey using equation from 3.5 to 3.11.

Why division of population into three groups? In the original GWO, best three members of the group are used to update the positions of remaining members (as shown if Figure 3.1). This use of best members leads to the intensive exploitation of the search space and local optima stagnation (as shown in Figure 3.1). In this figure circle denotes the whole search space, x_{best} shows the position of best particle and $x_1, x_2, \ldots x_N$ shows the position of remaining search particles with respect to the best. From the Figure 3.5 (a) it is clear that the search agents update their position according the best particle i.e. they are moving in some particular direction while other area remain unexplored. Hence, it leads to the exploitation. From the Figure 3.5 (b), it can be inferred that search agents always keep on following the best particle even if it gets stuck at some local solution and is far away from global solution. Under this scenario, algorithm fails to reach at global optima and leads to premature convergence.



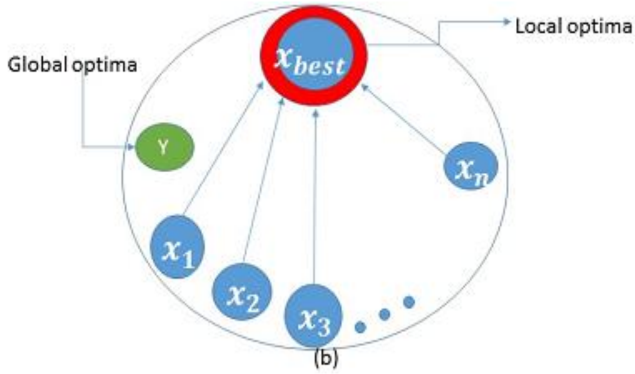


Figure 3.5 (a) Movement of random solutions toward the best; (b) Movement of random solutions toward the best and away from the global optima.

To get rid of this situation, the algorithm needs to generate diverse solutions. So, to promote diversification, population is randomly divided into three groups and best member of each group is selected as alpha2, beta2 and delta2. Then the remaining wolves will update their positions according to the newly generated values of alpha, beta and delta by using equation (3.11). This approach will provide a new direction to explore some new search areas and is expected to provide better results as compared to original GWO.

Using different random numbers: It can be inferred from the above equations (3.8, 3.9, and 3.10) that the vector $\vec{A}1$ is used to provide some weight to distance between its current position and alpha's position. Similarly, the vector $\vec{A}2$ and $\vec{A}3$ are used to provide some weight to distance between its current position and beta, delta position respectively. The value assigned to these three vectors is calculated according to the equation (3.3). Here three different random numbers are used to calculate the value of these parameters, which will increase the randomness in the algorithm and help in increasing diversification or the exploration of search space.

3.3.2 Modification 2

In order to enhance the exploration of the algorithm, search agents need to explore more search areas in the initial generations to find global solution and then later converge to exploit the global solution. This has been done using opposite based learning (OBL) technique in the initial iterations. The concept of OBL was introduced by Tizhoosh [152]. The main idea behind this theory is the use a random solution and its oppositely generated number in order to find global solution. The concept of opposite generated number is defined as:

Let $\boldsymbol{X_i}$ be a solution in space with d-dimension given as $\boldsymbol{X_i}$= { $x_{i,1}, x_{i,2}, x_{i,3},\ldots x_{i,d}$}. The opposite point ($X_i' =$ { $x_{i.1}', x_{i,2}',\ldots x_{i,n}'$}) to this approximated solution can be calculated by relation

$$x_{i.1}' = (x_{max,j} + x_{min,j}) - x_i \tag{3.16}$$

Why opposite based learning technique? Positions of grey wolves in the search space is extremely important task in grey wolf optimization. It affects the final solution and convergence speed. In this article, initial set of population is generated using random initialization. This population consist of 'n' solutions, where each solution consists of d-dimension. Random solutions can be generated by following equation:

$$x_{i,j} = x_{min,j} + \text{rand}\ (0, 1)\ (x_{max,j} - x_{min,j}) \tag{3.17}$$

where $x_{max,j}$ and $x_{min,j}$ are the upper and lower bound of the problem, *j* is the dimensionality of vector (*j=1, 2,3…d*), *I* denotes the number of solution (*i= 1, 2, 3…n*).

The random selection of the solution has a chance of revisiting some search area again and again. Hence, it reduces the diversity among the solutions. Thus, to enhance the diversity in the randomly generated population, the concept of opposition based learning (OBL) has been used. Using this method, some dimensions of solution can become opposite to the estimated solution. Therefore, the new combination of population will have diverse solutions [155]. Thus, the idea of OBL will helps in enhancing the exploration of the GWO.

How OBL is applied? In the starting 50% of the iterations, population is divided into two equal parts without sorting them. First half of the population is passed as such to the wolves. Then in the second part, partial opposition technique is used i.e., some dimensions of the solutions are replaced randomly by their opposite number. Here division of population is done without sorting to take advantage of randomness which enhances the diversity and results in better exploration ability of GWO. This excessive exploration in the starting iterations will help in finding the global solution which will be exploited by other wolves. The pseudo-code for the proposed algorithm is given in Figure 3.6. The flowchart of this proposed algorithm is also presented in the figure 3.7 and it helps in better illustration of the algorithm.

```
GWO-E
Initialize the random population of grey wolf X_i(i= 1,2,.....n)
Initialize parameters a, A and C
Evaluate the fitness value of all search agents
X_α1= fittest member of search space
X_β1= second fittest member of search space
X_δ1=third fittest member of search space
X_α2=fittest member of part1 of population
X_β2=fittest member of part2 of population
X_δ2= fittest member of part3 of population
While(k<max value of iteration)
    If( k<( max value of iteration)/2)
        Divide the population randomly into two parts
        Pass the first part as such and pass the second part after applying partial opposition
        Calculate the value of alpha, beta and delta
        Divide the population into three parts
        Evaluate alpha2, beta2, delta2
    else
        pass the whole population as such
        calculate the value of alpha, beta and delta
        Divide the population into three parts
        Evaluate alpha2, beta2, delta2
    end if
    evaluate the value of a, A, C
    for every member in search space
        if |A|<1
            update the current position by using equation (11) by using alpha, beta, and delta
        else
            update the current position with equation (11) by using alpha2, beta2 and delta2
        end if
    end for
    update parameters a, A, C
    Evaluate the fitness value of all search agents
    Update the value of X_α1, X_β1, X_δ1, X_α2, X_β2, X_δ2

    k=k+1
end while
return X_α
```

Figure 3.6 Pseudo code of GWO-E

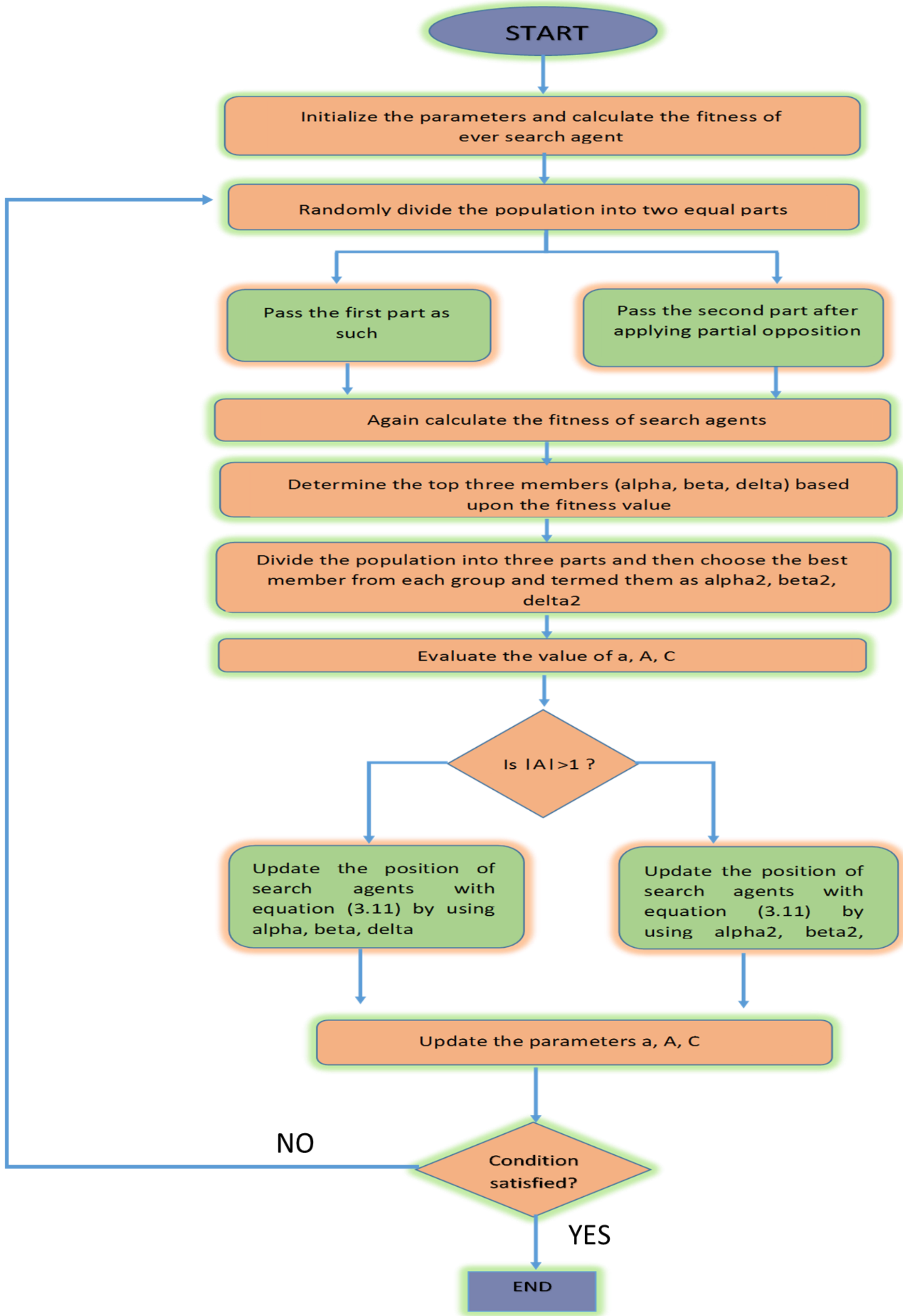


Figure 3.7 Flowchart of the GWO-E

CHAPTER 4

PROBLEM FORMULATION

The synthesis of unequally spaced (LAA) is classical electromagnetic problem. The non-uniform spacing between the antenna's yield the aperiodic array factor and this aperiodic nature of array factor helps in attaining the minimum SLL with lesser number of antennas for given aperture size. Also in NUSLA, amplitude and phase of excitation current is kept as constant. This results in the low cost and complexity of the network. But because of the non- linear relation between the array factor and position of elements, the synthesis process is complicated. In this present work, synthesis of this array is done using the proposed algorithm namely, GWO-E. The main aim of the work is to minimize the side lobe level and placement of nulls in desired direction. The results obtained by GWO-E are compared with other work which have found in literature. The synthesis of linear antenna array is discussed in the next section.

4.1 SYNTHESIS OF LINEAR ANTENNA ARRAYS

In linear antenna array, M isotropic elements are placed in the line. Figure 4.1 depicts the symmetrical linear antenna array with 2N elements placed along the x-axis. Form the figure, it can be seen that there are N elements on the each side of origin and this type of symmetric arrangement of elements helps in providing symmetric radiation pattern.

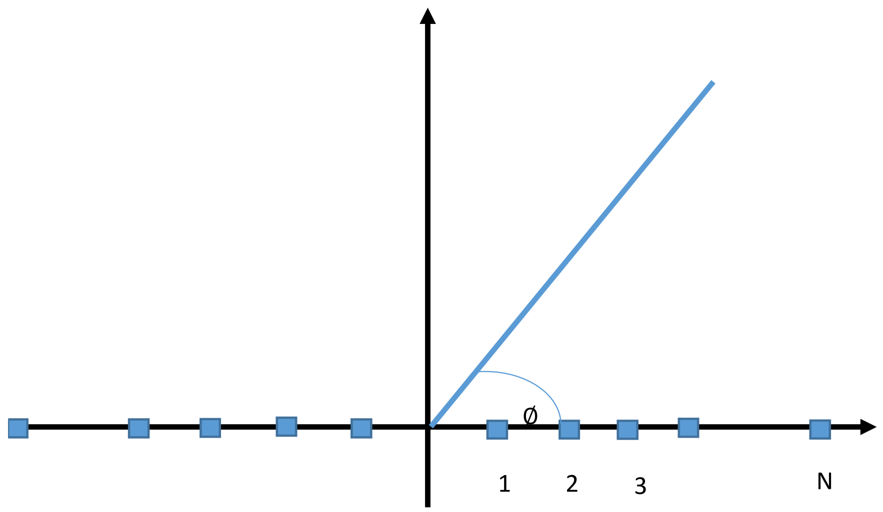


Figure 4.1 Arrangement of linear antenna array of 2N elements

In the symmetric LAA, only N elements are optimized instead of 2N elements and hence, it reduces the computational complexity. Due to symmetry, the array factor is given by relation [36]:

$$AF(\emptyset) = 2\sum_{n=1}^{N} I_n \cos[kx_n \cos(\emptyset) + \psi_n] \tag{4.1}$$

where I_n , ψ_n is the amplitude and phase of excitation current and x_n is the position of nth antenna in the array, $\emptyset$ is the azimuthal angle and the wave number k is represented as 2 π / λ. The array factor of LAA depends on phase and amplitude of excitation current, wave number and location of array element. Hence, linear antenna array can be optimized by controlling any of these parameter and by keeping other parameters as constant. In this thesis work, linear antenna array design problem is solved by controlling the inter-element spacing.

For the position optimization of an antenna array, the all the elements of an array are feed with constant amplitude and phase (I_n=1, ψ_n=0). Thus in this case, the equation of array factor modifies to

$$\mathrm{AF}(\emptyset) = 2\sum_{n=1}^{N} \cos(k\, x_n \cos(\emptyset)) \tag{4.2}$$

In this work, GWO-E is used to find the optimized position of element of unequally spaced LAA. If the antenna elements are placed too closely, this leads to mutual coupling effect and if the antenna elements are placed at far distance from each other, this leads to grating lobes. Thus the proper placement of small antenna in an array is very critical task and the following condition must be satisfied

$$|x_i - x_j| > 0.25\lambda \tag{4.3}$$

$$\min\{x_i\} > 0.125\lambda;\; i = 1,2,\ldots,N,\; i \neq j, \tag{4.4}$$

where x_j is the antenna position adjacent to the x_i and $\{x_i\}$ is the set of all antenna positions.

The aim of synthesis of antenna array is to achieve a radiation pattern with minimum value of side lobe level and to place nulls in desired directions. The objective function for optimization is defined as

$$F_{obj} = k_1 f_{SL}(\emptyset) + k_2 f_{NS}(\emptyset) \tag{4.5}$$

where f_{SL} is the function defined to minimize the side lobe level and f_{NS} is function for null placement, as given in equation (4.6 and 4.7). The k_1 and k_2 are weights of two different functions.

$$f_{SL}(\emptyset) = \max[AF(\emptyset)] \tag{4.6}$$

$$f_{NS}(\emptyset) = [\max(AF(\emptyset_k^{null}))_{k=1\ldots\ldots\ldots K} \tag{4.7}$$

CHAPTER 5
RESULT AND DISCUSSION

This chapter is divided into three sections. The first section, includes the detail of the all benchmark functions or test function used for the evaluation of proposed approaches. In the second section, the two proposed algorithms have been applied to these benchmark functions and the results are compared with other well-known algorithms. This section includes the convergence curves and global value achieved by the proposed approaches. Finally, the section three deals with the synthesis of linear antenna array using GWO-E. The simulation is performed on "Window 10" PC having 4GB RAM with Intel i5 processor. The MATLAB version used for the synthesis is R2013a.

5.1 TEST FUNCTIONS

Test functions or the benchmark function are those function which are used to test the quality of optimization algorithm i.e. whether an algorithm is able to achieve global optimum value or not. These functions are also helpful in calculating the convergence speed of an algorithm. These benchmarks problems are taken from the CEC 2005 technical report [83].

5.1.1 Test suite used for Enhanced Grey wolf optimization (EGWO)

For this algorithms we are using 9 benchmark functions. Mathematical equation of these benchmark functions along with their dimension, global optima and ranges are provided in Table 5.1.

Test problem	Objective Function	Search Range	Optimum Value	Dimension
Branin	$f_1(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	$x_1 \in [-5, 10]$, $x_2 \in [0, 15]$	0.397887	2
Cigar	$f_2(x) = x_0^2 + 10000\sum_{i=1}^{D} x_i^2$	[−10, 10]	0	30
Easom	$f_3(x) = -cosx_1 cosx_2 e^{(-(x_1-\pi)^2-(x_2-\pi)^2)}$	[−10, 10]	−1	2
Elliptic	$f_4(x) = \sum_{i=1}^{D}(10^6)^{\frac{i-1}{D-1}}x_i^2$	[−100, 100]	0	50
Griewank	$f_5 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[−600, 600]	0	30

Rastrigin	$f_6(x) = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$	[−5.12, 5.12]	0	30
Penalized 1	$f_7 = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)\ 2[$ $10\sin2(\pi y_{i+1}) + (y_n - 1)2\} + \sum_{i=1}^{n} u$ 100, 4) $y_i = 1 + \frac{x_{i+1}}{4}$ u(x_i, a, k, m) = $\begin{cases} k(x_i - a)^m x_i > a \\ 0 \quad -a < x_i < a \\ k(-x_i - a)^m \quad x_i < -a \end{cases}$	[−50, 50]	0	30
Penalized2	$f_8 = 0.1\{(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2 [1 + sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i,$ 5, 100, 4) u(x_i, a, k, m) = $\begin{cases} k(x_i - a)^m x_i > a \\ 0 \quad -a < x_i < a \\ k(-x_i - a)^m \quad x_i < -a \end{cases}$	[−50, 50]	0	30
Scaffer	$f_9(x) = \left[\frac{1}{n-1}\sqrt{s_i}.\left(\sin\left(50.0s_i^{\frac{1}{5}}\right) + 1\right)\right]^2 s_i = \sqrt{x_i^2 + x_{i+1}^2}$	[−100, 100]	0	30

Table 5.1 Description of benchmark function used for the evaluation of EGWO [83]

5.1.2 Test suite used for Extended Grey wolf optimization (GWO-E)

Performance of the proposed approach have been tested on nineteen benchmark functions. These benchmark functions are unimodal functions (having no local optima and single global solution), multimodal functions (having many local minima), and fixed dimension b benchmark functions. The details of these functions are given in Table 5.2, Table 5.3, and Table 5.4.

Function	Dim	Range	f_{min}
f_1=$\sum_{i=1}^{n} x_i^2$	30	[−100, 100]	0
f_2=$\sum_{i=1}^{n} \lvert x_i \rvert$+$\prod_{i=1}^{n} \lvert x_i \rvert$	30	[−10,10]	0
f_3=$\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	30	[−100, 100]	0
f_4= $max_i\{\lvert x_i\rvert, 1\leq i \leq n\}$	30	[−100, 100]	0
f_5=$\sum_{i=1}^{n-1}[100(x_{i+1} - x_i^2)$^2 $+ (x_i - 1)$^2]	30	[−30,30]	0
f_6=$\sum_{i=1}^{n}([x_i + 0.5])$^2	30	[−100, 100]	0
f_7=$\sum_{i=1}^{n} ix_i^4$+random[0,1)	30	[−1.28,1.28]	0

Table 5.2 Unimodal Benchmark function [83]

Function	Dim	Range	f_{min}
$f_8 = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)+10]$	30	[−5.12,5.12]	0
$f_9(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	30	[−32,32]	0
$f_{10}(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	30	[−600,600]	0
$f_{11}(x) = \frac{\pi}{n}\{10\sin(\pi y_1) + \sum_{i=1}^{n-1}(y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2\} + \sum_{i=1}^{n} u(x_i, 10, 100, 4)$ $y_i = 1 + \frac{x_{i+1}}{4}$	30	[−50,50]	0
$f_{12}(x) = 0.1\{\sin^2(3\pi x_1) + \sum_{i=1}^{n}(x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)] + (x_n - 1)^2[1 + \sin^2(2\pi x_n)]\} + \sum_{i=1}^{n} u(x_i, 5, 100, 4)$	30	[−50,50]	0

Table 5.3 Multimodal Benchmark Functions [83]

Function	Dim	Range	f_{min}
$f_{13}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\frac{1}{j+\sum_{i=1}^{2}(x_i - a_{ij})^6}\right)^{-1}$	2	[−65,65]	0
$f_{14}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	[−5,5]	0.00030
$f_{15}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	[−5,5]	−1.0316
$f_{16}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	2	[−5,5]	0.398
$f_{17}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	[−2,2]	3
$f_{18}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right)$	3	[1,3]	−3.86
$f_{19}(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right)$	6	[0,1]	−3.32

Table 5.4 Multimodal benchmark Functions with Fixed dimensions [83]

5.2 VALIDATION OF THE PROPOSED WORK ON BENCHMARK FUNCTIONS

5.2.1 Results of enhanced grey wolf optimization (EGWO)

The performance of this algorithm have been tested on nine benchmark function and is compared with BA, BFP, FA, FPA, and GWO. Population size used and maximum iteration size for all the algorithm is set to 50,500. Benchmark functions used consist of both unimodal and multimodal functions to study the exploitation and exploration properly. Mathematical equation of these benchmark functions along with their dimension, global optima and ranges are provided in Table 5.1 and parameter settings are given in Table 5.5. To analyze the performance of the proposed work, 50 runs are performed and best, mean, worst and standard deviation are calculated from these runs.

Algorithm	Parameter settings
BA	Loudness (A)=0.5, Pulse Rate(r)=0.5
BFP	Loudness(A)=0.5, Pulse rate(r)=0.5, Switch Probability (p) =0.8
FA	Random parameter(α)=0.25, Light absorption coefficient (γ)=1, Attractiveness (β_o)=1
FPA	Switch Probability (p)=0.8
GWO	a is linearly decreased from 2 to 0, A=[-2a, 2a], C=[0,2]
EGWO	a oscillate in[0,2], A=[-2a, 2a], C=[0,2],p=0.4, k=3

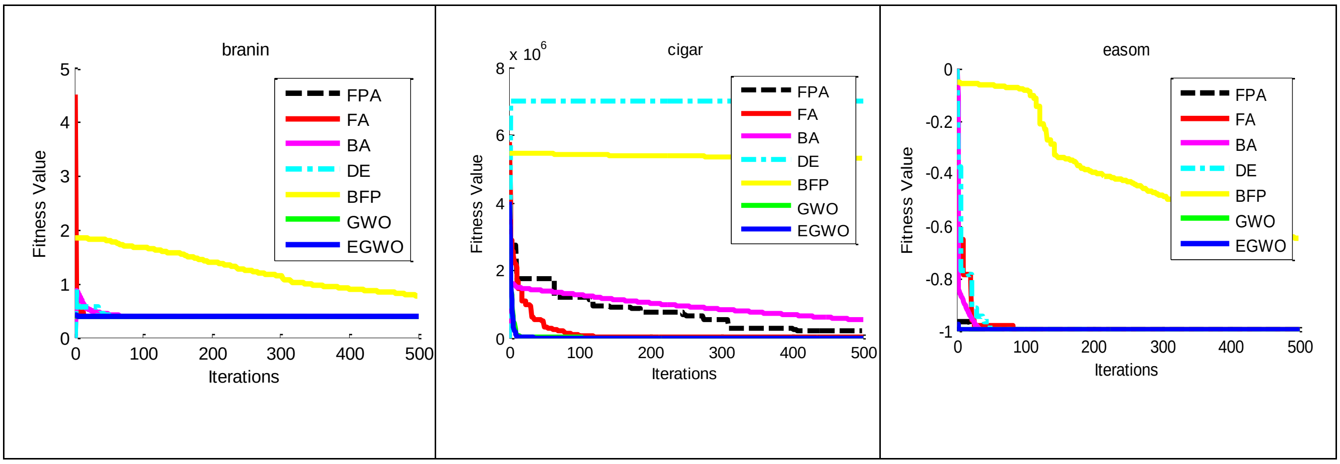
Table 5.5 Parameter setting of various algorithm

Simulation results of the proposed algorithm compared with other algorithms are given in Table 5.6. From the table it can be inferred that for function f_1 BA has better value for standard deviation. For function f_2 only EGWO is able to achieve global optima value. For function f_3 standard deviation of EGWO is good. For function f_4, only EGWO attains global optimum value, while other techniques stuck at non−optimum value. For f_5 and f_6, EGWO attains exact solution. For f_7 best, worst and average of GWO are competitive with EGWO but standard deviation of EGWO is better. For f_8 FA has better results. For f_9 both GWO and EGWO attains exact global solution. In this case BA, FA and GWO shows better results for one function each, while EGWO shows better performance for seven functions. So, EGWO is found to be best among standard algorithm. Convergence profiles in Figure 5.1 also prove the competitiveness of the proposed technique with respect to other state of art algorithms.

Function	Algorithm	Best	Worst	Mean	Standard Deviation
f_1	BA	3.98E−01	3.98E−01	3.98E−01	**5.35E−10**
	BFP	3.98E−01	2.02E+00	6.30E−01	3.64E−01
	FA	3.98E−01	3.98E−01	3.98E−01	3.12E−09
	FPA	3.98E−01	3.98E−01	3.98E−01	4.25E−06
	GWO	3.98E−01	3.98E−01	3.97E−02	6.58E−05
	EGWO	3.98E−01	3.98E−01	3.98E−01	6.11E−09
f_2	BA	5.01E+05	1.16E+06	4.90E+05	2.72E+05
	BFP	3.06E+06	6.82E+06	5.55E+06	8.31E+05
	FA	3.74E+00	1.54E+02	3.80E+01	2.84E+01
	FPA	2.05E+05	5.87E+05	3.69E+05	8.62E+04
	GWO	4.40E−36	2.32E−33	1.85E−34	4.02E−34
	EGWO	**5.64E−264**	**5.68E−228**	**1.14E−229**	**0.00E+00**
f_3	BA	−1.0000	−8.11E−05	−0.980002	0.14141
	BFP	−1.0000	−4.54E−07	−0.57767	0.38706
	FA	−1.0000	−1.0000	−1.0000	4.38E−09
	FPA	−1.0000	−1.0000	−1.0000	3.89E−10
	GWO	−1.0000	−1.0000	−1.0000	1.81E−07
	EGWO	−1.0000	−1.0000	−1.0000	**6.55E−10**
f_4	BA	1.58E+08	1.33E+09	5.07E+08	2.71E+08
	BFP	8.52E+08	3.22E+09	1.91E+09	5.58E+08
	FA	1.02E+06	8.23E+06	4.02E+06	1.73E+06
	FPA	2.10E+07	1.12E+08	5.26E+07	1.77E+07

	GWO	1.16E−34	1.00E−31	1.09E−32	2.13E−32
	EGWO	**1.19E−263**	**2.07E−237**	**4.1E−239**	**0.00E+00**
f_5	BA	6.86E+01	2.46E+02	1.36E+02	41.1407
	BFP	3.60E+02	6.72E+02	5.49E+02	75.25258
	FA	5.76E−03	2.97E−02	1.06E−02	0.005624
	FPA	2.17E+01	5.78E+01	3.50E+01	8.387046
	GWO	0.00E+00	0.044	0.0025	0.0076
	EGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_6	BA	2.89E+01	2.02E+02	7.53E+01	3.03E+01
	BFP	2.70E+02	4.03E+02	3.61E+02	2.72E+01
	FA	1.67E+01	8.64E+01	3.31E+01	1.20E+01
	FPA	1.61E+02	2.37E+02	2.04E+02	1.52E+01
	GWO	0.00E+00	1.9E+01	1.9E+00	3.6924
	EGWO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_7	BA	7.88E+02	2.31E+07	3.53E+06	4.51E+06
	BFP	5.88E+07	7.17E+08	4.13E+08	1.44E+08
	FA	1.20E−04	3.09E−03	6.69E−04	7.01E−04
	FPA	13.95913	5.01E+05	5.09E+04	9.61E+04
	GWO	1.97E−06	0.0719	0.0216	0.0141
	EGWO	**1.51E−06**	**2.90E−06**	**2.17E−06**	**2.93E−07**
f_8	BA	2.06E+06	1.12E+08	2.89E+07	2.65E+07
	BFP	1.36E+08	1.48E+09	9.56E+08	2.43E+08
	FA	1.09E−03	1.49E−02	3.49E−03	**2.10E−03**
	FPA	6.58E+04	3.85E+06	1.12E+06	8.82E+05
	GWO	2.47E−05	0.7774	0.3314	0.1738
	EGWO	1.98E−05	2.248427	0.7315	0.7447
f_9	BA	7.97E−14	3.22E−01	5.75E−02	7.38E−02
	BFP	3.43E−02	4.39E−01	2.26E−01	1.09E−01
	FA	1.58E−12	2.13E−02	2.62E−03	5.36E−03
	FPA	2.70E−10	1.49E−06	1.54E−07	2.70E−07
	GWO	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00
	EGWO	**0.00E+00**	0.00E+00	0.00E+00	0.00E+00

Table 5.6 Comparison results with respect to state-of the-art algorithms



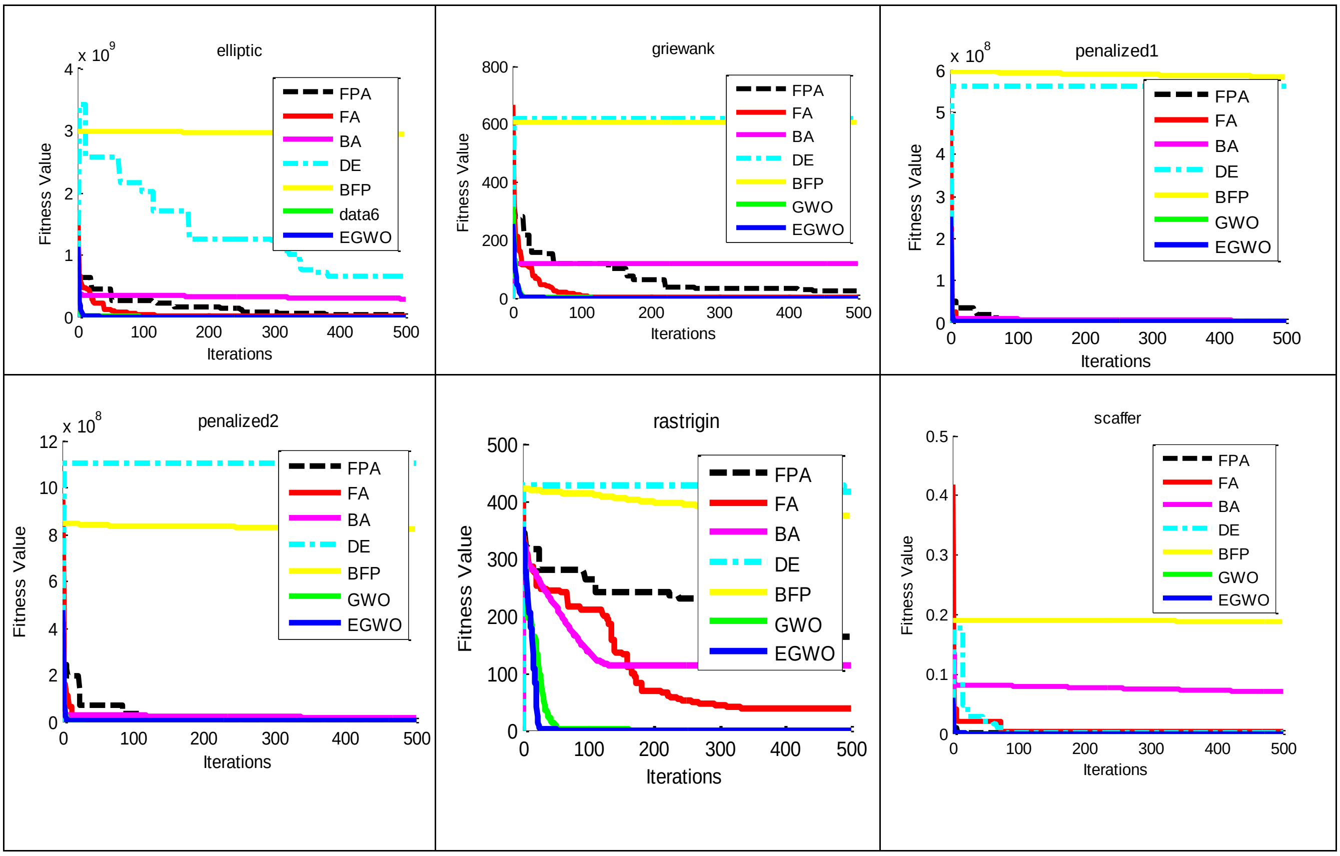


Figure 5.1 Convergence profile of various algorithms

Inference Drawn: The EGWO is found to be better than original GWO for all the benchmark functions and it is better among all the algorithms for seven functions. Although this algorithm provides better results over the basic algorithm but because of the use of opposition based learning method in initialization and evolutionary phase, the function evaluations increases. Higher are function evaluations, more is the time required for an algorithm to give the global solution. Hence, this OBL method limits the use of this algorithm for the real world applications where minimum time required is constraint for optimization.

5.2.2 Results of extended grey wolf optimization (GWO-E)

The performance of this algorithm have been tested on nineteen benchmark function and is compared with BA, BFP, DE, FA, FPA, and GWO. These test functions consist of both unimodal and multimodal function so as to study the exploitation and exploration properly. Detail of these functions are given in Table 5.2, Table 5.3, Table 5.4 and parameter setting of all the algorithm is given in Table 5.7. Different population sizes and different dimension sizes are used to study the performance of GWO-E.

Algorithms	Parameters	Values
BA	Loudness(A)	0.5
	Pulse rate(r)	0.5
	Maximum iterations	500
	Stopping criteria	Max. Iteration
BFP	Probability switch	0.8
	Maximum iterations	500
	Stopping criteria	Max. Iteration
DE	Control parameter(F)	0.5
	Crossover rate (CR)	0.5
	Maximum iterations	500
	Stopping criteria	Max. Iteration
FA	Randomization parameter(α)	0.5
	Attractiveness (β)	0.2
	Light absorption co-efficient (γ)	1
	Maximum iterations	500
	Stopping criteria	Max. Iteration
FPA	Switch probability	0.8
	Maximum iterations	500
	Stopping criteria	Max. Iteration
GWO	$\vec{a}$	Linearly decreased from 2 to 0
	Maximum iterations	500
	Stopping criteria	Max. Iteration

Table 5.7 Parameters setting of algorithms.

A. Effect of population

In this work three different population sizes (*n)* of 30, 60, and 90 with dimension size 30 (except for fixed dimension function as given in table4) are used to test the performance of GWO and GWO-E and the results are compared with other state-of-the-art algorithms. Table 5.8 shows the comparison for *n=30*, Table 5.9 shows the comparison for *n=60* and Table 5.10 for *n=90*. The results are discussed as below:

Case I (for population size of 30): In Table 5.8, simulation results for population size of 30 are presented. It can be inferred from table that for functions f_1, f_2, f_3, f_4 , GWO-E provides better results when compared to other algorithms. For f_5 GWO-E is able to achieve better value of global optima as compared with BA, BFP, FA, FPA though results of GWO are comparable. For function f_6 only GWO-E is able to find global optima while all other algorithm stuck at some non-optimum solutions. GWO-E gives better results for f_7 though GWO has comparable results. GWO-E provides exact optimum results for function f_8. For f_9 , GWO-E is better in term of best, worst, average and standard deviation as compared with other algorithms. For f_{10} GWO-E provides exact optimum results. FA achieves best results for f_{11} although results of GWO-E are quite comparable. For f_{12}, FA achieved global minima with good standard deviation. For f_{13} results of BA, FA, FPA, GWO, GWO-E are highly competitive but standard deviation of FPA is better. For f_{14}, best, worst, average of GWO-E are better though standard deviation is competitive with FPA; overall GWO-E can be considered as better. FA has the best standard deviation for fixed dimension multimodal functions f_{15} and f_{17}. BA outperforms all other algorithms for function f_{16} and DE outperforms for function f_{18} with good standard deviation and best value of all algorithms are highly competitive. For f_{19}, FPA has better performance with good standard deviation. Thus, for population size of 30, GWO-E shows better results for eleven functions, FA achieve good results for four functions, BA for one, FPA for two, DE for one. Thus, GWO-E proves to be best algorithm for population size of 30.

`Functions	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	8.93E+03	3.75E+04	1.99E+04	6.56E+03
	BFP	3.88E+04	8.09E+04	6.34E+04	1.01E+04
	DE	3.12E+04	7.93E+04	6.65E+04	1.01E+04
	FA	5.97E−03	4.27E−02	1.66E−02	7.41E−03
	FPA	1.34E+03	4.77E+03	2.65E+03	7.20E+02
	GWO	5.40E−30	1.46E−26	1.45E−27	2.39E−27
	GWO−E	**4.72E−64**	**4.23E−59**	**3.08E−60**	**7.38E−60**
f_2	BA	3.47E+01	5.02E+05	1.05E+04	7.09E+04
	BFP	3.04E+04	3.11E+10	1.92E+09	5.73E+09
	DE	4.84E+01	4.97E+04	1.54E+03	7.47E+03
	FA	2.47E−01	1.07E+00	4.97E−01	1.60E−01
	FPA	2.90E+01	7.62E+01	4.40E+01	1.02E+01
	GWO	1.56E−17	2.91E−16	8.41E−17	6.17E−17
	GWO−E	**2.74E−34**	**3.78E−31**	**1.70E−32**	**5.32E−32**
f_3	BA	2.46E+04	1.53E+05	6.89E+04	3.25E+04
	BFP	7.09E+04	2.19E+05	1.37E+05	4.21E+04
	DE	3.24E+04	1.46E+05	6.95E+04	2.68E+04
	FA	7.66E+02	6.32E+03	3.00E+03	1.11E+03
	FPA	1.25E+03	7.30E+03	4.30E+03	1.43E+03
	GWO	2.25E−08	1.91E−04	1.96E−05	4.19E−05
	GWO−E	**3.33E−38**	**5.34E−30**	**1.46E−31**	**7.61E−31**
f_4	BA	3.50E+01	7.13E+01	5.31E+01	8.40E+00
	BFP	7.34E+01	9.29E+01	8.66E+01	3.87E+00
	DE	7.75E+01	9.33E+01	8.72E+01	3.91E+00
	FA	8.06E−02	2.76E+00	2.67E−01	4.17E−01
	FPA	17.80127	3.41E+01	2.49E+01	3.66E+00
	GWO	4.46E−08	4.00E−06	8.40E−07	8.83E−07
	GWO−E	**1.13E−25**	**1.66E−21**	**6.35E−23**	**2.52E−22**
f_5	BA	4.10E+06	6.16E+07	2.10E+07	1.35E+07
	BFP	9.05E+07	3.25E+08	2.15E+08	6.11E+07
	DE	1.31E+08	3.36E+08	2.49E+08	4.88E+07
	FA	2.65E+01	1.49E+03	1.89E+02	3.39E+02
	FPA	9.57E+04	1.13E+06	5.05E+05	2.49E+05
	GWO	2.60E+01	2.85E+01	2.69E+01	0.653194

	GWO−E	**2.55E+01**	**2.88E+01**	**2.70E+01**	**0.77102**
f_6	BA	7.45E+03	4.37E+04	1.90E+04	7.27E+03
	BFP	3.99E+04	8.09E+04	6.52E+04	9.53E+03
	DE	4.85E+04	8.36E+04	6.87E+04	7.64E+03
	FA	6.31E−03	4.63E−02	1.59E−02	7.52E−03
	FPA	1.37E+03	4.73E+03	2.53E+03	5.99E+02
	GWO	0.24924	1.7302283	0.7575417	0.330702
	GWO−E	**7.10E−05**	1.63E+00	6.74E−01	3.80E−01
f_7	BA	2.178145	25.412423	8.77521139	4.997397
	BFP	51.20016	1.52E+02	1.00E+02	24.42522
	DE	59.03261	1.78E+02	1.20E+02	25.89624
	FA	0.010996	0.1523028	0.04768919	0.029645
	FPA	0.145749	1.3126232	0.51345695	0.251664
	GWO	2.88E−04	0.0062799	0.00176838	9.67E−04
	GWO−E	**3.42E−05**	0.0010722	2.63E−04	2.45E−04
f_8	BA	30.84799	2.17E+02	8.95E+01	3.80E+01
	BFP	3.01E+02	4.38E+02	3.92E+02	3.04E+01
	DE	2.31E+02	4.91E+02	4.23E+02	5.29E+01
	FA	2.43E+01	8.95E+01	4.43E+01	1.19E+01
	FPA	1.54E+02	2.41E+02	1.92E+02	1.70E+01
	GWO	5.68E−14	12.04693	2.38904345	3.436346
	GWO−E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	BA	1.30E+01	1.86E+01	1.63E+01	1.198819
	BFP	1.96E+01	2.08E+01	2.04E+01	0.244626
	DE	1.74E+01	2.09E+01	2.02E+01	0.884499
	FA	3.47E−02	0.1247877	6.64E−02	0.021447
	FPA	9.57E+00	1.46E+01	1.23E+01	1.199062
	GWO	6.48E−14	1.75E−13	1.02E−13	1.83E−14
	GWO−E	**4.44E−15**	1.15E−14	5.79E−15	1.88E−15
f_{10}	BA	88.69344	3.43E+02	1.95E+02	64.24727
	BFP	4.03E+02	7.16E+02	5.63E+02	75.47466
	DE	1.94E+02	7.49E+02	5.85E+02	1.16E+02
	FA	0.004177	0.0259097	0.00989394	0.004209
	FPA	12.8981	43.830748	25.4924148	7.055165
	GWO	0	0.0212209	0.00328396	0.006481
	GWO−E	**0**	**0**	**0**	**0**
f_{11}	BA	8.22E+05	2.27E+08	2.35E+07	3.86E+07
	BFP	7.03E+07	8.31E+08	4.64E+08	1.62E+08
	DE	3.55E+08	8.47E+08	5.81E+08	1.20E+08
	FA	**6.57E−05**	0.1115914	5.13E−03	0.021536
	FPA	1.33E+01	7.44E+04	2.63E+03	1.09E+04
	GWO	0.013145	0.0861381	0.04040624	0.017797
	GWO−E	6.63E−03	0.1227367	0.03882816	0.023792
f_{12}	BA	4.31E+06	1.80E+08	6.66E+07	4.45E+07
	BFP	1.84E+08	1.45E+09	9.66E+08	2.79E+08
	DE	5.17E+08	1.60E+09	1.13E+09	2.29E+08
	FA	0.001094	1.68E−02	4.60E−03	**2.57E−03**
	FPA	3.10E+03	2.15E+06	3.53E+05	4.18E+05

	GWO	0.103227	1.4477669	0.66453317	0.252479
	GWO−E	0.101593	1.0135696	0.5545413	0.222828
f_{13}	BA	0.998004	22.900634	10.3639395	7.503744
	BFP	1.001322	4.66E+02	1.17E+02	1.33E+02
	DE	0.998004	7.873993	2.26086043	2.106046
	FA	0.998004	6.9033357	2.06809447	1.288629
	FPA	0.998004	0.9987864	0.99804039	**1.32E−04**
	GWO	0.998004	12.670506	3.17557909	3.452361
	GWO−E	0.998004	10.763181	1.98739084	2.361803
f_{14}	BA	3.12E−04	0.110837	0.00705302	0.016219
	BFP	0.002146	0.3467017	0.0569699	0.058544
	DE	0.001027	0.0017941	0.00129879	3.20E−04
	FA	3.36E−04	0.0052979	0.00122428	9.83E−04
	FPA	8.61E−04	0.0011674	0.00103887	3.82E−05
	GWO	3.08E−04	0.0209439	0.00480718	0.008375
	GWO−E	**3.09E−04**	**0.0019887**	**9.53E−04**	3.28E−04
f_{15}	BA	−1.031628	−0.215464	−0.9826586	0.195796
	BFP	−1.031628	2.1042504	−0.4314169	0.622812
	DE	−1.031628	−1.031623	−1.0316284	7.07E−07
	FA	−1.031628	−1.031628	−1.0316284	**4.29E−09**
	FPA	−1.031628	−1.031628	−1.0316284	1.27E−07
	GWO	−1.031628	−1.031628	−1.0316285	2.06E−08
	GWO−E	−1.031628	−1.031396	−1.0315934	4.98E−05
f_{16}	BA	0.397887	0.3978874	0.39788736	**4.78E−10**
	BFP	0.397887	8.9015292	1.07030718	1.357477
	DE	0.397887	0.3981423	0.39789277	3.61E−05
	FA	0.397887	0.3978874	0.39788736	3.73E−09
	FPA	0.397887	0.3979167	0.39789491	7.95E−06
	GWO	0.397887	0.3978874	0.39788739	3.84E−06
	GWO−E	0.397887	0.4012151	0.39805721	6.62E−04
f_{17}	BA	3.00E+00	8.40E+01	4.62E+00	11.45513
	BFP	3.00E+00	9.20E+01	2.82E+01	2.63E+01
	DE	3.00E+00	3.00E+00	3.00E+00	2.10E−06
	FA	3.00E+00	3.00E+00	3.00E+00	**2.52E−08**
	FPA	3.00E+00	3.00E+00	3.00E+00	1.32E−08
	GWO	3.00E+00	84.000026	4.62004369	11.45513
	GWO−E	3.00E+00	3.0003036	3.00004272	6.26E−05
f_{18}	BA	−3.862782	−3.862782	−3.8627821	5.51E−08
	BFP	−3.86278	−3.041193	−3.7226379	0.212995
	DE	−3.862782	−3.862782	−3.8627821	**3.04E−10**
	FA	−3.862782	−3.862782	−3.8627821	1.23E−09
	FPA	−3.862782	−3.862781	−3.8627821	9.35E−08
	GWO	−3.862782	−3.854898	−3.8611064	0.002791
	GWO−E	−3.862782	−3.854848	−3.8618631	0.001908
f_{19}	BA	−3.321993	−3.202784	−3.2720279	0.05931
	BFP	−3.17074	−1.234585	−2.5261255	0.486519
	DE	−3.321993	−3.059302	−3.2067909	0.045087
	FA	−3.321995	−3.175559	−3.2739549	0.062103

	FPA	−3.315474	−3.202943	−3.2760638	**0.027019**
	GWO	−3.32199	−3.047868	−3.2479112	0.083517
	GWO−E	−3.321993	−3.075133	−3.2592015	0.069842

Table 5.8 Results of GWO-E compared with other algorithms for population size of 30

Case II (for population size of 60): Table 5.9 shows the comparison of GWO-E with other algorithms for *n= 60*. In this case, for f_1, f_2, f_3, f_4 only GWO-E is able to find global optimum value. In case of f_5, GWO gives the competitive results but standard deviation of GWO-E is better. Results obtained by GWO are better for function f_6, though GWO-E provides quite comparable results. GWO-E provides far better results for function f_7 and it provides exact optimum results for f_8. For f_9, GWO-E is better than GWO in terms of best, worst and average value and has same standard deviation. For f_{10}, only GWO-E is able to achieve exact optimum resuts. For f_{11}, GWO-E attains global optima with good standard deviation. For f_{12}, FA is better in term of worst, average and standard deviation over the GWO-E. For fixed dimension multi-modal functions f_{13} and f_{14}, FPA is better with good standard deviation. For function f_{15} and f_{18} DE performs better over the other algorithms with good standard deviation. For f_{16} and f_{17} BA act as good algorithm over other techniques with better standard deviation. For f_{19}, FPA shows better performance with good standard deviation. Thus, for population size of 60, GWO-E is best for ten functions, GWO for one, FA for one, DE and BA for two function each and FPA shows better performance for three functions. Thus, overall GWO-E is found to be best among the compared algorithm.

Function	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	4.38E+03	3.07E+04	1.26E+04	5.64E+03
	BFP	3.74E+04	7.95E+04	6.19E+04	7.94E+03
	DE	2.97E+04	7.71E+04	6.13E+04	8.41E+03
	FA	5.60E–03	2.86E−02	1.44E−02	5.40E−03
	FPA	2.72E+03	7.29E+03	4.75E+03	1.18E+03
	GWO	1.23E−36	1.91E−34	2.85E−35	4.21E−35
	GWO−E	**1.99E−71**	**6.84E−66**	**3.92E−67**	**1.11E−66**
f_2	BA	1.16E+01	1.47E+04	7.00E+02	2.73E+03
	BFP	6.51E+01	2.33E+08	2.24E+07	5.75E+07
	DE	5.27E+01	1.01E+06	2.82E+04	1.45E+05
	FA	0.2417	8.00E−01	4.72E−01	1.37E−01
	FPA	4.17E+01	1.08E+02	6.36E+01	1.51E+01
	GWO	1.03E−21	4.27E−20	7.27E−21	7.55E−21
	GWO−E	**9.23E−38**	**3.88E−35**	**4.31E−36**	**6.57E−36**
f_3	BA	1.32E+04	1.13E+05	3.56E+04	1.66E+04
	BFP	5.29E+04	2.01E+08	1.07E+05	3.06E+04
	DE	4.33E+04	1.83E+05	9.45E+04	3.15E+04
	FA	4.48E+02	3.27E+03	1.56E+03	6.42E+02

	FPA	4.67E+03	1.62E+04	1.02E+04	2.84E+03
	GWO	2.21E−12	5.09E−08	4.13E−09	9.99E−09
	GWO−E	**2.30E−44**	**9.13E−36**	**3.75E−37**	**1.36E−36**
f_4	BA	2.69E+01	6.31E+01	4.47E+01	8.29E+00
	BFP	6.29E+01	8.87E+01	8.30E+01	4.93E+00
	DE	6.29E+01	9.10E+01	8.52E+01	3.87E+00
	FA	7.50E+01	1.46E+00	3.09E−01	3.41E−01
	FPA	24.617	4.31E+01	3.39E+01	4.14E+00
	GWO	8.32E−10	2.44E−08	4.93E−09	4.55E−09
	GWO−E	**2.47E−28**	**3.82E−24**	**2.39E−25**	**6.48E−25**
f_5	BA	7.01E+05	3.91E+07	5.43E+06	5.78E+06
	BFP	4.15E+07	2.75E+08	1.76E+08	5.71E+07
	DE	1.58E+08	3.12E+08	2.30E+08	3.54E+07
	FA	2.70E+01	1.23E+03	9.07E+01	1.86E+02
	FPA	2.02E+05	4.90E+06	1.78E+06	8.98E+05
	GWO	25.216779	28.539182	26.6059739	0.747239
	GWO−E	25.3951	27.1625	26.5054	**0.5196**
f_6	BA	7.88E+03	2.02E+04	1.24E+04	3.13E+03
	BFP	2.79E+04	7.42E+04	5.84E+04	9.44E+03
	DE	3.88E+04	7.48E+04	6.33E+04	7.24E+03
	FA	6.00E−03	3.60E−02	1.42E−02	5.30E−03
	FPA	3.09E+03	8.40E+03	4.73E+03	1.08E+03
	GWO	**2.68E−05**	1.0087688	0.3778627	0.297462
	GWO−E	2.92E−05	1.2573	0.3493	0.2958
f_7	BA	1.2797557	9.2410578	3.71900484	1.796347
	BFP	34.624658	1.09E+02	77.5599737	15.43403
	DE	59.506099	1.50E+02	1.11E+02	22.30853
	FA	0.0069388	0.0936067	0.02891335	0.021279
	FPA	0.494059	1.9424172	1.20361744	0.376008
	GWO	3.39E−04	0.0024156	9.74E−04	4.13E−04
	GWO−E	**5.86E−06**	**3.38E−04**	**9.90E−05**	**8.37E−05**
f_8	BA	42.7863	1.71E+02	8.31E+01	3.07E+01
	BFP	2.55E+02	3.98E+02	3.40E+02	2.97E+01
	DE	3.55E+02	4.77E+02	4.21E+02	2.79E+01
	FA	1.31E+01	6.08E+01	2.81E+01	8.48E+00
	FPA	1.51E+02	2.54E+02	2.04E+02	1.64E+01
	GWO	0.00E+00	16.049901	2.48575127	4.046432
	GWO−E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	BA	1.18E+01	1.71E+01	1.47E+01	1.2524
	BFP	1.86E+01	2.03E+01	1.99E+01	0.3561
	DE	1.94E+01	2.08E+01	2.04E+01	0.2979
	FA	3.18E−02	1.06E−01	6.09E−02	0.0184
	FPA	1.18E+01	1.67E+01	1.47E+01	1.0161
	GWO	2.93E−14	4.35E−14	3.79E−14	4.37E−15
	GWO−E	**4.44E−15**	**7.99E−15**	**5.58E−15**	1.67E−15
f_{10}	BA	47.787431	3.00E+02	1.30E+02	46.22051
	BFP	3.08E+02	6.76E+02	5.27E+02	85.48164
	DE	3.79E+02	6.83E+02	5.64E+02	70.50904

	FA	0.0045138	0.0157903	0.00950704	0.002783
	FPA	16.372916	65.708356	42.0938233	12.65222
	GWO	0.00E+00	3.31E−02	0.00271886	0.006595
	GWO−E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{11}	BA	5.00E+03	2.66E+07	3.57E+06	4.58E+06
	BFP	1.61E+08	6.80E+08	4.39E+08	1.30E+08
	DE	3.19E+08	7.53E+08	5.31E+08	1.06E+08
	FA	1.43E−04	1.04E−01	4.80E−03	2.06E−02
	FPA	4.77E+01	8.08E+05	1.16E+05	1.66E+05
	GWO	0.0060901	0.0888267	0.03066472	0.017356
	GWO−E	**4.82E−06**	0.0596	1.98E−02	0.0101
f_{12}	BA	6.94E+05	5.53E+07	1.27E+07	1.03E+07
	BFP	4.27E+08	1.38E+09	9.10E+08	2.09E+08
	DE	4.24E+08	1.22E+09	9.30E+08	1.75E+08
	FA	9.64E−04	**1.65E−02**	**3.40E−03**	**2.30E−03**
	FPA	2.52E+05	1.08E+07	2.62E+06	2.07E+06
	GWO	0.0110256	0.7113301	0.30878473	0.151111
	GWO−E	4.30E−05	0.7803	2.50E−01	0.1634
f_{13}	BA	0.9980038	23.809434	9.04620727	7.162125
	BFP	1.323924	1.76E+02	19.4126067	25.38739
	DE	0.9980038	1.9920309	1.11728709	0.3263
	FA	0.9980038	6.0287541	1.81160971	0.931792
	FPA	0.9980038	0.9981212	0.99800827	**1.77E−05**
	GWO	0.9980038	12.670506	3.60743879	3.604444
	GWO−E	0.9980	10.7632	1.9812	2.3694
f_{14}	BA	3.65E−04	2.05E−02	3.6E−03	4.5E−03
	BFP	1.80E−03	1.14E−01	2.48E−03	2.25E−02
	DE	1.00E−03	1.80E−03	1.30E−03	3.14E−04
	FA	3.07E−04	1.6E−03	7.54E−04	3.34E−04
	FPA	1.00E−03	**1.10E−03**	**1.00E−03**	**1.97E−05**
	GWO	3.07E−04	0.0203633	0.00275114	0.006572
	GWO−E	3.07E−04	0.0012	6.74E−04	2.84E−04
f_{15}	BA	−1.0316	−2.15E−01	−9.99E−01	1.61E−01
	BFP	−1.0316	−1.09E−01	−8.59E−01	3.32E−01
	DE	−1.0316	−1.0306	−1.0316	**2.67E−16**
	FA	−1.0316	−1.0306	−1.0316	2.84E−09
	FPA	−1.0316	−1.0306	−1.0316	5.08E−08
	GWO	−1.031628	−1.031628	−1.0316285	1.37E−08
	GWO−E	−1.0316	−1.0315	−1.0315	3.27E−05
f_{16}	BA	0.3979	0.3979	0.3979	**3.87E−10**
	BFP	0.3979	1.4918	0.5087	2.28E−01
	DE	0.3979	0.3979	0.3979	3.36E−16
	FA	0.3979	0.3979	0.3979	2.29E−09
	FPA	0.3979	0.3979	0.3979	1.12E−05
	GWO	0.3978874	0.3981122	0.39789604	3.86E−05
	GWO−E	0.3979	0.3983	0.3979	6.55E−05
f_{17}	BA	3.00E+00	3.00E+00	3.00E+00	**4.48E−08**

	BFP	3.00E+00	8.40E+01	1.13E+01	1.56E+01
	DE	3.00E+00	3.00E+00	3.00E+00	3.64E−15
	FA	3.00E+00	3.00E+00	3.00E+00	2.11E−08
	FPA	3.00E+00	3.00E+00	3.00E+00	6.43E−09
	GWO	3.00E+00	3.00005	3.0000098	1.25E−05
	GWO−E	3.00E+00	3.0001	3	1.75E−05
f_{18}	BA	−3.862782	−3.862782	−3.8627821	2.94E−08
	BFP	−3.862782	−3.089763	−3.8092568	0.123292
	DE	−3.862782	−3.862782	−3.8627821	**3.07E−15**
	FA	−3.862782	−3.862782	−3.8627821	2.64E−09
	FPA	−3.862782	−3.862782	−3.8627821	3.57E−08
	GWO	−3.862782	−3.8549	−3.8618914	0.002107
	GWO−E	−3.8628	−3.8611	−3.8627	3.15E−04
f_{19}	BA	−3.321994	−3.203087	−3.2577882	0.059859
	BFP	−3.321939	−2.23138	−3.1032288	0.186281
	DE	−3.321977	−3.202857	−3.2169767	0.038173
	FA	−3.321995	−3.142032	−3.2607203	0.065013
	FPA	−3.320388	−3.210443	−3.281387	**0.022889**
	GWO	−3.321994	−3.078118	−3.2649538	0.075015
	GWO−E	−3.322	−3.1913	−3.2464	0.0599

Table5.9 Results of GWO-E compared with other algorithms for population size of 60

Case III (for population size of 90): Simulation results for all the algorithms for population size of 90 are shown in Table5.10. For f_1,f_2, f_3, f_4, only GWO-E gives near to optimum value. For f_5, GWO give very competitive results to GWO-E, although other algorithms fail to achieve near to optimum value. For f_6, GWO shows far better results as compared to BA, BFP, FA , FPA whereas GWO-E is competitive with respect to GWO. For f_7, GWO-E gives the best results. GWO-E is able to provide exact solution for function f_8 . For f_9, GWO-E has same standard deviation with GWO, and is better in terms of best, worst and average. For f_{10}, again GWO-E is better and it achieves exact solution. For f_{11}, GWO gives very competitive results but standard deviation of GWO-E is best. For f_{12}, GWO and GWO-E both have competitive results. For f_{13} and f_{14} FPA has good standard deviation and is found to provide better results for these fixed dimension functions. For f_{15}, f_{16}, f_{17} and f_{18} all algorithms provide comparable results, but DE has better performance with good standard deviation. FPA provides better standard deviation for fixed dimension function f_{19}.Thus for population size of 90, GWO-E is good for eleven function, FPA for three, DE for four and GWO for three.

Function	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	4.15E+03	1.49E+04	8.40E+03	2.48E+03
	BFP	3.43E+04	6.99E+04	5.69E+04	8.10E+03
	DE	3.21E+04	7.54E+04	6.12E+04	7.77E+03
	FA	5.73E−03	2.93E−02	1.37E−02	4.51E−03

	FPA	4.27E+03	9.24E+03	6.98E+03	1.29E+03
	GWO	4.40E−41	2.54E−38	1.67E−39	4.05E−39
	GWO−E	**2.81E−74**	**2.76E−69**	**9.24E−71**	**4.01E−70**
f_2	BA	7.65E+00	3.95E+05	7.95E+03	5.59E+04
	BFP	5.84E+01	8.48E+08	4.38E+07	1.41E+08
	DE	5.62E+01	4.86E+10	1.43E+09	7.17E+09
	FA	2.66E−01	7.83E−01	4.62E−01	1.18E−01
	FPA	4.96E+01	1.20E+02	7.51E+01	1.35E+01
	GWO	2.43E−24	6.05E−23	2.00E−23	1.45E−23
	GWO−E	**2.47E−39**	**2.60E−37**	**3.94E−38**	**5.62E−38**
f_3	BA	7.68E+03	5.34E+04	2.52E+04	9.94E+03
	BFP	3.17E+04	1.48E+05	9.82E+04	2.49E+04
	DE	5.56E+04	1.48E+05	9.38E+04	2.39E+04
	FA	3.48E+02	2.19E+03	1.10E+03	3.58E+02
	FPA	7.08E+03	2.39E+04	1.49E+04	4.70E+03
	GWO	1.91E−14	4.78E−10	5.77E−11	1.15E−10
	GWO−E	**6.31E−46**	**3.25E−38**	**1.61E−39**	**5.73E−39**
f_4	BA	2.27E+01	5.15E+01	3.64E+01	5.98E+00
	BFP	7.11E+01	8.86E+01	8.40E+01	3.54E+00
	DE	7.42E+01	8.95E+01	8.37E+01	3.46E+00
	FA	7.99E−02	2.19E+00	7.21E−01	5.92E−01
	FPA	28.905847	4.82E+01	3.94E+01	4.41E+00
	GWO	4.89E−11	1.23E−09	3.81E−10	3.00E−10
	GWO−E	**6.43E−30**	**2.39E−26**	**1.90E−27**	**4.36E−27**
f_5	BA	3.29E+04	5.28E+06	1.46E+06	1.16E+06
	BFP	7.64E+07	2.60E+08	1.83E+08	4.52E+07
	DE	1.30E+08	2.70E+08	2.12E+08	3.48E+07
	FA	2.66E+01	4.58E+02	7.20E+01	8.03E+01
	FPA	4.68E+05	1.02E+07	3.41E+06	1.91E+06
	GWO	25.228325	28.529666	26.5537577	0.668024
	GWO−E	25.2235	27.9085	2.64E+01	**0.5271**
f_6	BA	3.24E+03	1.67E+04	8.92E+03	2.82E+03
	BFP	3.10E+04	6.86E+04	5.64E+04	8.14E+03
	DE	4.25E+04	7.40E+04	6.20E+04	6.83E+03
	FA	5.22E−03	2.46E−02	1.32E−02	4.54E−03
	FPA	3.34E+03	1.10E+04	6.33E+03	1.50E+03
	GWO	**1.64E−05**	0.7452753	0.19055797	0.19291
	GWO−E	1.73E−05	7.46E−01	2.20E−01	2.06E−01
f_7	BA	0.5958407	4.7558428	1.86199658	0.84218
	BFP	28.362714	1.21E+02	70.3327057	17.86718
	DE	49.23489	1.33E+02	1.01E+02	19.01202
	FA	0.0036717	0.070721	0.02090585	0.01312
	FPA	0.9035215	4.6544705	1.92004564	0.716385
	GWO	1.64E−05	0.7452753	0.19055797	0.19291
	GWO−E	**2.75E−06**	**1.80E−04**	**5.89E−05**	**4.34E−05**
f_8	BA	36.816515	1.59E+02	7.28E+01	2.46E+01
	BFP	2.21E+02	3.57E+02	3.02E+02	3.43E+01
	DE	3.27E+02	4.54E+02	4.08E+02	2.68E+01

	FA	1.12E+01	4.52E+01	2.17E+01	6.34E+00
	FPA	1.81E+02	2.37E+02	2.06E+02	1.27E+01
	GWO	0.00E+00	13.42466	1.24917167	2.997451
	GWO−E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	BA	1.06E+01	1.58E+01	1.37E+01	1.108042
	BFP	1.84E+01	2.02E+01	1.96E+01	0.418443
	DE	1.89E+01	2.08E+01	2.04E+01	0.335019
	FA	3.02E−02	0.1295297	6.31E−02	0.019606
	FPA	1.38E+01	1.74E+01	1.60E+01	0.802644
	GWO	1.87E−14	4.00E−14	2.95E−14	3.83E−15
	GWO−E	**4.44E−15**	**7.99E−15**	**5.93E−15**	1.77E−15
f_{10}	BA	45.986908	1.91E+02	93.5047257	29.85408
	BFP	2.80E+02	6.22E+02	5.14E+02	80.13705
	DE	3.27E+02	6.65E+02	5.54E+02	62.70161
	FA	0.0049918	0.0166982	0.00958856	0.002851
	FPA	29.199295	93.227245	59.2610739	15.56441
	GWO	0.00E+00	0.0287057	0.00347344	0.006964
	GWO−E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{11}	BA	1.09E+01	7.07E+06	5.01E+05	1.19E+06
	BFP	5.89E+07	7.04E+08	3.95E+08	1.32E+08
	DE	3.01E+08	6.11E+08	4.68E+08	8.16E+07
	FA	1.66E−04	0.0041691	1.06E−03	0.001045
	FPA	3.86E+03	3.77E+06	5.33E+05	6.53E+05
	GWO	2.09E−06	0.0431057	0.01620568	0.010125
	GWO−E	1.46E−06	0.0314964	0.01345997	**0.008489**
f_{12}	BA	3.08E+05	5.24E+07	6.59E+06	1.04E+07
	BFP	2.09E+08	1.22E+09	7.67E+08	2.41E+08
	DE	5.46E+08	1.28E+09	9.28E+08	1.71E+08
	FA	0.001217	6.73E−03	3.18E−03	1.35E−03
	FPA	4.15E+05	1.68E+07	6.05E+06	4.09E+06
	GWO	**2.33E−05**	**0.5124585**	**0.16682047**	**0.138959**
	GWO−E	**2.23E−05**	**0.496**	**0.1499**	**0.1122**
f_{13}	BA	0.9980038	23.809434	6.73888678	5.757864
	BFP	0.9980038	22.901376	9.17415527	6.445822
	DE	0.9980038	1.9920309	1.03776492	0.196766
	FA	0.9980038	3.012527	1.71037589	0.534453
	FPA	0.9980038	0.9980126	0.99800449	**1.57E−06**
	GWO	0.9980038	10.763181	2.60943394	2.892441
	GWO−E	0.9980038	10.763181	1.6663851	1.99968
f_{14}	BA	3.08E−04	0.019419	0.0021082	0.003437
	BFP	0.001368	0.1222955	0.01826939	0.022049
	DE	0.0010125	0.0017941	0.00115504	2.62E−04
	FA	3.18E−04	0.0015993	6.41E−04	2.23E−04
	FPA	8.31E−04	0.0010502	0.00102589	**2.91E−05**
	GWO	3.07E−04	0.0203633	0.00203669	0.00547
	GWO−E	3.07E−04	0.0010377	6.58E−04	3.11E−04
f_{15}	BA	−1.031628	−1.031628	−1.0316285	5.37E−10
	BFP	−1.031628	−0.215464	−0.9663352	0.223668

	DE	−1.031628	−1.031628	−1.0316285	**2.24E−16**
	FA	−1.031628	−1.031628	−1.0316284	3.05E−09
	FPA	−1.031628	−1.031628	−1.0316284	3.19E−08
	GWO	−1.031628	−1.031628	−1.0316285	7.28E−09
	GWO−E	−1.031628	−1.031557	−1.0316174	1.74E−05
f_{16}	BA	0.3978874	0.3978874	0.39788736	2.02E−10
	BFP	0.3978874	0.6220505	0.41010229	0.038535
	DE	0.3978874	0.3978874	0.39788736	**3.36E−16**
	FA	0.3978874	0.3978874	0.39788736	2.53E−09
	FPA	0.3978874	0.3979012	0.39789028	2.88E−06
	GWO	0.3978874	0.3978874	0.39788736	9.00E−07
	GWO−E	0.3978874	0.4014623	0.39802149	5.27E−04
f_{17}	BA	3.00E+00	3.00E+00	3.00E+00	1.94E−08
	BFP	3.00E+00	3.00E+01	5.70E+00	8.18E+00
	DE	3.00E+00	3.00E+00	3.00E+00	**3.88E−15**
	FA	3.00E+00	3.00E+00	3.00E+00	2.48E−08
	FPA	3.00E+00	3.00E+00	3.00E+00	3.37E−09
	GWO	3.00E+00	3.0000206	3.00000428	4.51E−06
	GWO−E	3.00E+00	3.00E+00	3.00E+00	4.77E−06
f_{18}	BA	−3.862782	−3.862782	−3.8627821	2.50E−08
	BFP	−3.862782	−3.722205	−3.8522298	0.022964
	DE	−3.862782	−3.862782	−3.8627821	**3.09E−15**
	FA	−3.862782	−3.862782	−3.8627821	1.38E−09
	FPA	−3.862782	−3.862782	−3.8627821	3.27E−08
	GWO	−3.862782	−3.854899	−3.8619465	0.00228
	GWO−E	−3.862782	−3.86249	−3.862765	5.22E−05
f_{19}	BA	−3.321994	−3.203091	−3.2625456	0.060051
	BFP	−3.321994	−2.981417	−3.211743	0.095135
	DE	−3.321951	−3.177361	−3.2167819	0.039306
	FA	−3.321995	−3.119734	−3.2532662	0.067906
	FPA	−3.317744	−3.252178	−3.2857897	**0.014967**
	GWO	−3.321994	−3.137622	−3.2675317	0.062702
	GWO−E	−3.321993	−3.189232	−3.2438234	0.059307

Table 5.10 Results of GWO-E compared with other algorithms for population size of 90

Inference drawn: In this work, a new variant of GWO-E has been proposed which includes two new modifications over the original GWO. The proposed method is found to be better than basic GWO and other algorithms. For the population size of 30 GWO-E shows better results for eleven functions, FA achieved good results for four functions, BA and DE for one each, FPA for two each. For population size of 60, GWO-E is best for ten functions, FPA for three, DE and BA for two each, FA and GWO for one each. For population size of 90, GWO-E is good for eleven functions, FPA for three, DE for four and GWO achieves good results for three functions.

So, overall, among 57 functions, GWO-E is good for thirty-two functions, DE for seven functions, FA for five, FPA for eight functions, BA for three functions and GWO for four functions. Here BFP is found to be the worst algorithm. The total number of functions for all the three cases combined are 57, but the results are for 59 functions. This is because for some of the functions two or more algorithms provide same results and are counted more than once. This discussion shows that the proposed algorithm shows best performance among other state-of-the-art algorithms.

Now, for population size of any algorithm is concerned, it can be seen that with the increase in population size from 30 to 90 GWO-E provides better global optimum value. But the number of functions for which GWO-E is better over the other remains almost same. For population size of 30 and 60, GWO-E is good for twelve. For population size of 30, GWO-E provides better results for thirteen functions. It can be inferred that the change is modest whereas function evaluations doubles for the case of 60 population size and triples for 90 population size. Overall, we can say that population size of 30 can be chosen as the best possible case with minimum number of function evaluations.

B. *Effect of dimensions*

In this sub-section five different dimension size of 30, 50, 100, 200 and 500 are used to test the performance of GWO-E and GWO. In the above section, population size of 30 is considered as best so results of GWO and GWO-E are taken with population size of 30 and are compared with other algorithms such as BA, BFP, DE, FA and FPA (population size of 30). Out of nineteen functions used in this paper, seven are of fixed dimension. Hence performance is evaluated on only twelve functions (detail of these functions has been given in the Table 5.2, 5.3 and 5.4). Table 5.8 provides the comparison for *d=30,* Table 5.11 shows the comparison for *d=50,* Table 5.12 for *d=100,* Table 5.13 for *d=200* and Table 5.14 for 500.

Case I (for 30-D): The results for dimension size of 30 are given in Table 5.8 and has already been discussed under the effect of population size of 30. In this case, out of twelve benchmark functions (which are of variable dimension), GWO-E is good for ten functions and FA is good for two functions.

Case II (for 50-D): The results for dimension size of 50 are given in Table 5.11. For f_1, f_2, f_3 and f_4 only GWO-E is able to achieve near to optimum solution. GWO shows good results for function f_5, though results of GWO-E are quite comparable. FA shows better performance among all other algorithms for function f_6. For f_7, GWO-E provides best results. GWO-E is only algorithm which provides exact solution for function f_8. For f_9, results provided by GWO-E are best as compared with other algorithms. For f_{10}, again GWO-E is the best algorithm and it provides exact optimum value. FA provides better results for f_{11} and f_{12}. Thus, for dimension size of 50, GWO-E provides good results for eight functions, FA for three and GWO for one.

Function	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	1.72E+04	8.07E+04	3.70E+04	1.34E+04
	BFP	7.05E+04	1.43E+05	1.16E+05	1.74E+04
	DE	9.95E+04	1.47E+05	1.25E+05	1.02E+04
	FA	5.00E-02	1.38E-01	8.83E-02	1.93E-02
	FPA	3.54E+03	9.93E+03	5.97E+03	1.26E+03
	GWO	5.17E-21	8.05E-19	1.01E-19	1.56E-19
	GWO-E	**2.31E-54**	**2.88E-49**	**1.64E-50**	**4.46E-50**
f_2	BA	6.67E+01	7.08E+13	1.42E+12	1.00E+13
	BFP	1.77E+06	1.56E+20	9.13E+18	3.11E+19
	DE	1.12E+02	4.20E+21	8.50E+19	5.94E+20
	FA	1.00E+00	1.09E+01	2.98E+00	2.19E+00
	FPA	4.17E+01	5.96E+02	7.68E+01	7.65E+01
	GWO	6.60E-13	8.72E-12	2.35E-12	1.37E-12
	GWO-E	**3.35E-29**	**6.40E-27**	**1.11E-27**	**1.47E-27**
f_3	BA	4.64E+04	3.18E+05	1.62E+05	6.64E+04
	BFP	1.76E+05	7.07E+05	4.12E+05	1.30E+05
	DE	8.69E+04	4.38E+05	2.23E+05	7.82E+04
	FA	6.40E+03	2.30E+04	1.35E+04	3.76E+03
	FPA	6.66E+03	2.88E+04	1.35E+04	4.66E+03
	GWO	1.37E-12	3.09E+00	4.05E-01	7.12E-01
	GWO-E	**8.79E-32**	**9.54E-19**	**1.93E-20**	**1.35E-19**
f_4	BA	3.45E+01	7.79E+01	5.97E+01	9.69E+00
	BFP	7.02E+01	9.54E+01	9.06E+01	4.54E+00
	DE	8.32E+01	9.58E+01	9.19E+01	2.85E+00
	FA	3.51E+00	1.48E+01	8.48E+00	2.70E+00
	FPA	22.0128	3.75E+01	3.04E+01	3.71E+00
	GWO	5.75E-05	2.24E-03	4.21E-04	4.25E-04
	GWO-E	**5.35E-21**	**8.06E-17**	**6.58E-18**	**1.49E-17**
f_5	BA	1.06E+07	2.08E+08	5.01E+07	3.77E+07
	BFP	1.85E+08	6.19E+08	4.52E+08	1.17E+08
	DE	4.02E+08	6.02E+08	5.09E+08	4.81E+07
	FA	5.22E+01	2.08E+03	3.15E+02	4.92E+02
	FPA	7.30E+05	3.79E+06	1.77E+06	7.24E+05
	GWO	**45.83931**	48.677223	47.2643948	0.831087
	GWO-E	46.0067	48.638144	47.2761026	0.701336
f_6	BA	2.17E+04	7.00E+04	3.75E+04	1.25E+04
	BFP	7.49E+04	1.39E+05	1.15E+05	1.59E+04
	DE	1.11E+05	1.45E+05	1.26E+05	8.94E+03
	FA	**5.06E-02**	**1.34E-01**	**8.88E-02**	**2.08E-02**
	FPA	3.46E+03	1.16E+04	6.47E+03	1.37E+03
	GWO	1.253469	4.2713615	2.66105847	0.622134
	GWO-E	1.413594	3.7442948	2.46263713	0.538366
f_7	BA	5.88262	44.739227	20.6130127	8.240914
	BFP	98.73874	4.23E+02	3.00E+02	86.4321

	DE	2.72E+02	5.25E+02	4.02E+02	56.55539
	FA	0.032279	0.2382175	0.09800371	0.04612
	FPA	0.820147	4.8816715	1.8472293	0.876306
	GWO	8.68E-04	0.0098949	0.00351309	1.66E-03
	GWO-E	**2.26E-05**	**6.25E-04**	**1.90E-04**	**1.39E-04**
f_8	BA	66.67561	3.87E+02	1.38E+02	4.98E+01
	BFP	5.42E+02	7.76E+02	7.05E+02	5.09E+01
	DE	6.94E+02	8.54E+02	7.74E+02	4.11E+01
	FA	6.02E+01	1.53E+02	9.22E+01	1.72E+01
	FPA	3.44E+02	4.20E+02	3.79E+02	1.84E+01
	GWO	1.71E-13	19.639672	4.97768101	4.694277
	GWO-E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	BA	1.41E+01	1.86E+01	1.64E+01	1.072935
	BFP	1.91E+01	2.10E+01	2.05E+01	0.381641
	DE	1.97E+01	2.10E+01	2.07E+01	0.284926
	FA	9.84E-02	0.5210242	2.14E-01	0.084105
	FPA	1.11E+01	1.46E+01	1.28E+01	0.795616
	GWO	1.32E-11	1.12E-10	4.16E-11	2.51E-11
	GWO-E	**4.44E-15**	**2.22E-14**	**8.14E-15**	**3.86E-15**
f_{10}	BA	1.67E+02	6.67E+02	3.46E+02	95.32744
	BFP	5.72E+02	1.25E+03	1.02E+03	1.76E+02
	DE	9.70E+02	1.36E+03	1.13E+03	8.07E+01
	FA	0.013709	0.0583404	0.02881065	0.0087
	FPA	38.55057	82.280323	58.1605075	11.15001
	GWO	0	0.0404721	0.00591999	0.009627
	GWO-E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{11}	BA	2.58E+06	2.32E+08	4.97E+07	4.78E+07
	BFP	1.37E+08	1.51E+09	1.04E+09	3.23E+08
	DE	8.45E+08	1.51E+09	1.20E+09	1.58E+08
	FA	**7.67E-04**	1.3378571	3.47E-01	0.419352
	FPA	2.87E+01	2.32E+05	2.96E+04	5.20E+04
	GWO	0.059414	0.2072429	0.11670036	0.038029
	GWO-E	0.041472	0.2077608	0.09299693	0.035693
f_{12}	BA	8.16E+06	6.21E+08	1.64E+08	1.45E+08
	BFP	5.26E+08	2.87E+09	2.04E+09	4.97E+08
	DE	1.84E+09	2.84E+09	2.29E+09	2.47E+08
	FA	**0.007937**	3.48E-02	1.80E-02	5.75E-03
	FPA	5.58E+04	4.22E+06	1.46E+06	1.12E+06
	GWO	1.373187	3.1583911	2.15408856	0.358681
	GWO-E	1.102296	2.746508	1.94038151	0.341076

Table 5.11 Results of GWO-E compared with other algorithms for dimension size of 50

Case III (for 100-D): Table 5.12 shows the simulation results for dimension size of 100. For f_1, f_2, f_3 and f_4 only GWO-E is able to attain global optimum value while other algorithms got stuck at some local solutions. For f_5, GWO and GWO-E both provide very competitive results. For f_6, FA outperforms all the other algorithms. GWO-E provides best results for f_7, and it gives exact optimum value for function f_8. For f_9, although results of GWO are comparable with GWO-E, but the best results are obtained by GWO-E. For f_{10}, only GWO-E gives exact optimum value. Results of GWO and GWO-E are quite comparable for functions f_{11} and f_{12}, but the best results are achieved by GWO. Thus, for dimension size of 100, GWO-E provides better results for nine functions, FA for one and GWO for three.

Function	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	4.37E+04	1.20E+05	8.14E+04	1.79E+04
	BFP	1.28E+05	2.97E+05	2.41E+05	4.60E+04
	DE	2.42E+05	2.99E+05	2.73E+05	1.39E+04
	FA	3.20E-01	3.44E+00	8.22E-01	4.91E-01
	FPA	1.21E+04	2.74E+04	1.74E+04	3.29E+03
	GWO	1.21E-18	1.97E-17	5.87E-18	4.46E-18
	GWO-E	**9.35E-46**	**1.39E-40**	**7.16E-42**	**2.34E-41**
f_2	BA	1.43E+02	1.16E+32	2.36E+30	1.65E+31
	BFP	5.87E+27	8.16E+47	1.66E+46	1.15E+47
	DE	4.39E+16	2.78E+50	2.05E+49	6.61E+49
	FA	1.28E+01	7.11E+01	3.49E+01	1.35E+01
	FPA	1.02E+02	1.94E+02	1.28E+02	1.63E+01
	GWO	1.79E-11	7.73E-11	3.98E-11	1.20E-11
	GWO-E	**5.29E-25**	**1.46E-22**	**2.59E-23**	**3.13E-23**
f_3	BA	1.90E+05	1.24E+06	6.18E+05	2.62E+05
	BFP	7.28E+05	2.86E+06	1.48E+06	4.74E+05
	DE	3.03E+05	2.13E+06	8.98E+05	3.32E+05
	FA	4.64E+04	9.53E+04	6.67E+04	1.21E+04
	FPA	3.28E+04	9.16E+04	5.80E+04	1.51E+04
	GWO	3.08E-01	1.88E+02	2.51E+01	4.11E+01
	GWO-E	**1.50E-24**	**4.75E-11**	**9.74E-13**	**6.71E-12**
f_4	BA	5.02E+01	8.53E+01	6.46E+01	8.03E+00
	BFP	8.35E+01	9.77E+01	9.42E+01	2.86E+00
	DE	9.33E+01	9.78E+01	9.63E+01	1.11E+00
	FA	1.92E+01	3.14E+01	2.48E+01	2.84E+00
	FPA	29.47324	4.63E+01	3.65E+01	3.84E+00
	GWO	2.00E-03	1.64E-01	1.72E-02	2.38E-02
	GWO-E	**5.99E-16**	**3.02E-09**	**7.93E-11**	**4.29E-10**
f_5	BA	1.81E+07	4.15E+08	1.15E+08	6.73E+07
	BFP	4.90E+08	1.35E+09	1.01E+09	2.54E+08
	DE	9.30E+08	1.35E+09	1.19E+09	1.02E+08
	FA	4.46E+02	2.77E+04	2.65E+03	5.19E+03
	FPA	2.98E+06	1.07E+07	6.58E+06	2.25E+06
	GWO	**95.2939**	98.3588	97.3220	0.8688
	GWO-E	95.95591	98.512496	97.747780	0.804952
f_6	BA	4.69E+04	1.26E+05	8.01E+04	2.25E+04
	BFP	1.66E+05	3.02E+05	2.49E+05	4.41E+04
	DE	2.32E+05	3.05E+05	2.72E+05	1.60E+04
	FA	**2.72E-01**	5.06E+00	9.24E-01	**8.35E-01**
	FPA	1.04E+04	3.17E+04	1.69E+04	3.70E+03
	GWO	5.014002	9.163223	6.5428166	1.018899
	GWO-E	7.602999	11.577198	9.6534297	0.950118

f_7	BA	22.85242	1.18E+02	59.241376	21.72228
	BFP	5.46E+02	2.07E+03	1.52E+03	3.51E+02
	DE	1.51E+03	2.34E+03	1.89E+03	1.80E+02
	FA	0.262021	1.2014764	0.72751692	0.238535
	FPA	3.717874	24.873526	10.5227137	4.119925
	GWO	8.07E-04	0.0046946	0.00223838	9.73E-04
	GWO-E	**8.29E-06**	0.0020681	2.93E-04	3.57E-04
f_8	BA	1.66E+02	4.47E+02	2.82E+02	6.32E+01
	BFP	1.28E+03	1.66E+03	1.49E+03	8.41E+01
	DE	1.51E+03	1.73E+03	1.63E+03	4.97E+01
	FA	2.09E+02	3.84E+02	2.83E+02	3.78E+01
	FPA	7.78E+02	9.36E+02	8.68E+02	3.49E+01
	GWO	9.09E-13	25.090259	5.2589439	5.692318
	GWO-E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	BA	1.50E+01	1.88E+01	1.68E+01	0.875794
	BFP	1.92E+01	2.10E+01	2.07E+01	0.353484
	DE	2.08E+01	2.11E+01	2.10E+01	0.068484
	FA	1.06E+00	2.6007756	1.66E+00	0.390207
	FPA	1.22E+01	1.44E+01	1.33E+01	0.566559
	GWO	1.07E-10	5.45E-10	2.74E-10	9.01E-11
	GWO-E	**4.44E-15**	**4.00E-14**	**1.09E-14**	**6.29E-15**
f_{10}	BA	3.45E+02	1.16E+03	6.81E+02	1.88E+02
	BFP	1.08E+03	2.74E+03	2.27E+03	3.91E+02
	DE	2.02E+03	2.67E+03	2.45E+03	1.27E+02
	FA	0.128595	0.3215836	0.21346648	0.051305
	FPA	96.13075	2.21E+02	1.48E+02	24.01681
	GWO	0.00E+00	0.0162258	0.00120008	0.004121
	GWO-E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{11}	BA	2.08E+07	6.53E+08	1.74E+08	1.47E+08
	BFP	6.25E+08	3.30E+09	2.26E+09	8.19E+08
	DE	2.55E+09	3.38E+09	2.90E+09	2.14E+08
	FA	3.68E+00	12.033226	7.17E+00	1.846736
	FPA	5.48E+03	1.12E+06	4.07E+05	3.04E+05
	GWO	**0.09828**	**0.2802572**	**0.14982981**	**0.041154**
	GWO-E	0.148274	0.3328462	0.23377537	0.04126
f_{12}	BA	3.19E+07	9.82E+08	3.66E+08	2.10E+08
	BFP	1.20E+09	6.32E+09	4.57E+09	1.32E+09
	DE	4.45E+09	6.08E+09	5.28E+09	4.51E+08
	FA	59.01572	1.75E+02	1.10E+02	2.79E+01
	FPA	1.66E+06	1.40E+07	6.19E+06	3.28E+06
	GWO	**4.052517**	**6.5688199**	**5.11087125**	**0.459451**
	GWO-E	4.697778	7.2607004	6.37636211	0.489148

Table 5.12 Results of GWO-E compared with other algorithms for dimension size of 100

Case IV (for 200-D): The results of various algorithms for dimension size of 200 are presented in the Table 5.13. For f_1, f_2, f_3 and f_4 GWO-E is good in terms of best, worst, average and standard deviation. For f_5, results of GWO and GWO-E are highly competitive. GWO provides better results as compared to other techniques for f_6, however the results of GWO-E are comparable to GWO. For f_7, f_8, f_9 GWO-E provides better results. For f_{10}, GWO-E gives exact solution. For f_{11}, results of GWO and GWO-E are highly competitive. GWO is better in terms of best, worst and average for f_{12} although standard deviation is same with GWO-E. Thus, for dimension size of 200 GWO-E gives good results for ten functions, GWO for four functions.

Function	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	9.13E+04	3.18E+05	1.58E+05	4.51E+04
	BFP	2.85E+05	6.14E+05	5.26E+05	9.61E+04
	DE	5.22E+05	6.18E+05	5.80E+05	1.94E+04
	FA	1.76E+03	7.28E+03	3.61E+03	1.27E+03
	FPA	2.60E+04	4.86E+04	3.71E+04	5.42E+03
	GWO	1.43E-11	1.12E-10	4.51E-11	2.08E-11
	GWO-E	**2.56E-40**	**9.80E-33**	**2.16E-34**	**1.38E-33**
f_2	BA	2.75E+02	7.86E+64	3.11E+63	1.54E+64
	BFP	8.64E+57	5.39E+101	1.27E+100	7.66E+100
	DE	9.31E+77	3.44E+105	1.17E+104	5.42E+104
	FA	1.39E+02	2.65E+02	1.87E+02	2.80E+01
	FPA	2.21E+02	3.24E+02	2.59E+02	2.03E+01
	GWO	1.93E-07	5.82E-07	3.39E-07	9.36E-08
	GWO-E	**4.89E-22**	**5.54E-19**	**3.31E-20**	**8.00E-20**
f_3	BA	6.03E+05	8.12E+06	2.73E+06	1.36E+06
	BFP	2.67E+06	1.19E+07	6.11E+06	1.88E+06
	DE	1.58E+06	8.11E+06	3.57E+06	1.35E+06
	FA	2.01E+05	4.06E+05	2.87E+05	5.58E+04
	FPA	1.12E+05	4.25E+05	2.38E+05	6.97E+04
	GWO	3.21E+02	1.43E+04	4.08E+03	2.92E+03
	GWO-E	**2.26E-21**	**1.85E+03**	**3.70E+01**	**2.62E+02**
f_4	BA	4.77E+01	8.80E+01	6.80E+01	7.91E+00
	BFP	8.54E+01	9.89E+01	9.66E+01	2.70E+00
	DE	9.66E+01	9.91E+01	9.80E+01	5.52E-01
	FA	3.44E+01	4.90E+01	4.23E+01	3.16E+00
	FPA	31.05424	4.96E+01	4.11E+01	4.56E+00
	GWO	2.68E+00	2.30E+01	9.94E+00	5.08E+00
	GWO-E	**8.37E-10**	2.69E+01	5.76E-01	3.80E+00
f_5	BA	9.21E+07	6.53E+08	2.10E+08	1.09E+08
	BFP	5.68E+08	2.85E+09	2.10E+09	6.74E+08
	DE	2.17E+09	2.92E+09	2.64E+09	1.58E+08
	FA	2.07E+05	2.11E+06	7.31E+05	3.92E+05
	FPA	8.28E+06	3.53E+07	1.71E+07	4.68E+06
	GWO	**1.95E+02**	1.98E+02	1.97E+02	0.67151273
	GWO-E	1.97E+02	1.98E+02	1.98E+02	0.47987971
f_6	BA	8.81E+04	2.49E+05	1.47E+05	3.61E+04
	BFP	2.41E+05	6.24E+05	5.27E+05	9.45E+04
	DE	5.41E+05	6.28E+05	5.82E+05	1.98E+04
	FA	1.46E+03	7.06E+03	3.17E+03	1.13E+03

	FPA	2.80E+04	5.11E+04	3.67E+04	5.06E+03
	GWO	**19.49147**	**25.140207**	**22.4429322**	1.29331428
	GWO-E	24.67028	30.703554	27.858586	1.39177792
f_7	BA	67.23097	2.75E+02	1.44E+02	47.2734349
	BFP	2.75E+03	8.84E+03	6.85E+03	1.86E+03
	DE	7.24E+03	9.64E+03	8.48E+03	6.04E+02
	FA	5.541175	21.592277	13.6989028	3.62879114
	FPA	22.68828	88.032376	52.0311923	16.1010938
	GWO	2.05E-03	0.0107525	0.00572083	1.88E-03
	GWO-E	**1.30E-05**	**0.0023095**	**3.57E-04**	**4.59E-04**
f_8	BA	6.28E+02	1.18E+03	8.64E+02	1.15E+02
	BFP	2.45E+03	3.42E+03	3.17E+03	2.23E+02
	DE	3.21E+03	3.57E+03	3.42E+03	70.7248954
	FA	7.06E+02	1.10E+03	8.46E+02	87.6716941
	FPA	1.78E+03	2.01E+03	1.89E+03	48.945585
	GWO	2.81E-07	25.52919	12.0229622	5.84509863
	GWO-E	**0.00E+00**	**2.27E-13**	**4.55E-15**	**3.22E-14**
f_9	BA	1.55E+01	1.87E+01	1.69E+01	0.84289534
	BFP	1.94E+01	2.10E+01	2.07E+01	0.39239328
	DE	2.09E+01	2.12E+01	2.11E+01	0.04336134
	FA	5.19E+00	8.247417	6.97E+00	0.69275542
	FPA	1.26E+01	1.49E+01	1.37E+01	0.56520248
	GWO	2.71E-07	7.41E-07	4.82E-07	1.27E-07
	GWO-E	**4.44E-15**	**4.00E-14**	**1.39E-14**	**4.00E-14**
f_{10}	BA	7.36E+02	2.28E+03	1.46E+03	3.29E+02
	BFP	2.29E+03	5.50E+03	4.57E+03	8.69E+02
	DE	4.81E+03	5.68E+03	5.24E+03	1.61E+02
	FA	9.43559	60.681404	33.6559513	11.0644744
	FPA	2.65E+02	4.39E+02	3.40E+02	45.8115932
	GWO	2.27E-12	0.0285367	0.00137277	0.00563153
	GWO-E	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_{11}	BA	3.73E+07	1.03E+09	2.93E+08	2.11E+08
	BFP	2.05E+09	6.90E+09	5.48E+09	1.58E+09
	DE	5.56E+09	7.04E+09	6.26E+09	3.75E+08
	FA	2.51E+01	72.491046	4.54E+01	12.121195
	FPA	1.42E+05	7.36E+06	1.84E+06	1.54E+06
	GWO	**0.263405**	0.4584462	0.35946963	0.04424853
	GWO-E	0.350539	0.5385359	0.4347649	0.04121243
f_{12}	BA	1.75E+08	2.67E+09	9.33E+08	6.16E+08
	BFP	1.75E+09	1.27E+10	8.55E+09	3.39E+09
	DE	9.97E+09	1.35E+10	1.18E+10	6.08E+08
	FA	1.66E+03	3.34E+05	5.80E+04	7.00E+04
	FPA	5.09E+06	4.84E+07	2.11E+07	1.06E+07
	GWO	**13.28906**	**15.629881**	**14.8092821**	0.54349227
	GWO-E	15.11231	17.666914	16.1974725	0.52201895

Table 5.13 Results of GWO-E compared with other algorithms for dimension size of 200

Case V (for 500-D): Simulation results for this case are presented in Table 5.14. For functions f_1, f_2, f_3 and f_4 results of GWO-E are far better than the other algorithms while other algorithms are not even near to global optimum solution. For f_5, GWO and GWO-E are found to be highly competitive. Results of GWO are comparable with GWO-E in terms of best, worst and average for function f_6 but the standard deviation of GWO-E is better. For f_8, GWO-E gives zero as a global optimum value and also shows good results for worst, average and standard deviation. For f_9 and f_{10} GWO-E provides best results. Both GWO and GWO-E gives competitive results for f_{11}. For f_{12}, although results of GWO are comparable with GWO-E but GWO-E has better performance. Thus, for dimension size of 500, GWO-E outperforms all other algorithms for twelve functions and GWO provides good results for one function.

Function	Algorithm	Best	Worst	Mean	Standard deviation
f_1	BA	2.17E+05	6.52E+05	3.90E+05	8.42E+04
	BFP	5.99E+05	1.58E+06	1.36E+06	2.87E+05
	DE	1.43E+06	1.59E+06	1.53E+06	3.26E+04
	FA	7.89E+04	1.36E+05	1.06E+05	1.37E+04
	FPA	7.31E+04	1.52E+05	1.00E+05	1.47E+04
	GWO	5.81E-04	3.58E-03	1.52E-03	5.72E-04
	GWO-E	**1.74E-33**	**1.14E-26**	**2.53E-28**	**1.62E-27**
f_2	BA	1.02E+35	5.31E+148	1.06E+147	7.50E+147
	BFP	9.52E+143	1.92E+267	3.94E+265	6.48E+150
	DE	8.24E+249	3.55E+273	7.15E+271	8.58E+140
	FA	7.57E+02	1.03E+03	8.81E+02	6.45E+01
	FPA	5.18E+02	7.41E+02	6.18E+02	4.19E+01
	GWO	8.43E-03	1.59E-02	1.11E-02	1.58E-03
	GWO-E	**5.53E-19**	**1.98E-16**	**1.36E-17**	**3.15E-17**
f_3	BA	7.58E+06	4.22E+07	1.85E+07	7.54E+06
	BFP	1.76E+07	7.75E+07	3.62E+07	1.23E+07
	DE	7.87E+06	5.29E+07	2.24E+07	9.73E+06
	FA	1.27E+06	2.36E+06	1.81E+06	2.83E+05
	FPA	6.62E+05	2.91E+06	1.58E+06	4.32E+05
	GWO	1.76E+05	4.63E+05	3.13E+05	6.91E+04
	GWO-E	**8.44E-15**	**2.19E+05**	**5.15E+03**	**3.11E+04**
f_4	BA	6.30E+01	9.03E+01	7.49E+01	6.40E+00
	BFP	8.97E+01	9.94E+01	9.80E+01	1.85E+00
	DE	9.87E+01	9.96E+01	9.93E+01	2.00E-01
	FA	5.56E+01	9.17E+01	6.56E+01	1.05E+01
	FPA	39.6108019	6.61E+01	4.83E+01	5.29E+00
	GWO	5.44E+01	7.66E+01	6.49E+01	5.01E+00
	GWO-E	**3.66E-03**	7.47E+01	5.11E+01	2.43E+01
f_5	BA	2.50E+08	1.03E+09	5.27E+08	2.11E+08
	BFP	2.53E+09	7.48E+09	5.84E+09	1.78E+09
	DE	6.50E+09	7.49E+09	7.06E+09	2.20E+08
	FA	4.23E+07	9.33E+07	6.79E+07	1.21E+07

	FPA	2.18E+07	1.24E+08	5.13E+07	1.89E+07
	GWO	4.97E+02	4.99E+02	4.98E+02	0.27440
	GWO-E	**4.96E+02**	4.98E+02	4.98E+02	0.24566
f_6	BA	2.15E+05	6.95E+05	3.71E+05	8.38E+04
	BFP	7.90E+05	1.57E+06	1.33E+06	2.58E+05
	DE	1.45E+06	1.59E+06	1.53E+06	3.13E+04
	FA	8.60E+04	1.31E+05	1.07E+05	1.08E+04
	FPA	7.82E+04	1.38E+05	1.04E+05	1.40E+04
	GWO	88.3624667	95.162215	91.4169273	1.83684897
	GWO-E	**88.0799**	96.06156	92.3968923	1.67088846
f_7	BA	2.00E+02	3.11E+03	9.46E+02	5.95E+02
	BFP	1.59E+04	6.09E+04	5.07E+04	1.21E+04
	DE	5.20E+04	6.34E+04	5.78E+04	2.44E+03
	FA	4.24E+02	9.73E+02	7.00E+02	1.27E+02
	FPA	2.21E+02	6.83E+02	3.71E+02	1.08E+02
	GWO	2.21E-02	0.0828937	0.04611053	1.38E-02
	GWO-E	**1.44E-05**	0.0081584	5.02E-04	1.23E-03
f_8	BA	2.79E+03	4.82E+03	3.39E+03	3.67E+02
	BFP	6.43E+03	8.81E+03	8.25E+03	6.23E+02
	DE	8.47E+03	9.02E+03	8.81E+03	1.20E+02
	FA	2.98E+03	3.62E+03	3.25E+03	1.58E+02
	FPA	4.84E+03	5.27E+03	5.05E+03	8.97E+01
	GWO	2.38E+01	1.22E+02	70.252877	20.40803
	GWO-E	**0.00E+00**	**1.82E-12**	**1.09E-13**	**3.51E-13**
f_9	BA	1.62E+01	1.87E+01	1.77E+01	0.6107146
	BFP	1.96E+01	2.11E+01	2.09E+01	0.2942813
	DE	2.11E+01	2.12E+01	2.11E+01	0.0260643
	FA	1.31E+01	15.108151	1.41E+01	0.4437928
	FPA	1.25E+01	1.49E+01	1.38E+01	0.4758649
	GWO	1.24E-03	2.60E-03	1.91E-03	2.97E-04
	GWO-E	**4.44E-15**	**9.33E-14**	**2.50E-14**	**1.92E-14**
f_{10}	BA	2.26E+03	5.46E+03	3.57E+03	8.27E+02
	BFP	6.91E+03	1.41E+04	1.22E+04	2.37E+03
	DE	1.30E+04	1.44E+04	1.38E+04	2.96E+02
	FA	7.06E+02	1.24E+03	9.49E+02	1.06E+02
	FPA	7.26E+02	1.26E+03	9.28E+02	1.15E+02
	GWO	8.88E-05	0.1066056	0.01139253	0.03112046
	GWO-E	**0**	**1.11E-16**	**4.44E-18**	**2.20E-17**
f_{11}	BA	9.79E+07	2.17E+09	7.28E+08	5.09E+08
	BFP	2.57E+09	1.85E+10	1.47E+10	5.27E+09
	DE	1.54E+10	1.88E+10	1.75E+10	5.79E+08
	FA	1.60E+06	3.59E+07	9.25E+06	5.32E+06
	FPA	2.03E+06	2.93E+07	9.65E+06	5.78E+06
	GWO	0.65964605	0.9015555	0.75282598	0.05066956
	GWO-E	**0.5740446**	0.7662295	0.65933717	0.03733679
f_{12}	BA	5.04E+08	6.66E+09	2.01E+09	1.13E+09
	BFP	4.54E+09	3.44E+10	2.41E+10	9.04E+09
	DE	2.92E+10	3.39E+10	3.22E+10	1.11E+09
	FA	3.92E+07	2.02E+08	1.04E+08	3.66E+07

	FPA	3.62E+07	1.97E+08	7.28E+07	3.04E+07
	GWO	47.894945	54.338441	50.8833765	1.34359959
	GWO-E	**45.576715**	**48.307064**	**47.1228215**	**0.59372251**

Table 5.14 Results of GWO-E compared with other algorithms for dimension size of 500

Inference drawn: GWO-E is found to be better than GWO and other algorithms for different dimension sizes. For dimension size of 30, GWO-E is good for ten functions, FA for two and GWO for none. For dimension size of 50, GWO-E provides optimum results for eight functions, FA for three and GWO for one. For dimension of 100, GWO-E provides better results for nine functions, GWO for three and FA for one. In case of dimension size of 200, GWO-E gives good results for ten functions, GWO for three functions. For large dimension size of 500, GWO-E outperforms all other algorithms for twelve functions and GWO gives good results for one function. For all the five cases, GWO-E provides better performance for forty-nine functions, GWO for seven functions, FA for six functions while BA, BFP, DE and FPA fails to provide good result. Under these cases, total number of function used are 60 while results are for 62 functions. This shows that the proposed algorithm shows better performance among other state-of-the-art algorithms. With the increase in dimension size, the number of variables and hence complexity of the problem increase but still the performance of GWO-E does not deteriorate significantly.

C. Statistical Testing

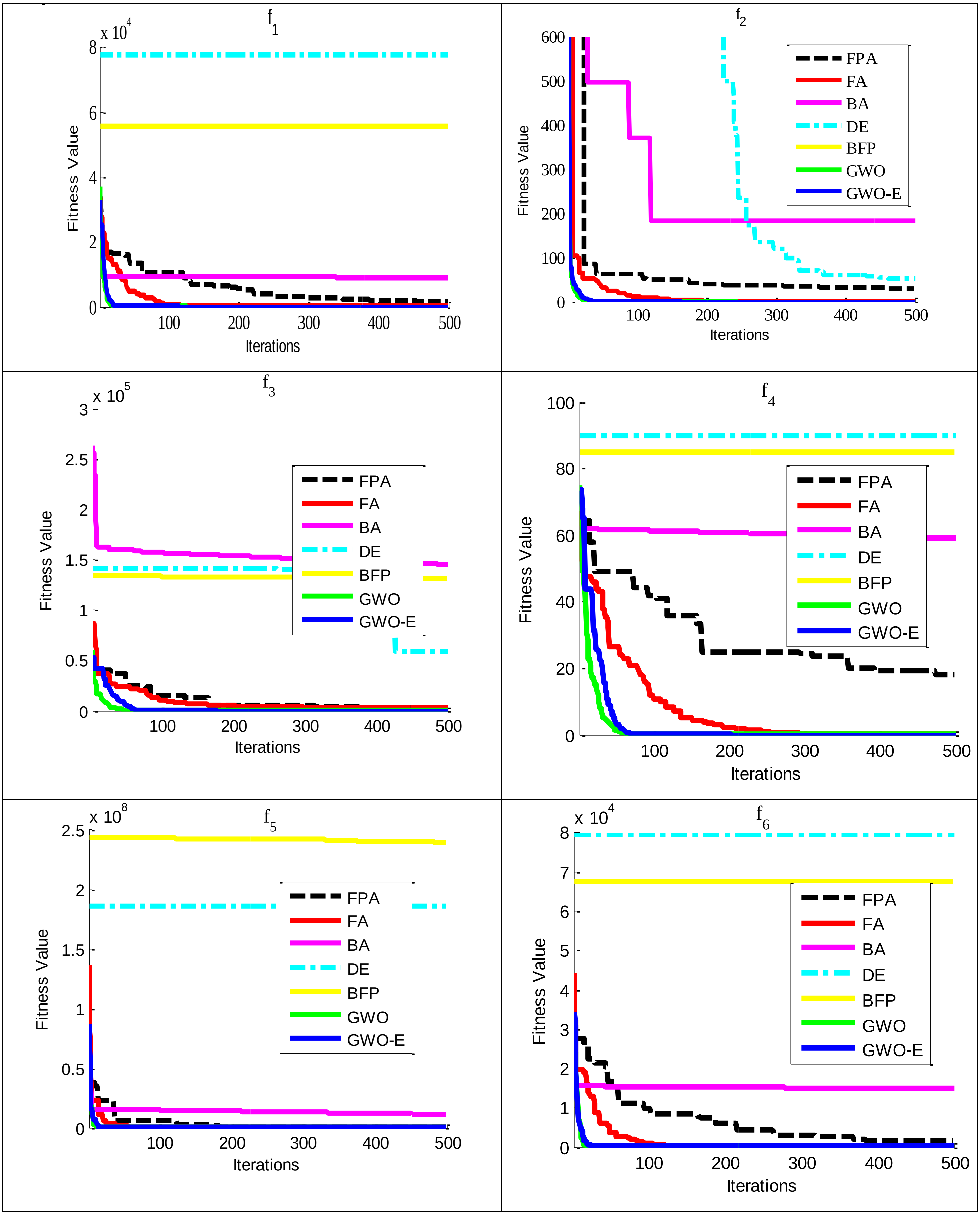
The Wilcoxon's rank-sum test [156] is conducted to evaluate the performance of the proposed algorithm with other state-of-the-art algorithms. This is a pairwise test used to find the significant difference between the two algorithms. This test is performed with a significant level of 0.05 i.e. the algorithm is significantly different from other algorithm if p- value is < 0.05. Statistical results are given in Table 5.15 and the results are taken for population size of 30 and dimension size of 30. The test is done by comparing GWO-E with BA, GWO, DE, BFP, FA and FPA. Here NA has been added at the place of best algorithm because we cannot compare the best algorithm with itself. From Table 5.16, it is clear that BA shows better statistical analysis for one function, BFP for none, DE for one, FA for four, FPA for two, GWO-E for eleven and GWO for none. So, even statistically, it can be said that GWO-E outperforms other well-known algorithms.

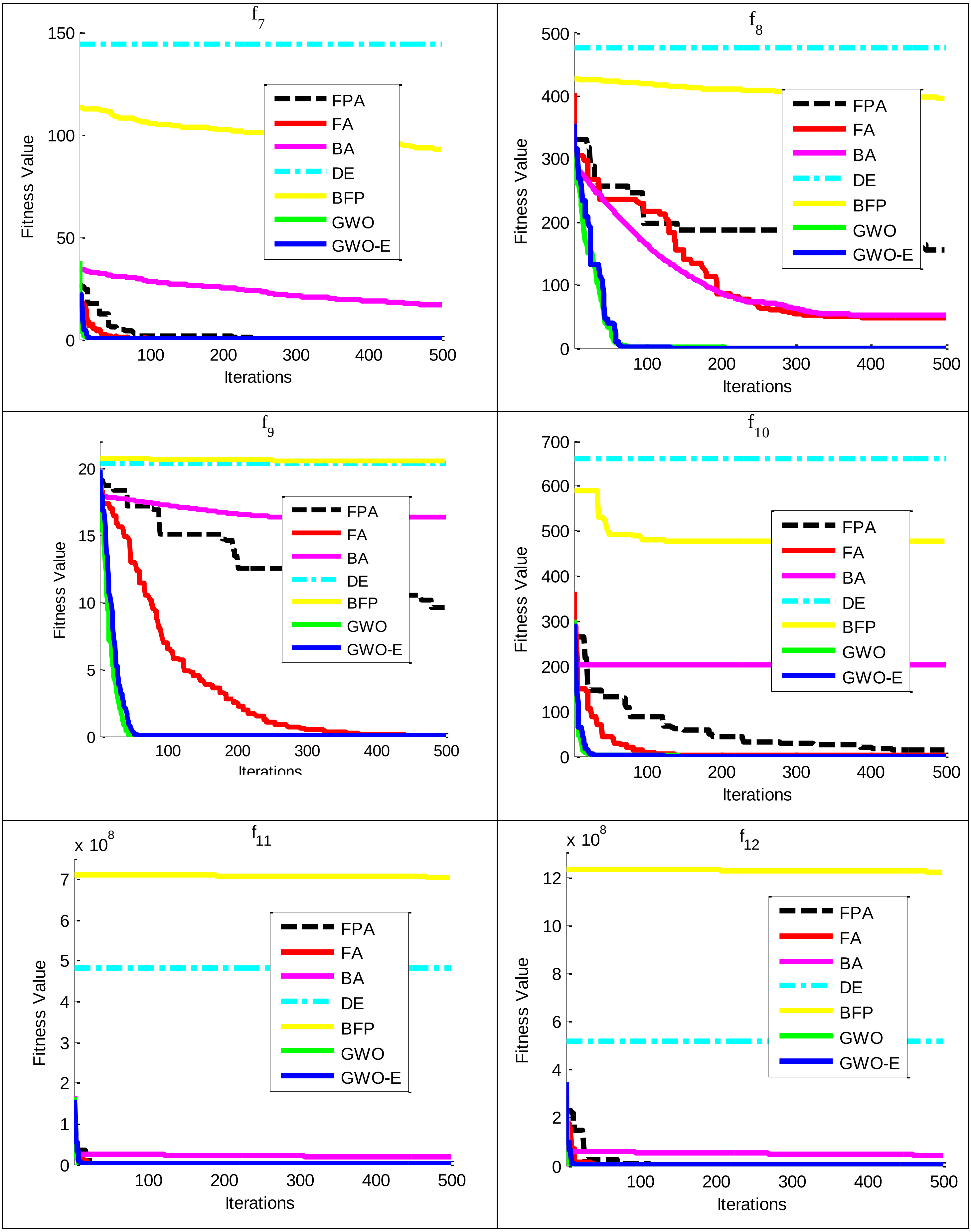
Function	BA	BFP	DE	FA	FPA	GWO	GWO-E
f_1	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	NA
f_2	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	NA
f_3	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	NA
f_4	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	NA
f_5	7.07E-18	7.07E-18	7.07E-18	3.98E-15	7.07E-18	0.6716	NA
f_6	7.07E-18	7.07E-18	7.07E-18	3.46E-14	7.07E-18	0.3225	NA
f_7	7.07E-18	7.07E-18	7.07E-18	7.07E-18	7.07E-18	1.35E-16	NA
f_8	3.31E-20	3.31E-20	3.31E-20	3.31E-20	3.31E-20	3.11E-20	NA
f_9	1.63E-18	1.63E-18	1.63E-18	1.63E-18	1.63E-18	1.45E-18	NA
f_{10}	3.31E-20	3.31E-20	3.31E-20	3.31E-20	3.31E-20	2.54E-04	NA
f_{11}	7.07E-18	7.07E-18	7.07E-18	NA	7.07E-18	2.29E-15	2.05E-15
f_{12}	7.07E-18	7.07E-18	7.07E-18	NA	7.07E-18	7.07E-18	7.07E-18
f_{13}	1.07E-10	7.07E-18	0.052	0.0059	NA	0.438	0.0014
f_{14}	7.62E-10	7.07E-18	2.31E-10	0.8659	5.46E-08	6.66E-05	NA
f_{15}	2.54E-01	7.07E-18	3.35E-17	NA	3.56E-11	9.46E-10	2.85E-16
f_{16}	NA	7.07E-18	6.95E-17	1.04E-06	7.07E-18	7.07E-18	7.07E-18
f_{17}	2.69E-04	7.07E-18	7.96E-17	NA	1.95E-13	4.46E-17	7.12E-17
f_{18}	2.43E-18	2.43E-18	NA	3.32E-17	4.50E-18	2.43E-18	2.43E-18
f_{19}	0.1320	7.0661e-18	3.7706e-12	0.0390	NA	0.7329	0.6918

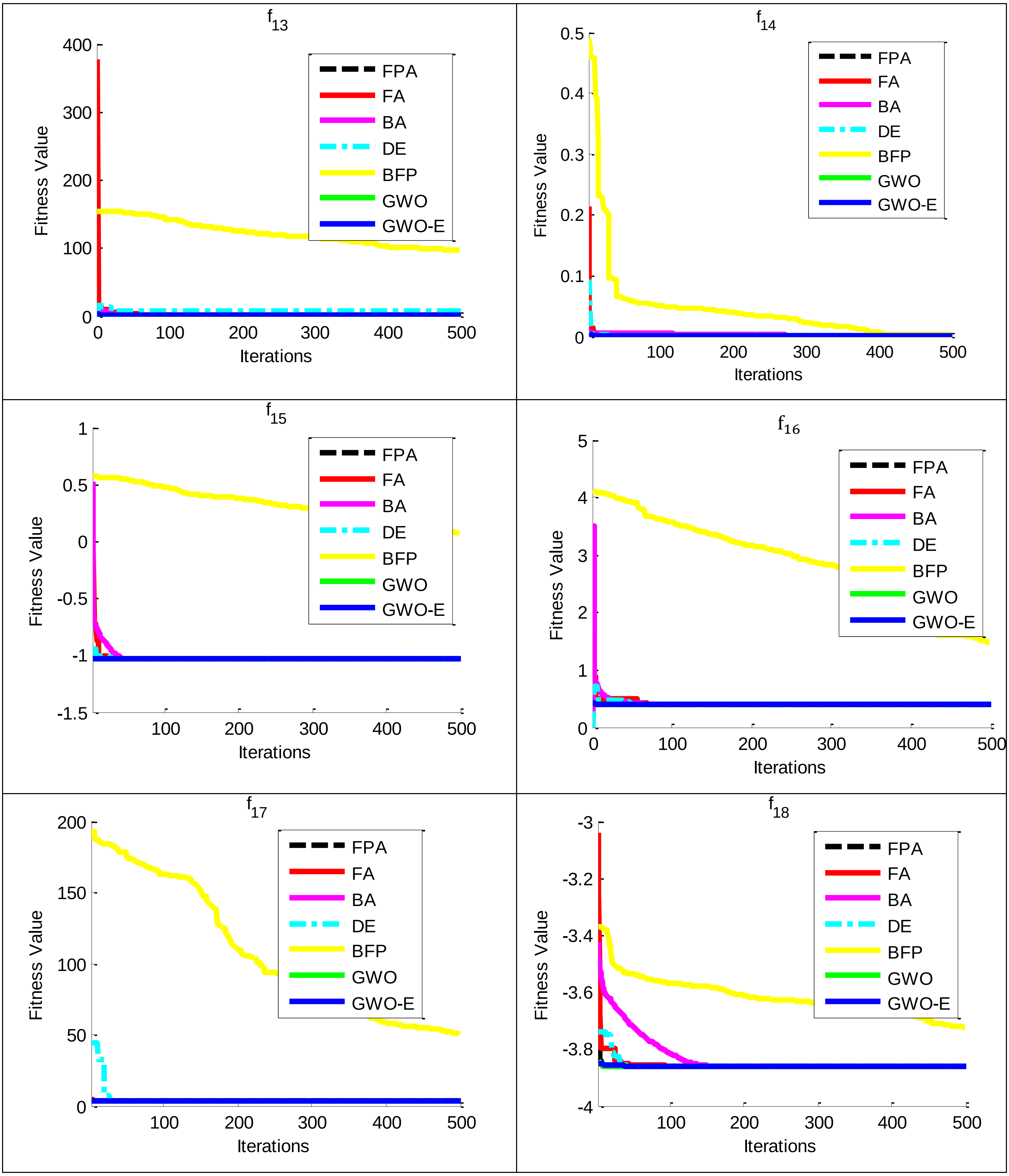
Table 5.15 Statistical testing of various algorithms.

D. Convergence Graphs

In this section, convergence characteristics of the proposed algorithm are investigated. Acc.to Berg *et al.* [157], abrupt changes in the initial set of iteration assist the exploration of the search space while the gradual changes in the end of iterations helps in the exploitation of the search space. These graphs are used to check the time utilized by this algorithm to reach at global solution. These convergence profiles are compared with other algorithms like GWO, FPA, FA, BFP, BA, DE and are shown in the Figure 5.2. These curves again show the superior performance of the GWO-E over the other algorithms.


f1
x 10^4
Fitness Value
Iterations
f2
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
GWO
GWO-E
f3
x 10^5
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
GWO
GWO-E
f4
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
GWO
GWO-E
f5
x 10^8
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
GWO
GWO-E
f6
x 10^4
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
GWO
GWO-E


f7
Fitness Value
Iterations
150
100
50
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
GWO
GWO-E
f8
Fitness Value
Iterations
500
400
300
200
100
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
GWO
GWO-E
f9
Fitness Value
Iterations
20
15
10
5
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
GWO
GWO-E
f10
Fitness Value
Iterations
700
600
500
400
300
200
100
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
GWO
GWO-E
f11
x 10^8
Fitness Value
Iterations
7
6
5
4
3
2
1
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
GWO
GWO-E
f12
x 10^8
Fitness Value
Iterations
12
10
8
6
4
2
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
GWO
GWO-E



f_{13}
f_{14}
f_{15}
f_{16}
f_{17}
f_{18}
FPA
FA
BA
DE
BFP
GWO
GWO-E
Fitness Value
Iterations

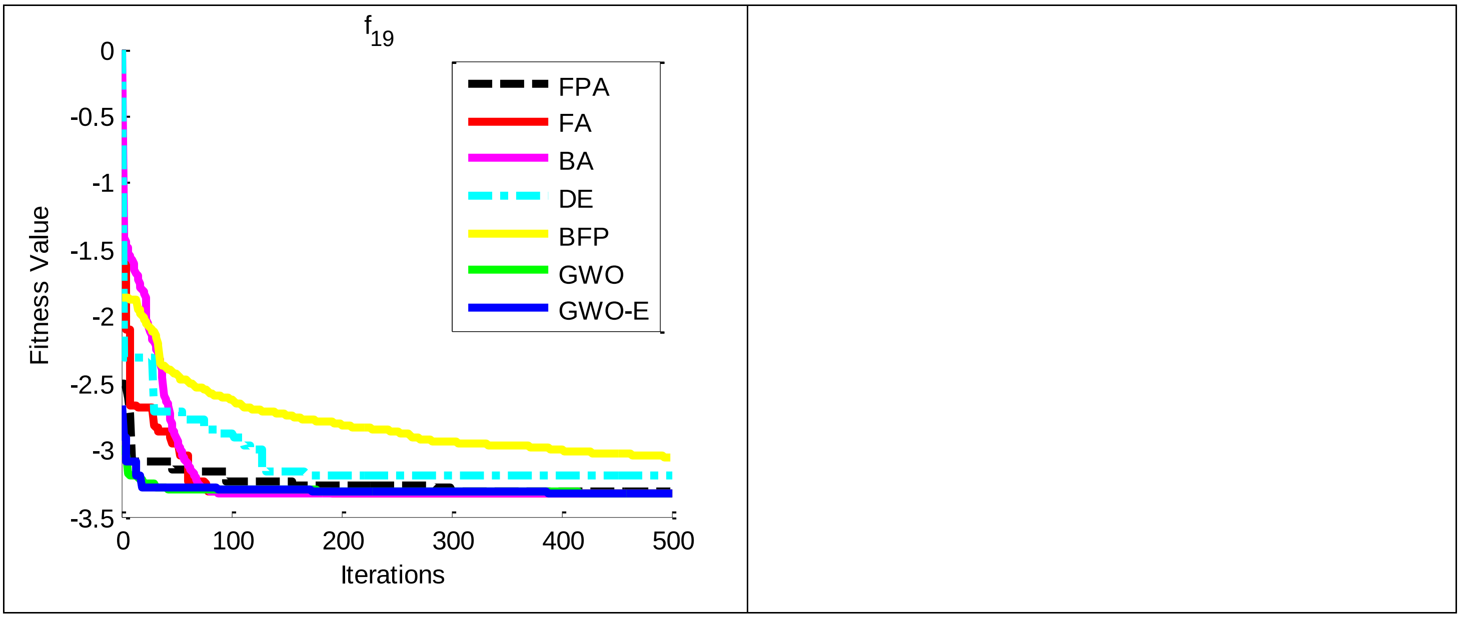


Figure 5.2 Convergence profile of GWO-E compared with various other algorithms

Inference drawn: The proposed algorithm GWO-E is better in exploring new search area and for exploiting the local search area over the other state-of-the-art algorithm. This can be judged from its performance for unimodal and multimodal functions. Apart from this, performance of GWO-E is also computed on different population size and this algorithm is able to adapt itself for different population size. Also the performance of GWO-E does not deteriorate with increase in dimension size. Hence, GWO is suitable for solving high- dimensional and complex problems. Further the statistical analysis and convergence curve again shows the superior performance of GWO-E. Hence this algorithm can be used for the synthesis of linear antenna array, in order to further evaluate its performance for real time application. The implementation of LAA is as explained below.

5.3 SYNTHESIS OF LINEAR ANTENNA ARRAY USING GWO-E

This section demonstrates the synthesis of unequally spaced liner antenna array with newly proposed algorithm GWO-E. Here, few case studies with different number of antenna elements have been considered to test the performance of GWO-E. This is done by comparing the optimized positions of the antenna element obtained by GWO-E with other well-known work existing in literature. For the synthesis of this type of linear antenna array, the excitation amplitude of the current is kept constant to the value of one. This uniform value of amplitude helps in reducing the complexity and cost of the network. This algorithm is executed for 20 times with population of 45. For the case 1 and 2, the maximum number of iterations are set to 500 and for rest of cases, it is set to 1000.

Case-1: 10 element with varying FNBW

In this example, the proposed algorithm GWO-E is used to determine the optimized position (x_m) of the elements of linear array without any condition imposed on FNBW. Optimization is done with a goal to reduce the side lobe in $\emptyset = [106°, 180°]$. The optimized positions of 10 elements are given in Table 5.16 and the results are compared with CSO [141], FPA [130], ACO [129], PSO [141], SMO [136], GWO [138], ALO [135] and e-ALO [147]. The radiation pattern for this case is illustrated in Figure 5.3. The peak SLL obtained by GWO-E is −26.04 dB, which is far better than the uniform method, PSO, CSO, ACO, FPA, ALO and SMO as shown in Table 5.17. The convergence characteristic of GWO-E for this case is shown in Figure 5.4.

Optimization Method	Optimized element Positions				
CSO[141]	0.1516	0.4115	0.7899	1.1048	1.6843
FPA[130]	0.1342	0.375	0.7522	0.9875	1.5661
ACO[129]	0.25	0.55	1.05	1.55	2.15
PSO [141]	0.2605	0.5105	1.0186	1.4694	2.1407
SMO[136]	0.236	0.528	1.007	1.471	2.126
GWO[138]	0.1250	0.3680	0.7522	0.9851	1.5546
ALO[135]	0.1259	0.3751	0.7515	0.9994	1.5652
e-ALO[147]	0.1270	0.3651	0.7422	0.9826	1.5568
GWO-E	0.1519	0.3218	0.7668	0.9171	1.5372

Table 5.16 Optimized positions of 10 element of LAA with varying FNBW

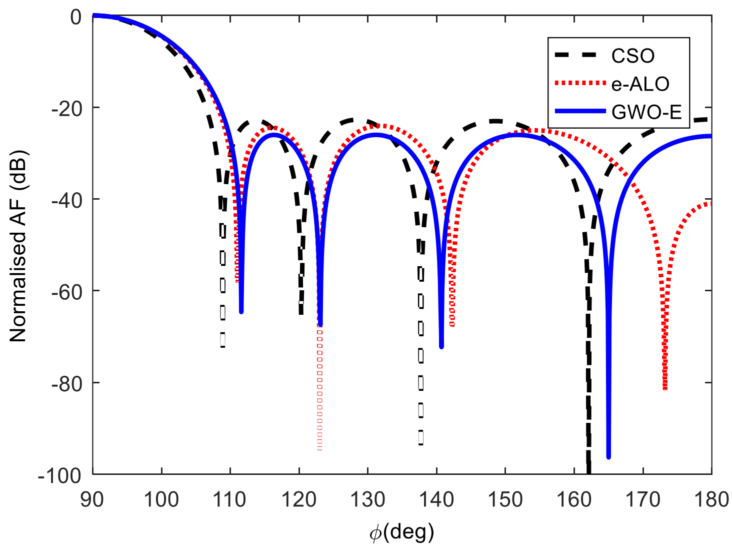


Figure 5.3 Optimized radiation pattern for 10 element (LAA) with varying FNBW

Algorithms	Conv	PSO[141]	CSO[141]	ACO[129]	FPA[130]	ALO[135]	SMO[136]	GWO-E[138]
SLL (dB)	−13.23	−20.72	−22.89	−22.66	−23.45	−23.29	−20.25	−26.04

Table 5.17 Comparison of peak SLL of 10 element (LAA) with varying FNBW

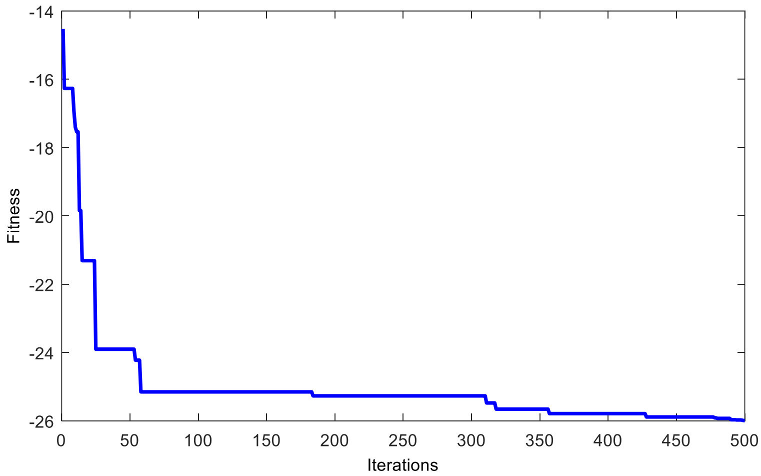


Figure 5.4 Convergence plot of 10 element with varying FNBW

Case-2: 10 elements with constant FNBW

In this case, 10 element array is synthesize with an aim to reduce the side lobe level and to maintain the desired FNBW constant at $\emptyset = 23°$. The tolerance for change in beam width is set to 1%. To achieve this desired goal the fitness function to be minimized is given as:

$$F_{obj} = k_1 f_{sl}(\emptyset) + \alpha.max\{0, |FNBW_c - FNBW_d| - 1\}$$

where $f_{sl}(\emptyset)$ denotes the peak calculated value of SLL, $FNBW_c$ and $FNBW_d$ are the calculated and desired first null beam width. This objective function is based on the penalty method i.e. if the obtained radiation pattern does not the meet the constraint of beam width then it provides the very high value of objective function. This is accomplished by assigning a high value to parameter α.

The array pattern obtained by GWO-E and by other compared algorithm is shown in Figure 5.5. From this Figure it is clear that there is reduction in SLL using GWO-E as compared to other algorithms. Table 5.18 shows the optimized position of the elements of LAA. The effectiveness of this algorithm for the synthesis of linear antenna can also be proved by comparing the peak value of SLL as given in Table 5.19. Figure 5.6 depicts the convergence characteristic of GWO-E for this case.

Optimization method	Optimized element positions				
CSO[141]	0.2081	0.6679	1.1347	1.7238	2.4036
PSO[141]	0.2040	0.6278	1.6641	1.4564	2.3033
ACO[129]	0.2500	0.5500	1.050	1.550	2.150
GWO-E	0.2475	0.7066	1.1868	1.6758	2.2742

Table 5.18 Optimized Positions of 10 element of LAA with constant FNBW

Algorithms	SLL (dB)
CSO[141]	−19.09
PSO[141]	−17.43
ACO[129]	−18.27
GWO-E	−20.47

Table 5.19 Comparison of peak SLL obtained by various algorithms with constant FNBW

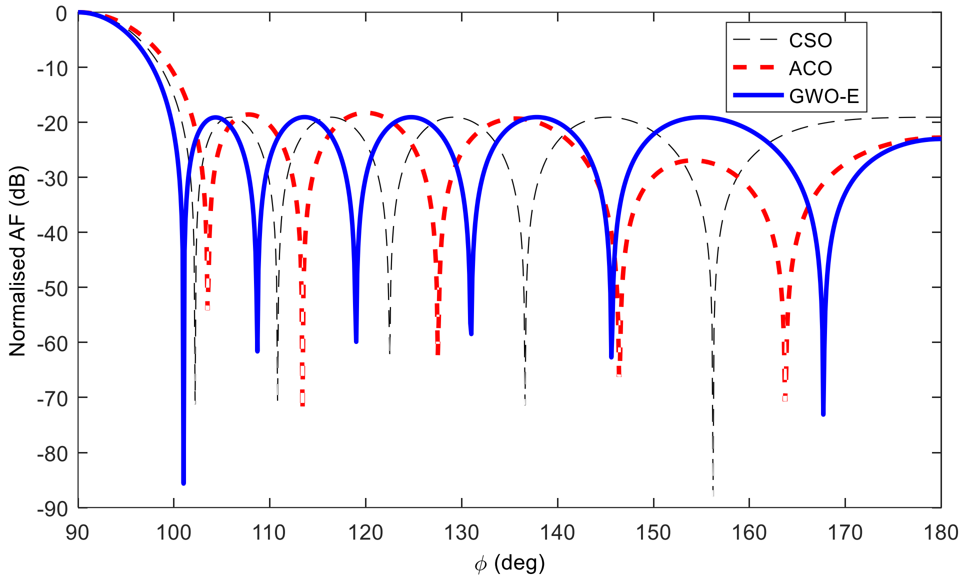


Figure 5.5 Radiation pattern obtained by GWO-E compared with CSO and ACO for 10 elements with constant FNBW

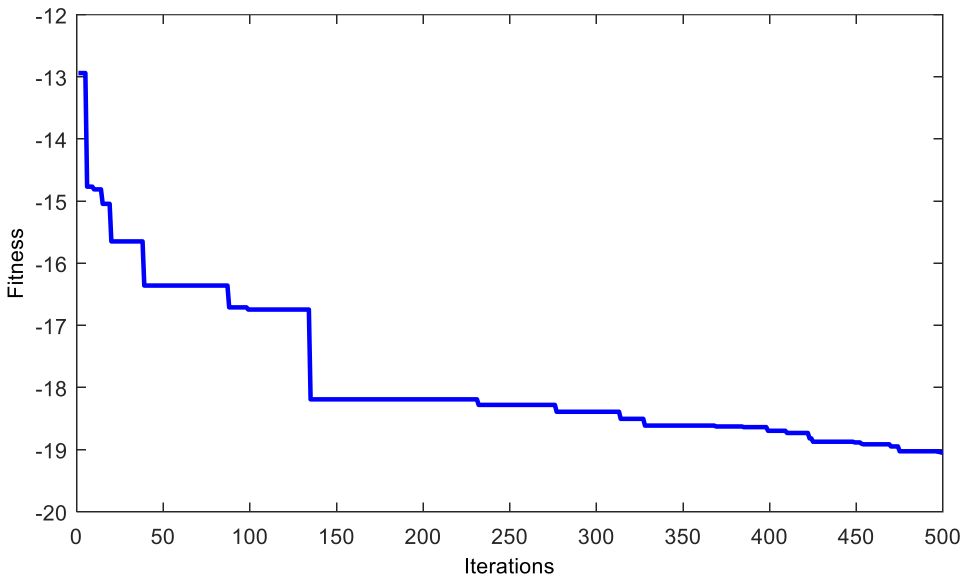


Figure 5.6 Convergence figure of 10 elements with varying FNBW

Case -3: 22 elements with null at 99°

In this case, 22 element is considered for the optimization of linear antenna array using GWO-E. The antenna is synthesize with an aim to minimize the side lobe level and also to place null at 99°. The optimized position of the elements obtained by GWO-E is shown in Table 5.20 and these results are compared with FPA [131], HSA [128] and BSA [140]. The SLL obtained by GWO-E is −24.81 dB with a null depth of −116.6 dB which is better than other algorithms as listed in Table 5.21. The radiation pattern of GWO-E is shown in the Figure 5.7 and is compared with HSA, BSA and FPA. This figure also illustrates the better value of null depth and SLL obtained by GWO-E.

Optimization method	Optimized positions					
FPA[131]	0.4269	0.7919	1.2906	1.9942	2.6005	3.0342
	3.9015	4.4903	5.5128	6.64482	7.7264	
HSA[128]	0.3995	0.7760	1.3627	1.7992	2.6738	3.0134
	4.0611	4.3301	5.5176	6.4005	7.6316	
BSA[140]	0.4155	0.8323	1.2136	2.0382	2.5284	3.0606
	3.8890	4.5311	5.4127	6.5161	7.6200	
GWO-E	0.3378	0.932	1.2426	2.1294	2.515	3.1986
	3.9328	4.656	5.5196	6.6812	8.203	

Table 5.20 Optimized element positions for 22-element with null at 99°

Algorithms	SLL (dB)	Null (dB) (99°)
TS[125]	−17.17	−67.94
BSA[140]	−23.54	−104.61
MS[125]	−18.11	−73.92
GA[125]	−15.73	−54.29
HSA[128]	−23.28	−103.3
MOWA/D-DE[143]	−20.93	−69.64
PSO[146]	−20.68	−49.94
FPA[131]	−23.81	−101.71
GWO-E	−24.81	−116.6

Table 5.21 Comparison of different algorithm with null at 99°

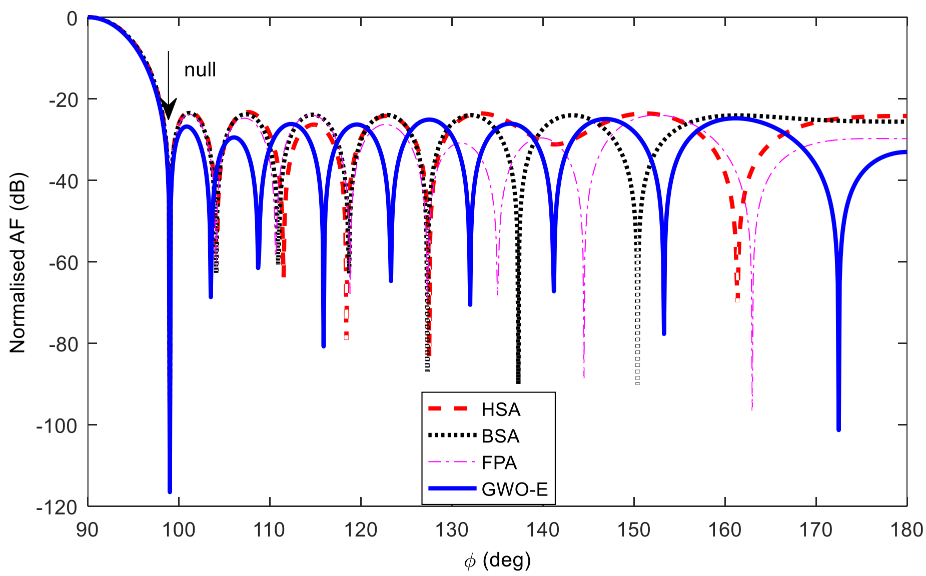


Figure 5.7 Optimized radiation pattern for 22 element with null placement at 99°

Case-4: 26 elements with null at $120°$, $168°$.

In this example, LAA with 26 element is synthesized with an aim to minimize the side lobe in band $\emptyset = [100°, 180°]$ and to place the nulls in the direction of $120°$ and $168°$. The optimized position for this case is compared with FPA [131], GA [125] and TS [125] and the same is listed in Table 5.22. The SLL and null depth obtained by GWO-E and by some other algorithms are shown in Table 5.23. Figure 5.8 depicts the radiation pattern for this case.

Optimization method	Optimized Positions						
FPA[131]	0.3832	1.3986	2.2314	3.3338	4.1948	5.2293	6.2419
	7.3744	8.5678	9.9085	11.6729	13.1859	14.8374	
GA[125]	0.4242	0.8472	1.5790	2.4680	2.9930	4.3910	4.6290
	5.6400	6.3990	7.7910	8.7950	9.9740	11.3800	
TS[125]	0.5311	1.4910	2.4680	3.4450	4.5240	5.5010	6.4910
	7.2680	8.4980	9.5140	10.2700	11.4400	12.4200	
GWO-E	0.4054	1.2048	2.1327	2.9801	3.7972	4.7508	5.6815
	6.9103	7.8111	9.1252	10.7977	12.4644	14.1113	

Table 5.22 Optimized positons of 26 elements with nulls at $120°$ and $168°$

Algorithms	SLL(dB)	Null (dB) (120°)	Null (dB) (168°)
TS[125]	−13.20	−49.65	−42.39
MS[125]	−18.11	−55.55	−77.17
GA[125]	−15.73	−55.76	−47.70
MOEA/D-DE [143]	−18.2	−89.98	−74.16
PSO[146]	−11.75	−56.45	−37.8
FPA[131]	−21.65	−126.9	−116.2
GWO-E	−23.001	−85.10	−87.07

Table 5.23 Comparison of SLL and nulls depth obtained by various algorithm for 26 elements LAA.

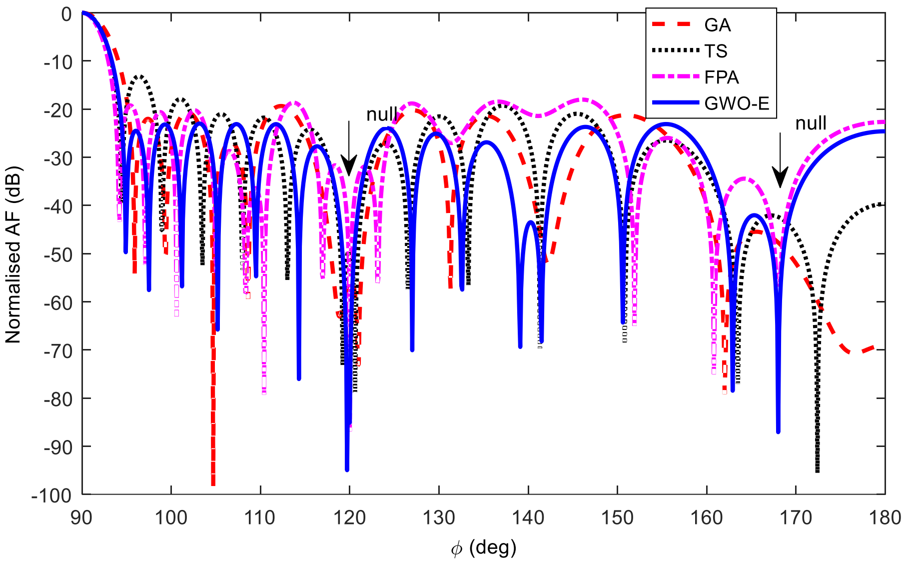


Figure 5.8 Radiation pattern of 26 elements with null placement and minimized SLL obtained by GWO-E and compared with GA, TS and FPA.

Case5: 28 element with varying FNBW

In this case, the synthesis of 28 element is done by using GWO-E with an objective to minimize the SLL in the region of $\emptyset = [106°, 180°]$. The optimized position obtained by GWO-E and by other algorithms are shown in Table 5.24. The peak value of SLL obtained by GWO-E is much lower than other algorithms as shown in Table 5.25. The optimized radiation pattern obtained by GWO-E is shown in Fig 5.9. The peak SLL value of GWO-E is 4.44 dB and 1.8 dB lower than PSO and CSO respectively.

Optimization method	Optimized element positions						
CSO[141]	0.4688	1.056	1.8448	2.593	3.3098	4.2854	4.8774
	5.8738	6.7506	7.856	8.8182	10.2334	11.8376	13.4844
PSO[141]	0.3406	1.286	1.9018	2.849	3.5698	4.0794	4.9022
	6.1044	7.2498	8.0952	9.2604	10.5968	11.9164	13.4236
e-ALO[147]	0.2578	1.0454	1.832	2.504	3.282	4.0188	4.7908
	5.7872	6.6636	7.768	8.730	10.146	11.75	13.3968
GWO-E	0.316	1.155	1.6818	2.6826	3.063	4.1764	4.7492
	5.784	6.5094	7.6732	8.6688	10.0198	11.6758	13.3492

Table 5.24 Optimized positions of 28 element (LAA) with varying FNBW

Algorithm	SLL(dB)
PSO[141]	−21.49
CSO[141]	−24.13
GWO-E	−25.93

Table 5.25 Comparison of peak SLL of 28 element (LAA) with varying FNBW

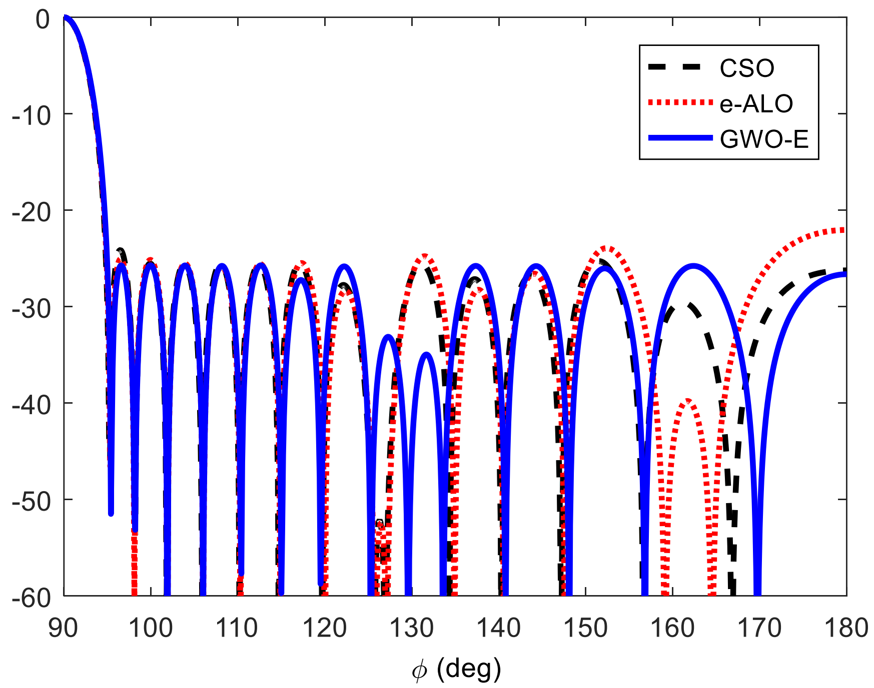


Figure 5.9 28 element with varying FNBW

Case-6: 28 element with constant FNBW

In this case, LAA of 28 elements is synthesized with the aim of minimizing the SLL and maintaining the FNBW constant at $\emptyset = 8°$. The tolerance for change in beam width is set to 1%. To achieve this desired goal the fitness function to be minimized is given as:

$$F_{obj} = k_1 f_{sl}(\emptyset) + \alpha.max\{0, |FNBW_c - FNBW_d| - 1\}$$

where $f_{sl}(\emptyset)$ denotes the peak calculated value of SLL, $FNBW_c$ and $FNBW_d$ are the calculated and desired first null beam width. This objective function is based on the penalty method i.e. if the obtained radiation pattern does not the meet the constraint of beam width then it provides the very high value of objective function. This is accomplished by assigning a high value to parameter α.

The array pattern obtained in this case is illustrated in Figure 5.10. The detail of optimized location of antennas and peak SLL is represented in Table 5.26 and 5.27 respectively.

Optimization Method	Optimized element positions						
CSO[141]	0.4874	1.288	2.046	3.019	3.6888	4.7948	5.767
	6.5314	7.700	8.9452	10.2136	11.6734	13.013	13.9998
PSO[141]	0.654	1.1542	0.4178	3.029	4.474	4.7878	5.7584
	6.89	8.6092	9.7856	10.584	11.814	12.855	13.9998
e-ALO[147]	0.548	1.092	1.6808	2.312	2.9572	3.6122	4.3218
	5.1508	6.0122	6.9776	7.9706	8.9888	10.264	11.6186
GWO-E	0.396	1.3514	2.2488	3.1756	4.129	5.0518	6.0242
	7.242	8.3876	9.3688	10.9776	12.6776	14.3776	16.0776

Table 5.26 Optimized positions of 28 element (LAA) with constant FNBW

Algorithms	PSO[126]	CSO[141]	GWO-E
Peak SLL (dB)	−17.09	−20.04	−23.024

Table 5.27 Comparison of 28 element (LAA) with constant FNBW

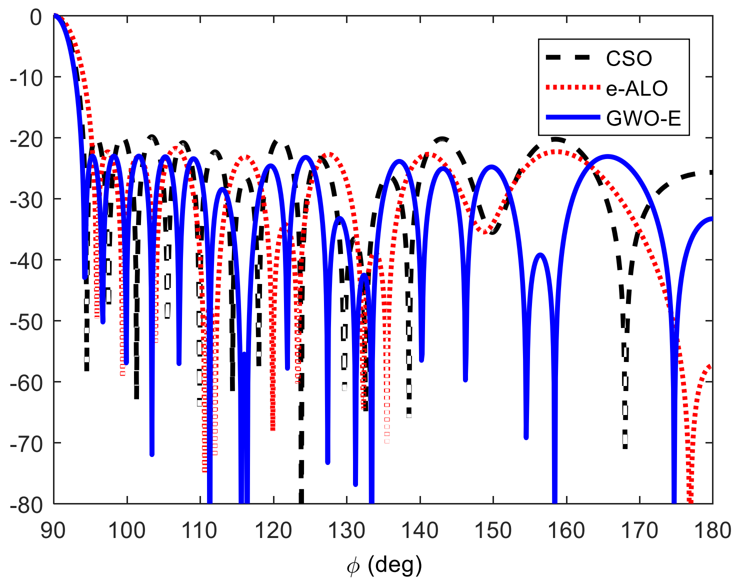


Figure 5.10 Comparison of optimized radiation pattern of 28 element with fixed FNBW

Case-7: 28 elements with multiple nulls

In this example, GWO-E has been applied for the synthesis of LAA with 28 elements. The goal of the implementation is to minimize the SLL in the regions of $\emptyset = [96°, 180°]$ with desired nulls at $\emptyset = 120°, 122.5°, 125°$. The optimized positions obtained by GWO-E and by other algorithms is shown in Table 5.28.

Algorithms	Optimized Positions						
CSO[141]	0.544	1.5094	2.2798	3.413	4.6574	5.735	6.7072
	7.5386	8.4444	9.7982	10.8122	11.4778	12.3128	13.4346
ALO[135]	0.4918	1.3584	2.4598	3.496	4.4896	5.3516	6.3352
	7.3986	8.4334	9.203	10.3998	11.4002	12.4998	13.9924
BSA[140]	0.4218	1.4685	2.2241	3.2013	4.0176	5.1537	6.2439
	7.114	8.3492	9.5940	10.9847	12.7586	14.2894	15.8698
FPA[130]	0.67	1.083	1.758	2.5344	3.2682	4.1296	4.7978
	5.696	6.5954	7.4164	8.9024	9.9174	10.9578	12.2666
e-ALO[147]	0.422	1.468	2.0222	3.2034	4.0376	5.1558	6.2442
	7.1274	8.3472	9.5936	10.9828	12.756	14.514	15.866
ACO [129]	0.70	1.70	3.1	3.3	4.64	5.7	5.9
	7.70	8.5	9.50	10.7	11.5	12.5	13.9
IWO[149]	0.3828	1.6264	2.1508	3.3462	4.0158	5.2808	5.891
	7.0022	7.6244	8.8592	9.8032	10.8038	11.4988	12.7556
COA[137]	0.4719	1.3622	2.3091	3.1064	4.0688	5.2663	6.2660
	7.0839	8.2914	9.6443	11.1218	12.7414	14.2912	14.244
MIWO[150]	0.5938	1.6318	2.4846	3.243	4.146	5.1318	5.8824
	6.8244	7.8086	8.6328	10.0398	11.1082	11.928	13.3944
GWO-E	0.5246	1.022	2.0102	2.673	3.6092	4.3596	5.2108
	6.1458	7.2372	8.414	9.31138	10.989	12.5106	14.0814

Table 5.28 Optimized Position of N=28 elements of (LAA) with multiple nulls

The radiation pattern obtained by GWO-E is illustrated in Figure 5.11 and this pattern is compared with QPM [126], HSA [128], e-ALO [147]. Table 5.29 represents the value of null depths and peak SLL obtained by GWO-E and by other optimization algorithms.

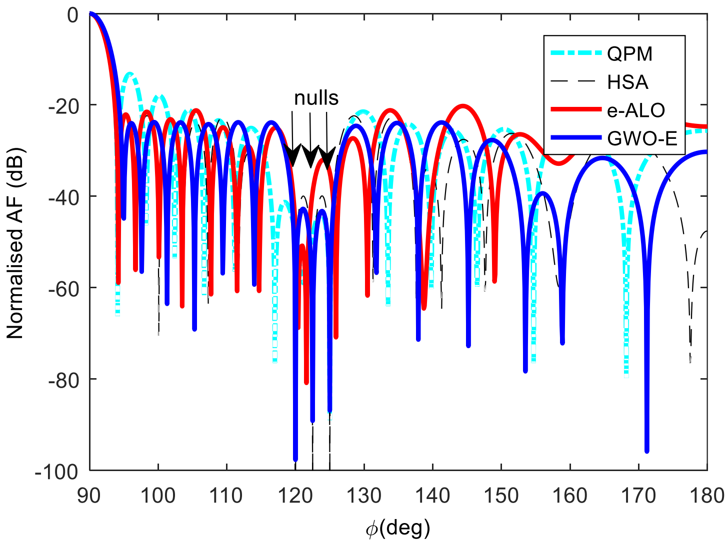


Figure 5.11 Radiation pattern of 28 element multiple nulls

Optimization Method	Null Depth in dB			Peak SLL (in dB)
Null Locations	120°	122.5°	125°	
COA[137]	−60.10	−60.05	−60.08	−16.79
ALO[135]	−76.89	−84.58	−81.96	−14.12
FPA[130]	−90.81	−95.12	−89.42	−20.46
BSA[140]	−80.49	−93.59	−82.49	−21.90
CLPSO[148]	−60.00	−60.01	−60.04	−21.63
MIWO[150]	−70.60	−76.19	−79.48	−14.76
IWO[149]	−76.19	−75.04	−83.18	−13.57
GWO[138]	−78.41	−84.42	−83.22	−12.87
MVMO[151]	−71.79	−78.79	−70.19	−14.76
e-ALO[147]	−80.84	−63.27	−70.86	−22.32
GWO-E	−97.6708	−89.14515	−86.8660	−23.756

Table 5.29 Null Depth after position optimization by different techniques for N = 28 elements

Case-8: 32 element with single null

This example demonstrates the synthesis of 32 elements array with desired nulls at $\emptyset = 81°$ and$\emptyset = 99$. The peak SLL and null depth obtained by different optimization algorithm is shown in Table 5.30. It is clear from the Table 5.30, that the GWO-E provides the first side lobe at -23.62 dB and shows the null at -143.7908 dB which is far better from all the other algorithms. The goal of GWO-E is to achieve minimum SLL with desired nulls. The optimized position obtained by GWO-E is shown in Table 5.31 and the results are compared with various algorithms. Further, on examining the results it is observed that the peak value obtained by GWO-E is comparable with e-ALO and COA. The peak value of IWO is slightly better than the GWO-E but its null depth response is very poor. The optimized radiation pattern for this case is given in Figure 5.12.

Optimization method	Null depth (dB)	PSLL (in dB)
CSO[141]	−80	−18.24
PSO[126]	−60	−18.71
ACO[129]	−50	−16.4
BSA[140]	−107.50	−19.51
ALO[135]	−111.8	−15.2
HSA[128]	−88.08	−17.73
GWO[138]	−106	−20.50
FPA[130]	−85.27	18.32
MIWO[150]	−60.11	−24.00
COA[137]	−79.85	−23.95
IWO[149]	−43.99	−24.00
e-ALO[147]	−82.07	−23.81
QPM[126]	−59.58	−18.41
GWO-E	−143.7908	−23.6254

Table 5.30 Comparative analysis of null depth and peak SLL for N=32 element

Algorithms	Optimized element positions						
CSO[141]	0.5766	1.366	2.3858	3.0398	3.9536	4.6494	5.3772	6.364
	6.9696	7.9076	8.7644	9.8504	10.9634	12.4182	14.0824	15.5
COA[137]	0.4328	1.2397	2.1242	2.9799	3.8806	4.7725	5.6434	6.6828
	7.6270	8.7630	9.8809	11.120	12.526	14.2445	16.176	17.605
PSO [126]	0.5300	1.3700	2.3500	3.1100	3.9700	4.6600	5.3300	6.1100
	6.8600	7.8000	8.7600	9.9000	11.100	12.4800	14.100	15.510
FPA[130]	0.5	1.4992	2.4996	3.4934	4.452	5.2954	6.0168	6.811
	7.5266	8.5124	9.5	10.5008	11.502	12.8722	14.498	15.995
ALO[135]	0.4242	1.4824	2.4998	3.5	4.471	5.3876	6.056	6.9026
	7.5254	8.5368	9.5	10.5064	11.5	12.9018	14.5	15.998
ACO[129]	0.5	1.5	2.64	3.3	3.9	5.1	6.64	6.7
	7.5	8.7	10.64	11.5	12.7	13.7	15.1	16.64
BSA[140]	0.56084	1.38547	2.1014	2.9924	3.6179	4.4692	5.3779	6.1846
	6.98827	7.68024	8.6182	9.5657	11.0268	12.5317	14.417	15.500
IWO[149]	0.579	1.329	2.4756	3.0568	4.1656	2.3497	5.68	6.3632
	7.1966	4.0294	8.0588	9.9332	10.9988	12.4012	14.0498	15.5
MIWO[150]	0.5138	1.2638	2.2116	2.985	3.718	4.6778	5.1778	6.372
	6.7732	7.8318	8.58	9.6444	10.7856	12.3158	14.0134	15.5
QPM[216]	0.5600	1.4500	2.3500	3.1900	3.9600	4.6700	5.3700	6.1100
	6.9200	7.8500	8.8900	10.060	11.330	12.650	14.010	15.350
HSA[128]	0.5116	1.24909	2.1722	3.06560	3.8114	4.8211	5.1627	6.3120
	6.69505	7.87192	8.6634	9.8225	11.1077	12.377	14.162	15.722
GWO[138]	0.3886	1.4814	2.4982	3.495	4.4824	5.4284	5.9996	6.903
	7.5078	8.4918	9.5	10.5112	11.5034	12.9118	14.5	16
GWO-E	0.6356	1.0208	2.2136	2.9242	3.9938	4.782	5.684	6.5702
	7.7198	8.8382	9.8896	11.0316	12.6634	14.326	16.0232	17.7146

Table 5.31 Optimized position of N= 32 elements of (LAA) with single nulls

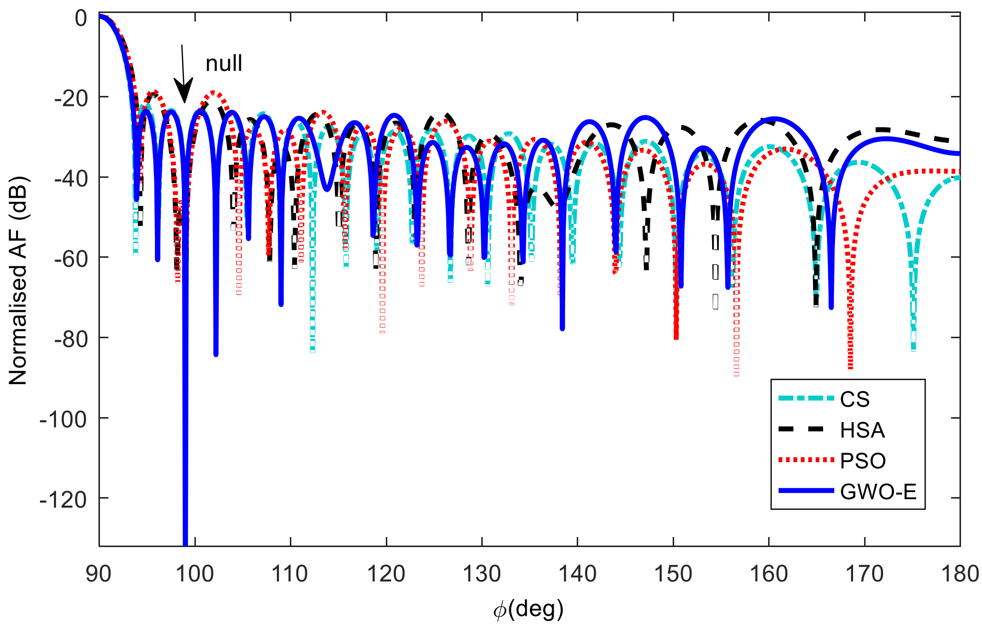


Figure 5.12 Optimized radiation pattern for 32 element

CHAPTER-6

CONCLUSION and FUTURE SCOPE

6.1 CONCLUSION

Nature-inspired algorithms are the powerful and valuable tool for solving real world optimization problem. In this thesis, two new approaches have been discussed, namely EGWO and GWO-E. The EGWO, combines the advantages of opposite based learning (OBL) technique and oscillatory function. The OBL helps in enhancing the explorative ability of the GWO and oscillatory function controls the extent of both exploration and exploitation. This algorithm has been tested on nine benchmark functions and the results are compared with well know algorithms like BA, BFP, FA, FPA, GWO. After the experimental analysis with respect to standard state-of-the-art algorithms, this algorithm is found to provide very competitive results.

Another algorithm proposed in this thesis is GWO-E. Two modifications to GWO have been proposed which make it competitive with other popular techniques. Firstly, this algorithm uses new set of parameters to evaluate the position of alpha, beta and delta during the exploration phase. This modification helps in avoiding the premature convergence. Secondly, the algorithm employs the concept of opposition based learning method in first half of the iterations in order to produce the diversity among the search agents. Both these modifications help in enhancing the exploration of the GWO algorithm. The performance of the GWO-E has been evaluated by finding solutions to different benchmark functions. The results are compared with the other popular techniques such as BA, DE, FA, FPA and GWO by measuring the best, worst, mean and standard deviation. The experimental results of unimodal and multimodal functions show that the exploitation and exploration of the algorithm has been enhanced. Performance of the proposed method has also been tested with different population size and variable dimensions. With increase in population, the performance of GWO-E improves slightly and is better than other state-of-the-art algorithms. GWO-E is found to be suitable for solving high dimension and complex problems. Statistical tests also indicate the superior performance of GWO-E over other popular algorithms. Convergence characteristics have also been plotted which show faster speed of convergence of GWO-E.

These both approaches are better than the original GWO and hence can be used for solving the real world applications. From the chapter 5 it is clear that GWO-E has fast response and better convergence. Hence, such encouraging results motivated us to apply this proposed approach for the synthesis of linear antenna array. GWO-E is applied to optimize the inter-element position in order to achieve the desired radiation characteristic with minimum side lobe level.

Minimizing the side lobe level while maintaining the beam width and null positioning is also demonstrated in this work. The performance of GWO-E have been compared with the results obtained by other nature-inspired algorithms while synthesizing linear antenna array. The results of GWO-E provides enhancement in results from other work exists in literature.

6.2 FUTURE SCOPE

As in this thesis, EGWO has been applied to standard numerical benchmark functions only so for the future work this work can be extended to various real world application like antenna and electromagnetic field, for optimization of waveguide devices, business optimization. The proposed algorithm GWO-E has been applied on benchmark function and on linear antenna array synthesis. This algorithm can be extended for other field like economic load dispatch problem, digital signal processing, wireless sensor networks, neural networks, concentric circular array.

Binary version of proposed approaches can also be designed and extended to the field of bioinformatics, feature classification, knapsack problem, thinning of linear and planar antenna array and other applications. In future, parallel, hybrid and modified version of the proposed approaches can be designed.

REFRENCES

[1] Rosenthal A (1951). The history of calculus, *The American Mathematical Monthly*, 58(2), 75-86.

[2] Bertsekas DP. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[3] Goldstein AA (1962). Cauchy's method of minimization, *Numerische Mathematik*, 4(1), 146-150.

[4] Fraser CG (1994). The origins of Euler's variational calculus, *Archive for history of exact sciences*, 47(2), 103-141.

[5] Kirkpatrick S, Gelatt CD and Vecchi MP (1983). Optimization by simulated annealing, *science*, 220(4598), 671-680.

[6] Holland JH (1992). Genetic algorithms, *Scientific American*, 267(1), 66-72.

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

[8] Dorigo M, Birattari M and Stutzle T (2006). Ant colony optimization, *IEEE computational intelligence magazine*, 1(4), 28-39.

[9] Basturk B and Karaboga D (2006). An artificial bee colony algorithm for numeric function optimization. *In IEEE swarm intelligence symposium,* 8(1), 687-697.

[10] Kennedy J and Eberhart R (1995). Particle swarm optimization, IEEE international conference on Neural Networks [Perth, Australia: 1995], pp. 1942–1948.

[11] Yang XS (2012). Flower pollination algorithm for global optimization, In *International Conference on Unconventional Computing and Natural Computation* [11^{th}: Orlean, France: 2012], pp. 240-249. Springer Berlin Heidelberg.

[12] Mirjalili S, Mirjalili SM and Lewis A (2014), Grey wolf optimizer, *Advances in Engineering Software*, 69, 46-61.

[13] Yang XS and Deb S (2009). Cuckoo search via Lévy flights, *IEEE World Congress on Nature & Biologically Inspired Computing* [Coimbatore, India: 2009], pp. 210-214.

[14] Yang XS (2010). A new metaheuristic bat-inspired algorithm, *Nature inspired cooperative strategies for optimization*, 284, 65-74.

[15] Wolpert DH and Macready WG (1997). No free lunch theorems for optimization, *IEEE transactions on evolutionary computation*, 1(1), 67-82.

[16] Blum C and Roli A (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison, *ACM Computing Surveys*, 35(3), 268-308.

[17] Alba E and Dorronsoro B (2005). The exploration/exploitation tradeoff in dynamic cellular genetic algorithms, *IEEE Transactions on Evolutionary Computation*, 9(2), 126-142.

[18] Olorunda O and Engelbrecht AP (2008). Measuring exploration/exploitation in particle swarms using swarm diversity, *In IEEE World Congress on Computational Intelligence on Evolutionary Computation [Hong Kong, China: 2008],* pp. 1128-1134.

[19] Črepinšek M, Liu SH and Mernik M (2013). Exploration and exploitation in evolutionary algorithms: A survey, *ACM Computing Surveys*, *45*(3), 35.

[20] Lin L and Gen M (2009). Auto-tuning strategy for evolutionary algorithms: balancing between exploration and exploitation, *Soft Computing-A Fusion of Foundations, Methodologies and Applications*, 13(2), 157-168.

[21] Talbi EG . *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[22] Gutjahr WJ (2009). Convergence analysis of metaheuristics, *In Metaheuristics*, 10, 159-187.

[23] Koza, John R. *Genetic programming: on the programming of computers by means of natural selection*. MIT press, 1992.

[24] Yao X, Liu Y and Lin G (1999). Evolutionary programming made faster, *IEEE Transactions on Evolutionary computation*, 3(2), 82-102.

[25] Hansen N, Müller SD and Koumoutsakos P (2003). Reducing the time complexity of the de-randomized evolution strategy with covariance matrix adaptation, *Evolutionary computation*, 11(1), 1-18.

[26] Rechenberg I (1973). Evolution Strategy: Optimization of Technical systems by means of biological evolution, *Fromman-Holzboog, Stuttgart*, *104*.

[27] Mirjalili S (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, *Knowledge-Based Systems*, 89, 228-249.

[28] Simon D (2008). Biogeography-based optimization, *IEEE transactions on evolutionary computation*, 12(6), 702-713.

[29] Webster B, Bernhard PJ. A local search optimization algorithm based on natural principles of gravitation. *Proceedings of the 2003 international conference on information and knowledge engineering* [Las Vegas, Nevada, USA: 2003], pp. 255–61.

[30] Erol OK and Eksin I (2006). A new optimization method: big bang–big crunch, Advance Engineering Software, 37, 106–11.

[31] Hatamlou A (2013). Black hole: A new heuristic optimization approach for data clustering, *Information sciences*, 222, 175-184.

[32] Zheng YJ (2015). Water wave optimization: a new nature-inspired metaheuristic, *Computers & Operations Research*, 55, 1-11.

[33] Beni G and Wang J. Swarm intelligence in cellular robotic systems, *Robots and Biological Systems: Towards a New Bionics*, 703-712.

[34] Bonabeau E, Dorigo M and Theraulaz G (1999). *Swarm intelligence: from natural to artificial systems*. Oxford university press, 1999.

[35] Salgotra R and Singh U. A novel bat flower pollination algorithm for synthesis of linear antenna arrays, *Neural Computing and Applications*, 1-14, 2016.

[36] Balanis CA. *Antenna Theory: Analysis and Design*. A John Wiley and Sons, 2005.

[37] Balanis CA and Birtcher CR (2008). *Antenna measurements in Modern Antenna Handbook. John Wiley and Sons, 2008*.

[38] Hansen RC. *Phased Arrays in Antenna Engineering Handbook.* J. L. Volakis, Ed., McGraw-Hill, 4th edition, 2007.

[39] Kraus JD, Marhefka RJ and Khan AS, *Antennas for all Applications*, Tata McGraw-Hill, 3rd edition, 2007.

[40] Recioui A *et al.* (2008). Synthesis of linear arrays with side lobe level reduction constraint using genetic algorithms, *International journal of microwave and optical technology*, 3(5), 524-530.

[41] Zaman MA (2011). Phased Array Synthesis Using Modified Particle Swarm Optimization, *Journal of Engineering Science & Technology Review*, 4 (1).

[42] Goldberg D. Genetic algorithms in optimization, search and machine learning. *Reading: Addison-Wesley* 1989.

[43] Omara FA and Arafa MM (2010). Genetic algorithms for task scheduling problem, *Journal of Parallel and Distributed Computing*, 70(1), 13-22.

[44] Montana DJ and Davis L (1989). Training Feedforward Neural Networks Using Genetic Algorithms, *In International Joint Conference on Artificial Intelligence* [Cambridge: 1989], vol. 89, pp. 762-767.

[45] Linden DS (2002). Antenna design using genetic algorithms, In *Proceedings of the Annual Conference on Genetic and Evolutionary Computation* [4th: New York: 2002], pp. 1133-1140.

[46] Chang EI and Lippmann RP (1991). Using genetic algorithms to improve pattern classification performance. *In Advances in neural information processing systems*, 797-803.

[47] Maulik U and Bandyopadhyay S (2000). Genetic algorithm-based clustering technique, *Pattern recognition*, 33(9), 1455-1465.

[48] Chipperfield, AJ and Fleming PJ (1994). Parallel genetic algorithms: A survey, *Automatic Control and System Engineering.*

[49] Sharawi M *et al.* (2014). Flower pollination optimization algorithm for wireless sensor network lifetime global optimization, *International Journal of Soft Computing and Engineering*, 4(3), 54-59.

[50] Emary E *et al.* (2014). Retinal vessel segmentation based on flower pollination search algorithm, *In Proceedings of the Fifth International Conference on Innovations in Bio-Inspired Computing and Applications* [5th :2014], pp. 93-100.

[51] Bekdaş G, Nigdeli SM and Yang XS (2015). Sizing optimization of truss structures using flower pollination algorithm, *Applied Soft Computing*, 37, 322-331.

[52] Kaur G, Singh, D and Kaur M (2013). Robust and efficient 'RGB' based fractal image compression: Flower pollination based optimization, *International Journal of Computer Applications*, 78(10).

[53] Abdelaziz AY, Ali ES and Elazim SA (2016). Flower pollination algorithm to solve combined economic and emission dispatch problems, *Engineering Science and Technology, an International Journal*, 19(2), 980-990.

[54] Wang R *et al.* (2015). Local greedy flower pollination algorithm for solving planar graph coloring problem, *Journal of Computational and Theoretical Nanoscience*, 12(11), 4087-4096.

[55] Vedula VSSS, Paladuga SR and Prithvi MR (2015). Synthesis of circular array antenna for side lobe level and aperture size control using flower pollination algorithm, *International Journal of Antennas and Propagation*.

[56] Rodrigues D *et al.*(2015). Binary flower pollination algorithm and its application to feature selection, *In Recent Advances in Swarm Intelligence and Evolutionary Computation*, 85-100.

[57] Pashaei E, Ozen M and Aydin N (2015). An application of black hole algorithm and decision tree for medical problem, I*n IEEE International Conference on Bioinformatics and Bioengineering (BIBE)* [15th:Belgrade, Serbia: 2015], pp. 1-6.

[58] Yaghoobi S, Hemayat S and Mojallali H(2015). Image gray-level enhancement using Black Hole algorithm, *IEEE International Conference on Pattern Recognition and Image Analysis* [2nd:Rasht, Iran: 2015], pp. 1-5.

[59] Kuo Lw *et al. (2016).* Clustering based on black hole phenomenon, International Journal of Advanced Computational Engineering and Networking, 4(10), 49-51.

[60] Hasan Z and El-Hawary (2014). Optimal power flow by black hole optimization algorithm, *In IEEE conference on Electrical Power and Energy* [Calgary, Canada: 2014], pp. 134-141.

[61] Gómez A *et al.* (2016). An binary black hole algorithm to solve the set covering problem, *In IEEE Iberian Conference on Information Systems and Technologies* [11th: Las Palmas, Spain: 2016], pp. 1-5.

[62] Ebadifard F and Babamir SM (2017). Optimizing multi objective based workflow scheduling in cloud computing using black hole algorithm, *In IEEE International Conference on Web Research* [3th: Tehran, Iran: 2017], pp. 102-108.

[63] Aslani H, Yaghoobi M and Akbarzadeh-T MR. (2015). Chaotic inertia weight in black hole algorithm for function optimization. *IEEE International Congress on Technology, Communication and Knowledge* [Mashhad, Iran: 2015], pp. 123-129.

[64] Rashedi E, Nezamabadi-Pour H and Saryazdi S (2009). GSA: a gravitational search algorithm, *Information sciences*, 179(13), 2232-2248.

[65] Emara HM and Fattah HA. Continuous swarm optimization technique with stability analysis, *Proceedings of American Control Conference* [Boston, USA: 2004], pp. 2811-2817.

[66] Shi Y and Eberhart R. A modified particle swarm optimizer, *IEEE International Conference on Evolutionary Computation Proceedings* [Anchorage, USA: 1998] pp. 69-73.

[67] Eberhart RC and Shi Y. Tracking and optimizing dynamic systems with particle swarms. *Proceedings of Congress on Evolutionary Computation* [Seoul, South Korea: 2001], pp. 94-100.

[68] Shi Y (2001). Particle swarm optimization: developments, applications and resources. *Proceedings of Congress on evolutionary computation* [Seoul, South Korea: 2001], pp. 81-86.

[69] Bergh F. An Analysis of Particle Swarm Optimizers. Ph.D. Thesis, University of Pretoria, Pretoria, South Africa, 2002.

[70] Ratnaweera A, Halgamuge SK and Watson HC (2004). Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients, *IEEE Transactions on evolutionary computation*, 8(3), 240-255.

[71] Nguyen BH, Xue B and Andreae P (2017). A Novel Binary Particle Swarm Optimization Algorithm and Its Applications on Knapsack and Feature Selection Problems, *In Intelligent and Evolutionary Systems,* 8, 319-332.

[72] Grimaldi EA *et al.* (2004). PSO as an effective learning algorithm for neural network applications, *International Conference on Computational Electromagnetics and Its Applications* [3rd, Bejing, China : 2004], pp. 557-560.

[73] Ryala MH *et al*. Evaluation of Particle Swarm Optimization for Medical Image Segmentation, *In International Conference on Systems Science*, pp. 61-72.

[74] Maitra M and 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.

[75] Hu M, Hu L and Wang L (2016). Astronomical image superposition using an new improved particle swarm optimization, *IEEE International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery* [12th: Changsha, China: 2016], pp. 173-178.

[76] Verma D *et al.* (2016). Maximum power point tracking (MPPT) techniques: Recapitulation in solar photovoltaic systems, *Renewable and Sustainable Energy Reviews*, 54, 1018-1034.

[77] Merkle D, Middendorf M and Schmeck H (2002). Ant colony optimization for resource-constrained project scheduling, *IEEE transactions on evolutionary computation*, 6(4), 333-346.

[78] Dowsland KA and Thompson JM (2008). An improved ant colony optimization heuristic for graph colouring, *Discrete Applied Mathematics*, 156(3), 313-324.

[79] Lu X and Zhou. Image segmentation based on markov random field with ant colony system, *IEEE International Conference on Robotics and Biomimetics* [Sanya, China: 2007], pp. 1793-1797.

[80] Fathima KSA and Sindhanaiselvan K (2013). Ant colony optimization based routing in wireless sensor networks, *International Journal of Advanced Networking and Applications*, 4(4), 1686.

[81] Saremi S, Mirjalili SZ and Mirjalili SM (2015). Evolutionary population dynamics and grey wolf optimizer, *Neural Computing and Applications*, 26(5), 1257-1263.

[82] Lewis A, Mostaghim, S and Randall M (2008). Evolutionary population dynamics and multi-objective optimization problems, *Multi-Objective Optimization in Computational Intelligence: Theory and Practice*, 185-206.

[83] Suganthan PN *et al.* (2005). Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, *Kanpur genetic algorithm laboratory report*, *2005005*, 2005.

[84] Rodríguez L, Castillo O and Soria J (2016, July). Grey wolf optimizer with dynamic adaptation of parameters using fuzzy logic, *IEEE Congress on Evolutionary Computation* [Vancouver, Canada: 2016], pp. 3116-3123.

[85] Emary E, Zawbaa H M and Hassanie A E (2016). Binary grey wolf optimization approaches for feature selection, *Neurocomputing*, 172, 371-381.

[86] Kennedy J and Eberhart RC (1997). A discrete binary version of the particle swarm algorithm, *IEEE International Conference on Computational Cybernetics and Simulation in Systems, Man, and Cybernetics* [Orlando, USA: 1997], pp. 4104-4108.

[87] Eiben AE, Raue PE and Ruttkay Z (1994, October). Genetic algorithms with multi-parent recombination, *In International Conference on Parallel Problem Solving from Nature* [Berlin Heidelberg: 1994], pp. 78-87.

[88] Mahdad B and Srairi K (2015). Blackout risk prevention in a smart grid based flexible optimal strategy using Grey Wolf-pattern search algorithms, *Energy Conversion and Management*, 98, 411-429.

[89] Zhu, A *et al.* (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.

[90] Yang B *et al.* (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.

[91] Vrionis T D, Koutiva XI and Vovos NA (2014). A genetic algorithm-based low voltage ride-through control strategy for grid connected doubly fed induction wind generators, *IEEE Transactions on Power Systems*, 29(3), 1325-1334.

[92] Bekakra Y and Attous DB (2014). Optimal tuning of PI controller using PSO optimization for indirect power control for DFIG based wind turbine with MPPT, *International journal of system assurance engineering and management*, 5(3), 219-229.

[93] Muangkote N, Sunat K and Chiewchanwattana S (2014, July). An improved grey wolf optimizer for training q-Gaussian Radial Basis Functional-link nets, *IEEE International Conference on Computer Science and Engineering* [Khon Kean, Thailand: 2014] (pp. 209-214).

[94] Chandra M, *et al.* (2016). Web service selection with global constraints using modified gray wolf optimizer, *International Conference on Advances in Computing, Communications and Informatics* [Jaipur, India: 2016], pp. 1989-1994.

[95] Kishor A and Singh PK (2016). Empirical study of grey wolf optimizer. *In Proceedings of International Conference on Soft Computing for Problem Solving* [5th: Singapore: 2016], pp. 1037-1049.

[96] Canfora G, *et al.* (2005). An approach for QoS-aware service composition based on genetic algorithms, *In Proceedings of annual conference on Genetic and evolutionary computation* [7th: Washington, DC: 2005], pp. 1069-1075.

[97] Sharma Y and Saikia LC (2015). Automatic generation control of a multi-area ST–Thermal power system using Grey Wolf Optimizer algorithm based classical controllers, *International Journal of Electrical Power & Energy Systems*, 73, 853-862.

[98] Lal DK, Barisal A K and Tripathy M (2016). Grey wolf optimizer algorithm based Fuzzy PID controller for AGC of multi-area power system with TCPS, *Procedia Computer Science*, *92*, 99-105.

[99] Das KR, Das D and Das J (2015). Optimal tuning of PID controller using GWO algorithm for speed control in DC motor, IEEE *International Conference on Soft Computing Techniques and Implementations*[Faridabad, India: 2015] pp. 108-112.

[100] Sodeifian G, *et al.* (2016). Application of supercritical carbon dioxide to extract essential oil from Cleome coluteoides Boiss: Experimental, response surface and grey wolf optimization methodology, *The Journal of Supercritical Fluids*, *114*, 55-63.

[101] Mohanty S, Subudhi B and Ray PK (2016). A new MPPT design using grey wolf optimization technique for photovoltaic system under partial shading conditions, *IEEE Transactions on Sustainable Energy*, 7(1), 181-188.

[102] Komaki GM and 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.

[103] Song X *et al.* (2015). Grey Wolf Optimizer for parameter estimation in surface waves, *Soil Dynamics and Earthquake Engineering*, 75, 147-157.

[104] Zhang S *et al.* (2016). Grey wolf optimizer for unmanned combat aerial vehicle path planning, *Advances in Engineering Software*, 99, 121-136.

[105] Elhariri E *et al.* (2015). Grey wolf optimization for one-against-one multi-class support vector machines. *IEEE International Conference of Soft Computing and Pattern Recognition* [7th:Fukuoka, Japan: 2015], pp. 7-12.

[106] Medjahed SA, *et al.* (2016). Grey Wolf Optimizer for hyperspectral band selection, *Applied Soft Computing*, 40, 178-186.

[107] Mirjalili S (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons, *Applied Intelligence*, 43(1), 150-161.

[108] Guha D, Roy PK and Banerjee S (2016). Load frequency control of interconnected power system using grey wolf optimization, *Swarm and Evolutionary Computation*, 27, 97-115.

[109] Jayakumar N *et al.* (2016). Grey wolf optimization for combined heat and power dispatch with cogeneration systems, *International Journal of Electrical Power & Energy Systems*, 74, 252-264.

[110] Sultana U *et al.* (2016). Grey wolf optimizer based placement and sizing of multiple distributed generation in the distribution system. *Energy*, 111, 525-536.

[111] Kamboj VK, Bath SK and Dhillon JS (2016). Solution of non-convex economic load dispatch problem using Grey Wolf Optimizer, *Neural Computing and Applications*, 27(5), 1301-1316.

[112] Sulaiman MH *et al.* (2015). Using the gray wolf optimizer for solving optimal reactive power dispatch problem, *Applied Soft Computing*, 32, 286-292.

[113] Chaman-Motlagh A (2015). Superdefect photonic crystal filter optimization using grey wolf optimizer, *IEEE Photonics Technology Letters*, 27(22), 2355-2358.

[114] Shakarami MR and Davoudkhani IF (2016). Wide-area power system stabilizer design based on Grey Wolf Optimization algorithm considering the time delay, *Electric Power Systems Research*, 133, 149-159.

[115] Zuriani M and Yuhanis Y (2015). Time Series Forecasting of Energy Commodity using Grey Wolf Optimizer.

[116] Precup RE, David RC and Petriu EM (2017). Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity, *IEEE Transactions on Industrial Electronics*, 64(1), 527-534.

[117] Chen TB (2005). Synthesis of antenna array using particle swarm optimization, *Proceedings in IEEE* Asia-Pacific *Microwave Conference* [Suzhou, China: 2005], pp. 4.

[118] Zaman MA (2011). Synthesis of antenna arrays using artificial bee colony optimization algorithm, *International Journal of Microwave and Optical Technology*, 6(4), 234-241.

[119] Harrington R (1961). Side lobe reduction by non-uniform element spacing, *IRE Transactions on Antennas and Propagation*, 9(2), 187-192.

[120] Hodjat F and Hovanessian S (1978). Non-uniformly spaced linear and planar array antennas for side lobe reduction, *IEEE Transactions on Antennas and Propagation*, 26(2), 198-204.

[121] Ren XF, Azevedo JA and Casimiro AM (2009). Synthesis of non-uniformly spaced arrays using the Fourier transform and window techniques, *IET microwaves, antennas & propagation*, 3(8), 1245-1253.

[122] Kurup DG, Himdi M and Rydberg A (2003). Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm, *IEEE Transactions on Antennas and Propagation*, 51(9), 2210-2217.

[123] Chen K, He Z and Han C (2006). A modified real GA for the sparse linear array synthesis with multiple constraints, *IEEE Transactions on Antennas and Propagation,* 54(7), 2169.

[124] Rattan M, Patterh MS and Sohi BS (2007). Synthesis of aperiodic linear antenna arrays using genetic algorithm, *IEEE International Conference on Applied Electromagnetics and Communications* [19th:Dubrovnik, Croatia: 2007], pp. 1-4.

[125] Cengiz Y and Tokat H (2008). Linear antenna array design with use of genetic, memetic and tabu search optimization algorithms, *Progress in Electromagnetics Research C*, 1, 63-72.

[126] Khodier MM and Christodoulou CG (2005). Linear array geometry synthesis with minimum side lobe level and null control using particle swarm optimization, *IEEE Transactions on Antennas and Propagation*, 53(8), 2674-2679.

[127] Murino V, Trucco A and Regazzoni CS (1996). Synthesis of unequally spaced arrays by simulated annealing, *IEEE Transactions on signal processing*, 44(1), 119-122.

[128] Guney K and Onay M (2011). Optimal synthesis of linear antenna arrays using a harmony search algorithm, *Expert Systems with Applications*, 38(12), 15455-15462.

[129] Rajo-Iglesias E and Quevedo-Teruel O (2007). Linear array synthesis using an ant-colony-optimization-based algorithm, *IEEE Antennas and propagation Magazine*, 49(2), 70-79.

[130] Saxena P and Kothari A (2016). Linear antenna array optimization using flower pollination algorithm, *SpringerPlus*, 5(1), 306.

[131] Singh U and Salgotra R (2016). Synthesis of linear antenna array using flower pollination algorithm, *Neural Computing and Applications*, 1-11.

[132] Sharaqa A and Dib N (2014). Design of Linear and Elliptical Antenna Arrays Using Biogeography Based Optimization, *Arabian Journal for Science & Engineering*, 39(4).

[133] Singh U, Kumar H and Kamal TS (2010). Linear array synthesis using biogeography based optimization, *Progress in Electromagnetics Research* M, 11, 25-36.

[134] Merad L, Bendimerad F and Meriah S (2008). Design of linear antenna arrays for side lobe reduction using the tabu search method, *The International Arab Journal of Information Technology*, 5(3), 219-222.

[135] Saxena P and Kothari A (2016). Ant Lion Optimization algorithm to control side lobe level and null depths in linear antenna arrays, *International Journal of Electronics and Communications*, 70(9), 1339-1349.

[136] Singh U and Salgotra R (2016). Optimal synthesis of linear antenna arrays using modified spider monkey optimization, *Arabian Journal for Science and Engineering*, 41(8), 2957-2973.

[137] Singh U and Rattan M (2014). Design of linear and circular antenna arrays using cuckoo optimization algorithm, *Progress in Electromagnetics Research C*, 46, 1-11.

[138] Saxena P and Kothari A (2016). Optimal pattern synthesis of linear antenna array using grey wolf optimization algorithm, *International Journal of Antennas and Propagation*.

[139] Mangaraj BB and Swain P (2017). An optimal LAA subsystem designed using Gravitational Search Algorithm, *An International Journal on Engineering Science and Technology,* 20(2), 494-501.

[140] Guney K and Durmus A (2015). Pattern nulling of linear antenna arrays using backtracking search optimization algorithm, *International Journal of Antennas and Propagation*.

[141] Pappula L and Ghosh D (2014). Linear antenna array synthesis using cat swarm optimization, *AEU-International Journal of Electronics and Communications*, 68(6), 540-549.

[142] Oraizi H and Fallahpour M (2008). Nonuniformly spaced linear array design for the specified beamwidth/sidelobe level or specified directivity/sidelobe level with coupling consideration, *Progress in Electromagnetics Research M*, 4, 185-209.

[143] Pal S *et al* (2010). Linear antenna array synthesis with constrained multi-objective differential evolution, *Progress In Electromagnetics Research B*, 21, 87-111.

[144] Goudos SK *et al.* (2010). Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with side lobe level suppression and null control, *IEEE antennas and wireless propagation letters*, 9, 125-129.

[145] Cen L *et al.* (2012). Linear aperiodic array synthesis using an improved genetic algorithm, *IEEE Transactions on Antennas and Propagation*, 60(2), 895-902.

[146] Chowdhury A *et al.* (2010). Linear antenna array synthesis using fitness-adaptive differential evolution algorithm, *IEEE Congress on Evolutionary Computation* [Barcelona, Spain: 2010], pp. 1-8.

[147] Subhashini KR and Satapathy JK (2017). Development of an Enhanced Ant Lion Optimization Algorithm and its Application in Antenna Array Synthesis, *Applied Soft Computing.*

[148] Goudos SK *et al* (2010). Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with side lobe level suppression and null control, *IEEE antennas and wireless propagation letters*, 9, 125-129.

[149] Pappula L and Ghosh D (2013). Large array synthesis using invasive weed optimization. In IEEE *International Conference on Microwave and Photonics* [Dhanbad, India: 2013], pp.1-6.

[150] Pappula L and Ghosh D (2014). Constraint-based synthesis of linear antenna array using modified invasive weed optimization, *Progress in Electromagnetics Research M*, 36, 9-22.

[151] Guney K and Basbug S (2014). Linear antenna array synthesis using mean variance mapping method, *Electromagnetics*, 34(2), 67-84.

[152] Tizhoosh HR (2005). Opposition-based learning: a new scheme for machine intelligence, *International Conference on intelligent agents, Web,Technologies and Internet Commerce* [Vienna, Austria: 2005], pp. 695-701.

[153] Kentzoglanakis K and Poole M (2009). Particle swarm optimization with an oscillating inertia weight, *Annual Conference on Genetic and evolutionary computation* [11th :UK: 2009], pp. 1749-1750, 2009.

[154] Rahnamayan S, Tizhoosh HR and Salama MM (2008). Opposition−based differential evolution. *IEEE Transactions on Evolutionary computation*, 12(1), 64−79.

[155] Yusof Y and Mustaffa Z (2015). Time Series Forecasting of Energy Commodity using Grey Wolf Optimizer, *Proceeding of International Multi Conference of Engineers and Computer Scientists* [Hong Kong: 2015], pp. 2-9.

[156] Derrac J *et al*. (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.

[157] Van den Bergh F and Engelbrecht AP (2006). A study of particle swarm optimization particle trajectories, *Information sciences*, 176(8), 937-971.

LIST OF PUBLICATIONS

[1] Sharma S, Salgotra R and Singh U (2017). An enhanced grey wolf optimization algorithm for numerical optimization, *IEEE International conference on the innovation, embedded and communication systems* [4^{th}: Coimbatore, India: 2017]. (Status- Accepted)

[2] Sharma S, Singh U and Salgotra R. On the improvement of grey wolf optimization, *In Applied Intelligence.* (Status- in communication)

Sakshi_17.7.17

ORIGINALITY REPORT

%3	%3	%1	%1
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	uvb.nrel.colostate.edu Internet Source	%2
2	handcorpus.org Internet Source	<%1
3	www.docstoc.com Internet Source	<%1
4	Submitted to Cranfield University Student Paper	<%1
5	Wang, Dan Xiong, Congcong Zhang, Xiankun. "An opposition-based group search optimizer with diversity guidance.(Research Article)(Technical repo", Mathematical Problems in Engineering, Annual 2015 Issue Publication	<%1
6	Jian feng Qiu; Ji wen Wang; Dan Yang; Juan Xie and Nan zhen Yao. "A Parameter Adaptive Artificial Bee Colony Algorithm for Real-Parameter Optimization", International Journal of Online Engineering, 2013. Publication	<%1

7	econstor.eu Internet Source	<%1
8	www.911tabs.com Internet Source	<%1
9	Engineering Computations, Volume 21, Issue 1 (2006-09-19) Publication	<%1
10	rksi.info Internet Source	<%1
11	Wang, Lijin Zhong, Yiwen. "Cuckoo search algorithm with chaotic maps.(Research Article)(Report)", Mathematical Problems in Engineering, Annual 2015 Issue Publication	<%1
12	dspace.thapar.edu:8080 Internet Source	<%1
13	asd-www.larc.nasa.gov Internet Source	<%1
14	www-nds.iaea.or.at Internet Source	<%1
15	www.jamstec.go.jp Internet Source	<%1
16	www-amdis.iaea.org Internet Source	<%1