DESIGN OF UNIFORM LINEAR ANTENNA ARRAY USING ENHANCED MOTH FLAME OPTIMIZATION ALGORITHM

A Dissertation Submitted in Partial Fulfillment of the Requirement for the Award of the Degree of

MASTER OF ENGINEERING

in

Electronics and Communication

Submitted By

KOMALPREET KAUR

801561013

Under Supervision of

Dr. URVINDER SINGH

Assistant Professor, ECED

Thapar University, Patiala

ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA, PUNJAB

JULY, 2017

DECLARATION

I, Komalpreet Kaur hereby declare that the work presented in this thesis entitled " DESIGN OF UNIFORM LINEAR ANTENNA ARRAY USING ENHANCED MOTH FLAME OPTIMIZATION" in partial fulfillment 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, ECED, Thapar University) from 2015 to 2017. The matter presented in this this has not been submitted either in part or full to any other university or institute for the award of any other degree.

Komalpreet

Date: 17-07-2017

Komalpreet Kaur
801561013

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

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 express my heart full indebtedness and owe a deep sense of gratitude to my teacher and my faculty guide **Dr. Urvinder Singh,** Assistant Professor, ECED for their sincere guidance and support with encouragement to go ahead.

I am also thankful to **Dr. Alpana Aggarwal**, Professor and Head, ECED, for providing us with the adequate infrastructure for carrying out the work. I am also thankful to **Dr. Amit Mishra**, Assistant Professor, ECE Program Coordinate and **Dr. Hem Dutt Joshi,** Assistant Professor, P.G. Coordinator, ECED, for the motivation and inspiration and that triggered me for the work.

I would also like to thank my friends **Ms. Sakshi Sharma** and **Ms. Deepika Singh** 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.

KOMALPREET KAUR

ABSTRACT

In this work, a new optimization algorithm named as enhanced moth flame optimization (E-MFO), has been designed by modifying the basic moth flame optimization (MFO). This proposed algorithm has been employed for the synthesis of uniform linear antenna arrays (LAA). In E-MFO, four modifications have been proposed to overcome the drawbacks of MFO. A random attraction model is added to enhance the exploration capability, influence of best flame has been incorporated to improve the exploitation. Moreover, an adaptive step size and division of iterations is applied to maintain a good balance between the exploration and exploitation. To validate the applicability of E-MFO, it has been applied to twenty benchmark functions and results are compared with other meta-heuristic algorithms like BA, DE, FA, FPA and BFP. The effect of population and dimension size on the performance of E-MFO has been discussed. Also, statistical testing of E-MFO has been done to prove its significance. The numerical results show the superior performance of E-MFO over other algorithms in terms of convergence rate and solution quality.

LAA is a difficult and non-linear electromagnetic problem. Hence, to verify the performance of E-MFO, it is used to synthesise the complex LAA design problem. The basic goal of the antenna design is to achieve minimum side lobes and null steering by optimizing the amplitude excitations. The performance of E-MFO has been evaluated by using seven different examples of LAA and results are compared with other well-known algorithms. The results show that E-MFO outperforms other algorithms in terms of reduction of side lobes and null control.

TABLE OF CONTENTS

LISTS OF TABLES

LIST OF FIGURES

LIST OF ABBERVATIONS

ABC	Artificial Bee Colony
ACO	Ant Colony Optimization
ALO	Ant Lion Optimization
BA	Bat Algorithm
BBO	Biogeography Based Optimization
BFP	Bat Flower Pollination Algorithm
BSA	Backtracking Search Algorithm
CCAA	Circular Concentric Antenna Array
CS	Cuckoo Search
CSO	Cat Swarm Optimization
DE	Differential Evolution
EA	Evolutionary Algorithm
EP	Evolutionary Programming
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
HAS	Harmony Search Algorithm
IFPA	Improved Flower Pollination Algorithm
LAA	Linear Antenna Array
MSMO	Modified Spider Monkey Optimization
MTACO	Modified Touring Ant Colony Optimization
NIA	Nature Inspired Algorithms
PSO	Particle Swarm Optimization
SI	Swarm Intelligence
SOA	Seeker Optimization Algorithm
SLL	Side Lobe Level
WOA	Whale Optimization Algorithm

CHAPTER 1

INTRODUCTION

1.1 ANTENNA ARRAY

Guglielmo Marconi was the first scientist who received first radio massage in December 12,1901 [1]. The message contained the Morse code for the letter 'S'. This was the most significant invention in radio and wireless communication. There had been so much improvement in communication system after that time. Developments in field of electronics, signal processing and antenna theory significantly contributed to the upgrading wireless communication system. Antenna is key element in all sort of communication like radar, wireless, radio etc. It is a device which is used for transmitting and receiving the electromagnetic signals. A single antenna is not enough for communication process because it does not have appropriate gain and capability to cover the entire geographic area. Hence, single antenna is replaced by large single aperture antenna to increase the efficiency of it. The disadvantage of using large aperture antenna is that it increases the design cost and also it is not easy to move such large antenna from one place to another. So, to overcome this problem antenna array was designed in 1940s for military applications [2]. Antenna arrays provide significant improvement in communication system by providing accurate transmission and reception of signals. Antenna array is basically an assembly of small antennas which are arranged in electrical or geometrical configuration. The antennas are arranged in such a way that their radiating fields converge in desired direction and cancel out each other's effect in other space. So, in desired direction output is equal to the sum of individual antenna gains. To represent the antenna array in co-ordinate system, **N** number of elements are chosen and positioned on coordinate system. The positions of elements are defined according to equation 1.1 and Figure 1.1.

$$D_n = [x_n \; y_n \; z_n] \tag{1.1}$$

Suppose the output of N elements is $[X_1 \; X_2 \dots \dots \dots X_N]$ and, also get multiplied by weights $[w_1 \; w_2 \dots \dots \dots w_n]$, then output will be

$$y = \sum_{n=1}^{N} w_n X_n \tag{1.2}$$

So, the output of antenna array is summation of all the outputs as shown in Figure 1.2. Antenna array can also tackle cost and shifting problem efficiently. Hence, antenna arrays are widely used in radar and wireless communication.

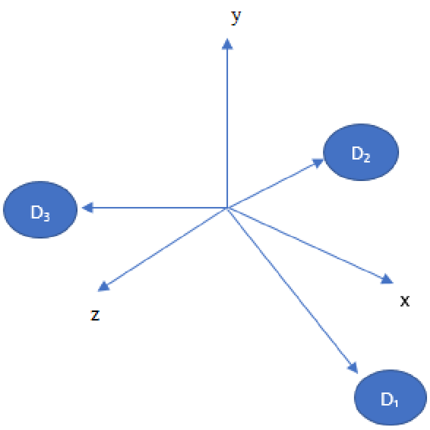


Figure 1.1 Positions of antenna elements

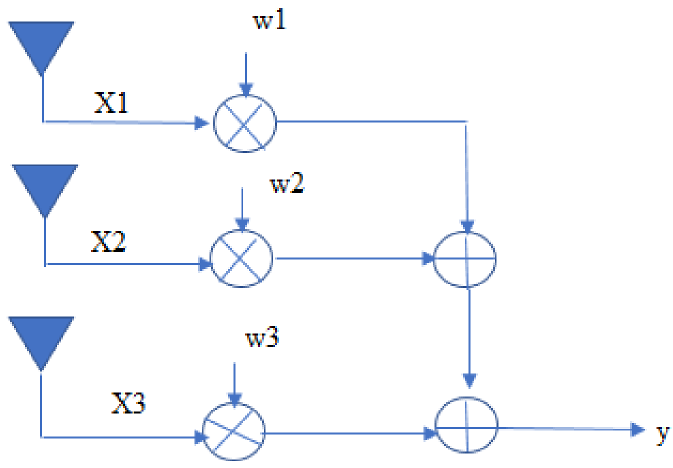


Figure 1.2 Arrangement of antennas showing weighted sum output

1.1.1 Features of antenna array

Antenna array has some interesting features due to which it is popular these days. The features are:

a) Antenna gain is basically the measure of power transmitted in the direction of maximum radiation to that of isotropic antenna. Single antenna provides less gain whereas antenna array provides maximum gain as well as signal to noise ratio (SINR).
b) Antenna arrays provide diversity reception. In wireless communication, fading is the process, in which signal power vary according to the distance. So, to overcome the fading multiple antenna are combined to achieve the maximum signal. Figure 1.3 shows the variation of power with respect to distance. Here three antennas are placed at certain distance and antenna three will be selected because it receives the maximum power.
c) The radiation pattern of single antenna is fixed but in antenna array the radiation pattern can be changed by providing different currents to different elements. Hence, desired radiation pattern can be achieved by changing excitations of different elements. This characteristic provides freedom to choose desired pattern without changing physical configuration.
d) The output signals of antenna arrays can be used for further applications like target tracking, gain enhancement etc.
e) It can scan the large geographic area without moving the antenna itself.

f) It has capability of achieving adaptive beamforming by arranging array elements in proper configuration.

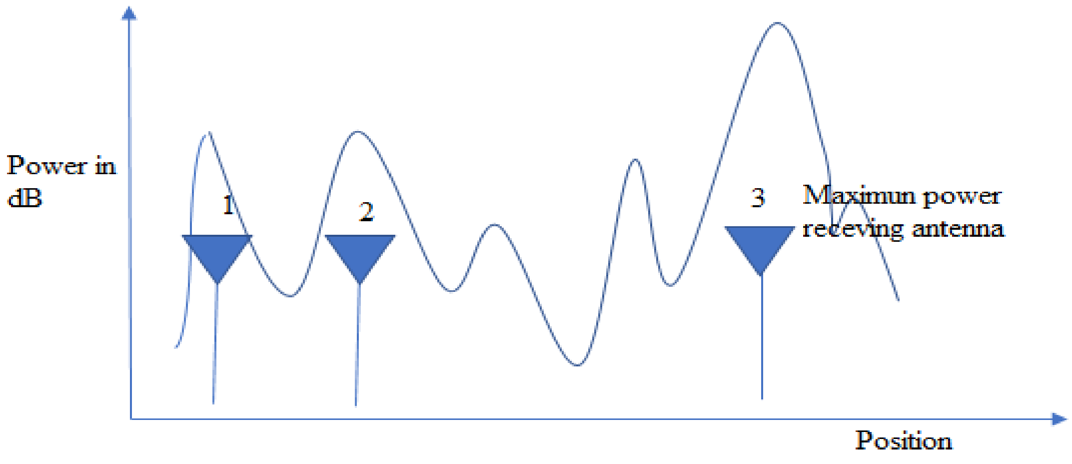


Figure 1.3 Variation of power with respect to position in three element arrangements

1.1.2 Types of antenna array

The antenna array is of different types based on geometrical configuration like linear, circular and planer. The types are as follows:

Linear antenna array (LAA): In linear array, elements are arranged in one line with either equal or unequal spacing. Linear arrays have simple geometry and they are also easy to handle.

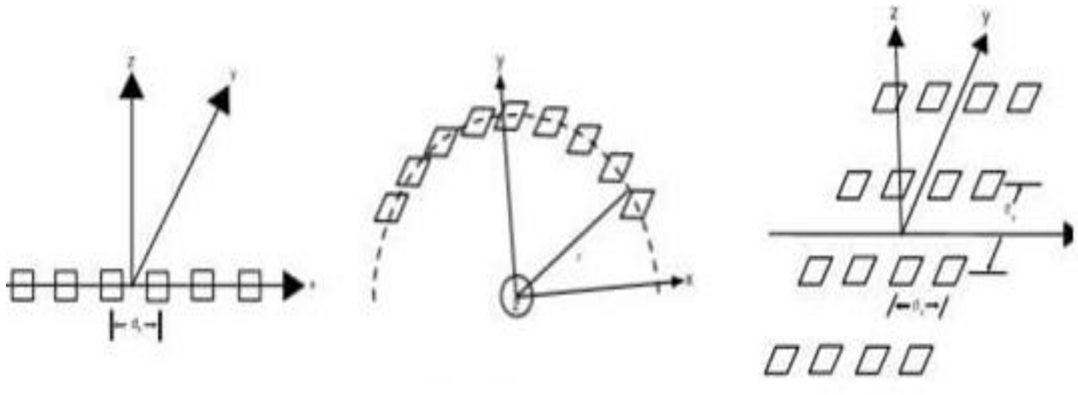
Circular antenna array (CAA): The circular antenna array is basically 1-D linear array arranged in circular form. In circular array N numbers of antennas are arranged in form of circle. Circular arrays are widely used in satellite and mobile communication. The main advantages of circular antenna array are:

a) It can provide 2-D angular scan, both horizontal and vertical scans.
b) Circular array can scan 360 ° in horizontal direction without any distortions.
c) In case of circular array, it is easy to deal with mutual coupling effect in comparison to linear array.

Concentric Circular antenna array (CCAA): Concentric array is structure, in which several circular arrays of different radii are arranged. These concentric rings have specified number of elements. CCAA has been widely used in mobile and wireless communication [3-4]. CCAA has various advantages:

a) CCAA has more flexibility in designing and synthesizing array pattern. So, it can be used in both narrowband and broadband applications.
b) It provides invariant azimuth angle converge in direction of arrival (DOA).

Planar array: In planar array, the radiating elements are arranged in such a way that it forms a rectangular grid. It provides simple elevation scanning and full antenna gain. It can be used to control the beam-width by changing phase of radiating elements [5]. Planar array is not as popular as other types of antenna array.



Linear array Circular array Planar array

Figure 1.4 Types of antenna array [6]

Antenna array can also be classified based on interelement spacing. Hence, there are two type of arrays one is uniformly spaced and other is non-uniformly spaced antenna array. In uniform spaced antenna array the amplitude and phase is controlled while keeping distance between two consecutive elements constant and in non-uniform spaced antenna array reverse procedure is followed. The radiation pattern in desired direction depends upon few parameters like spacing between two array elements, magnitude and phase of excitation current of each element and geometrical configuration. By controlling these parameters, any antenna array can be designed. The physical geometry of antenna effects the performance of antenna array so, there is need to optimize the antenna parameters by changing its geometry.

1.1.3 Synthesis of antenna array

The aim of synthesis of antenna array is to design a geometrical configuration whose radiation pattern is almost same as desired pattern. The desired pattern may vary according to the application. Most of synthesis methods deal with suppressing side lobe levels (SLL). Side lobes cause interference to main lobe so practical antenna array should have minimum side lobes while keeping directive beam or gain in desired direction. This can be achieved by controlling magnitude and phase of excitation current. Some of the other synthesis methods deals with null steering. Null steering is a technique of jamming the signals from undesired directions by keeping their radiation energy zero or by placing nulls in that direction. The concept of antenna array makes possible to receive or transmit signal in one direction while rejecting signals from other directions. In this present work, null steering is done by controlling amplitude of radiating elements using meta-heuristic algorithm inspired from the work represented in [7]. This can be achieved by installing attenuators [8] with antenna array. The advantage of using amplitude only control is that, it is easy implement and less sensitive to quantization error [9]. On the other hand, disadvantage of amplitude control method is, its non-linear nature and difficult to solve by analytic methods. Due to drawbacks of analytic methods, synthesis of LAA is done by using meta-heuristic techniques.

1.2 OPTIMIZATION

Optimization is the process of finding best solution for a problem or objective function under given constraints. The main aim of optimization is to achieve highest achievable performance by maximizing the desired results and minimizing the undesired ones. Hence, it is minimization and maximization of function within certain boundaries limits. Optimization is used in various real-world problems for their design and maintenance. The concept of solving optimization problems came into light in 300 BC when Euclid solved several problems associated to their geometrical work. He concluded from their work that minimum distance between two points is the length of straight line joining them. When there was no existence of calculus of variations, only problems like determining optimized dimensions of the wine barrel [10] and time taken by light to travel between two points [11], were solved. Afterwards, mathematical analysis was created by Newton and Leibniz [12] on the basis of differential calculation to solve finite set of optimization problems. In 1740, new method was proposed on advance and general theory of calculus of variations by Euler [13]. Further, optimization method for constrained problems was invented by Lagrange [14] and for unconstrained problems Cauchy [15] gave the first method known as gradient method. These methods were further followed by Simplex method [16], but this method fails in solving various engineering tasks. So, McCormick designed new method, inspired from techniques of unconstrained optimization. These all methods work on single programming model which means it can optimize only one single objective function. Therefore, for multi objective functions, multi programming models were developed [17].

1.2.1 Components of optimization problem

Any optimization problem depends on some components to reach at optimum value. The first and foremost component is objective function. The objective function is a quantity or mathematical equation with decision variables that need to be minimized or maximized. Basically, objective function tries to maximize the profit and minimize the loss based on number of variables and constraints. The constraints can be like resources, capacity, availability, technology etc. for any real-world problem. Any optimization problem may have single objective function or multi objective function. The problem arises while working with multi objective functions because of compatibility issues between different objectives. So, to solve multi objective function, two methods are possible one is to form weighted combinations of objectives and second is to treat some of objectives as constraints. Further objective function depends on number of variables in equation. Variables are those quantities or attributes of mathematical function which can take any value. In other words, variables are set of unknown quantities which define the objective function and its constraints. While choosing the values for variables, it is required that those values must satisfy the functional and other requirements. There are three types of design variables named as continuous, discrete and boolean.

The next important component of optimization problem is constraints. These are basically set of conditions which must be satisfied while designing optimization problem. These conditions allow only certain values of design variables so that objective function comes up with desired results. So, to design any optimization problem its components like variables, constraints, objective function and relationship between them must be defined a prior. The non-linear, single- objective and constrained optimization problem is described below:

$$Minimization\ of\ F(x), x = [x_1, x_2, x_3, x_4 \dots \dots \dots x_p] \quad (1.3)$$

This problem deals with minimization of objective function *F(x)* where *x* is vector set of *p* design variable. It is subjected to few constraints like

$$g_j(x) \leq 0 \text{ for } j\text{=1 to } m \quad (1.4)$$

$$h_k(x) = 0 \text{ for } k\text{=1 to } n \quad (1.5)$$

$$x_{iL} \leq x_i \leq x_{iU} \text{ for } i\text{=1 to } p \quad (1.6)$$

In above equations $g_j(x)$ represents the inequality constraint, $h_k(x)$ represents the equality constraint and x_{iL}, x_{iU} are lower and upper boundary conditions respectively. These boundary conditions are also called as side constraints. The modifications are carried out on *p* design variables to reach at global optimum point.

1.2.2 Classification of optimization problems

Optimization problems can be classified based on type of constraints, nature and number of design variables and type of objective functions. Various types of problems are discussed below:

- *Problems based on constraints:* The mathematical process in which objective function is optimized in presence of some constraints is called constrained optimization. If there are no constraints then that problem come under the category of unconstrained optimization.

- *Problems based on nature of equations:* The objective function and constraints of any optimization problem is described as mathematical equations. So, optimization problems can be categorized according to the nature of those equations. On the basis of computational point of view of equations, there are three categories which are:
  - Linear programming problem (LPP): In this case, objective function and constraints are linear function of design variables. In other words, LPP is optimization technique in which linear objective function, subject to linear constraints.
  - Quadratic programming problem (QPP): If objective function is quadratic in nature and constraints is linear function of optimization variables then that particular problem said as QPP. To solve such problems, extension methods of LPP is employed.
  - Nonlinear optimization problem (NLP): This optimization problem includes one or more nonlinear functions among objective and constraints function. Most of real world problems fall under this category.
- *Problems based on decision variable:* These kinds of problems are grouped into two parts which are:
  - Integer programming problem: If decision variables of an optimization problem are allowed to take only integer values then the problem is called an integer programming problem. For example, optimization is to find the number of articles required for operation with minimum efforts. Now, the minimum efforts are the required objective function which further depends on the design variables i.e. number of articles. Here number of article must take only integer values.
  - Real valued programming problem: This problem belongs to optimization of real function either by maximizing it or minimizing it. This real function depends on real variables whose values are chosen from one allowed set. If that set contain only real values then that problem is called real valued programming problem.
- *Problems based on number of objective functions*: Objective functions are of two types single objective and multi objective function.
  - Single objective problem: When there is only one objective function in optimization problem then that problem termed as single objective problem.
  - Multi objective problem: These problems can be defined as:

    Find x which minimizes $f_1(x), f_2(x), f_3(x) \dots \dots \dots \dots f_k(x)$; subject to $g_j(x) \leq 0$, j=1, 2…m

    here $f_1, f_2, f_3 \dots \dots \dots .. f_k$ are the objective functions that need to be optimize. There are also m numbers of constraints. In multi objective function one term which is widely used known as 'optimum'. This term was proposed by Pareto and describes that one decision variable x is called optimum if there does not exist any other variable providing less objective function value than that. Suppose there exist another variable y which satisfies the condition $f_i(y) \leq f_i(x)$ for $i = 1, 2, 3 \ldots \ldots k$, from this it is observed that now x is not optimum variable because objective function is providing minimum value at variable y.

- *Problems based on nature of optimization*
  - Global or local optimization problems: The problems can also be classified on the basis of number of local minima. If in one problem only one local minima exist then it is called local optimization problem. Sometimes problem is consists of number of local minima and the main task is to find smallest among all minima which is called global minima. This type of optimization is called global optimization problem. The Figure 1.5 shows the difference between local and global minima.
  - Deterministic or stochastic optimization problems: In deterministic problems, all the information regarding decision variable is known accurately whereas in stochastic optimization, decision variables are random in nature. The uncertainty is added to stochastic optimization to make it more robust.

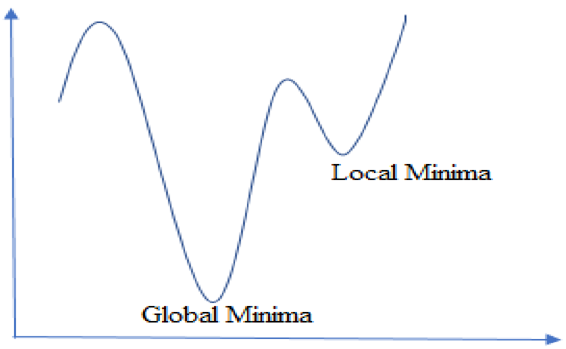


Figure 1.5 Local minima and Global minima

1.2.3 Solution of optimization problems

Most of real world problems are nonlinear in nature and has complicated objective function. Sometimes, there feasible solutions are not available due to complexity. In such cases, objective function is computed by taking one trail solution and then that solution subsequently improved by using optimization algorithms. There are so many optimization techniques but choice of suitable technique depends on type of problem. Firstly, we will discuss the classical optimization techniques and then advance optimization techniques.

Classical Techniques: The classical optimization techniques are useful in finding optimum solution for continuous and differential problems. Basically, these techniques make use of differential calculus to locate the optimum locations. Classical techniques are useful for solving single and multi-dimensional problems. Few of them are discussed below:

a) Direct method: Direct methods are simple techniques which does not involve the partial derivative of functions that's why called as non-gradient techniques. It only depends on countable set of functional values of objective function. For one-dimensional problems, golden section search and quadratic methods can be used whereas for multi-dimensional problems random search and univariate methods can be employed. The golden section search is simple and general-purpose method which starts from the guessing the lower and upper bounds. Afterwards new solutions are created according to golden ratio $\left(\frac{\sqrt{5}-1}{2}\right)$ and objective function is evaluated there. Quadratic interpolation method works on the basis of that there will be only one quadratic connecting three points. This technique often provides good results near to optimum ones. Random search methods evaluate the function at random locations of variables and generate the trail solutions using random number generators. Hence, random solutions will able to find the optimum solutions effectively. There are three type of random methods random jump method, random walk method and random walk with direct exploitation. In random jump method, random solutions are generated with uniform distribution for decision variables and then best solution is chosen among them according to the best objective function. Random walk method is improved version of random jump method with modifications like scalar step size and a unit random vector. The random walk with direct exploitation method consists of two parts first is to find out the direction of best trail solutions and then to take maximum possible steps in that direction. The main drawback of random search methods is that it can re-evaluate the same points in same direction because of randomness. To overcome these drawbacks, univariate methods are designed in which only one decision variable change its value while keeping the other constant. This method is more efficient than random method. In this method search direction is along with coordinate axis which makes rate of convergence very slow so to improve the convergence, search direction is taken towards the best solution and that method is called pattern direction method.

b) Indirect method: This method is also called gradient method because it exploits the first and second order derivatives of decision variables. First derivative of variable provides its slope and at local minima its value become zero. Basically, slope indicates the direction to move locally. The gradient or slope of variable can be calculated using Newton's method.

$$x_{i+1} = x_i - \frac{f'(x_i)}{f''(x_i)} \tag{1.7}$$

The gradient of variable is represented as $\nabla f^T = \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2} \dots \dots \dots \frac{\partial f}{\partial x_n}\right]$. For number of iterations, solution will vary according to equation

$$x_{k+1} = x_k + \alpha_k . D_k \tag{1.8}$$

where $\alpha_k$and D_k are step size and direction vector respectively. This gradient based search is consists of two steps, first to find the search direction using derivative information and second step is to move in that direction with step size α_k . In steepest gradient method, ∇f is chosen as direction vector whereas in conjugate method direction vector is chosen in such a way that two successive direction vectors can never be in same direction.

Advanced optimization techniques: Classical techniques fails in solving non-linear and non-differentiable problems. Hence, there is need for other techniques which can efficiently solve these problems. Advance optimization techniques were designed in same context which include several population based stochastic optimization techniques. These techniques work on two factors which are exploration and exploitation [18] which are explained below:

a) Exploration: Exploration is the process of searching the whole search space and then generating the diverse solutions to find the global optima. In meta-heuristic algorithms exploration process is carried out based on randomization that ensures the diversity in solutions. It also helps the solutions to jump out of local minima so that they can explore globally in search space. There are few simple and complex methods for randomization process like uniform distribution, gaussian distribution, Brownian random walk and Levy flights. This process is also called diversification.
b) Exploitation: Exploitation is the process of searching the global optima in local region. During exploration process information is gathered about the global minima whereas in exploitation that information is exploited and the search is limited to only promising regions. The new search is conducted usually around the best solution found so far. Exploitation uses the local information like gradients and history of search algorithm.

These are two main aspects of any optimization algorithm which defines the efficiency of algorithm. The simulation behavior of algorithms suggests that strong exploitation increase the convergence speed while strong exploration decreases the convergence speed. On the other hand, strong exploration increases the chances of finding global minima and too much exploitation fails to avoid local minima problem. Hence, there is need to maintain the fine balance between these two components. From last few decades, many new optimization algorithms were designed which shows different degree of exploration and exploitation. The first nature inspired algorithm was genetic algorithm (GA) [19] which follows the Darwin's theory of evolution. Then, differential evolution (DE) [20] was designed which is same as GA with special mutation and crossover operators. Some algorithm works on foraging behavior of animals like ant colony algorithm (ACO) [21] and particle swarm optimization (PSO) [22] which mimics the social behavior of ants and flock of birds respectively. Artificial bee colony (ABC) [23] is also based on foraging behavior of honey bees. Gray wolf optimization (GWO) [24] exploits the hunting behavior of wolfs in nature. According to no free lunch (NFL) [25] theorem, no algorithm is best suited for all the applications so the research is still going on. The research on enhancing performance of optimization algorithms is also going on and become popular in industry.

1.3 NATURE INSPIRED ALGORITHMS

Classical optimization techniques are useful in solving continuous and differential problem with help of differential calculus. But, it has some drawbacks as discussed below:

a) These techniques are dependent on starting point.
b) Classical techniques suffer from premature convergence and it also converges to local minima or revisit the same optima, very often.
c) These are only valid for continuous and differentiable problems and used to find only local minima.
d) Gradient method and linear programming methods require only continuous and differentiable objective function, which increases the complexity of algorithm.
e) Moreover, these techniques are difficult to implement. So, it has limited scope for practical applications.

Sometimes, the search space for the problem is very large that finding global minima is difficult task. So, there is need for other techniques which can deal with large search area, mixed variables etc. Nature inspired algorithms (NIA) were defined in the same context to deal with the problems of classical optimization algorithms.

It is fact that nature has served as a great source of inspiration for human to solve real world problems from millions of years. There are so many real-world processes which depict the nature inspired computing like decision making, immune system, collective behavior, learning etc. Based on these phenomena various nature inspired optimization algorithms were designed. Presently, nature inspired algorithms have become popular in almost every area of research because of their faster and flexible nature. The main reason for their popularity is that these algorithms are population based and don't require any gradient information for finding the global solution. Another reason for their popularity is that they don't require any initial guess as required by their classical counterparts. For efficient NIA, it should have some properties as discussed below:

a) The algorithm should reach to global minima in less number of iterations and function evaluations. This property helps the algorithm to solve global complex problems.
b) The algorithms must be able to improve the newly generated solutions from previous ones.
c) It should have capability to cover the entire search space to find global minima.
d) An algorithm must be able to escape the local optima so the it can't be trapped at local minima forever.

These properties lead to good optimization algorithm with higher efficiency. There are so many NIA algorithms which are further divided into two parts. It has few advantages like easy to implement, has more ability to reach near global minima, robust and well suited for discrete problems. NIA mainly grouped into two categories namely Swarm intelligent algorithms (SIA) and Evolutionary algorithms (EA).

1.3.1 Swarm Intelligence

SI has gain great popularity in industry due to its capabilities of solving non-linear and non-differentiable engineering problems from past few years. The concept of SI was firstly introduced by Gerardo Beni and Jing Wang in 1989 [26]. They have described SI according to metaphor of cellular robotic system. After a decade, Bonabeau [27] refine the concept of SI and presented as a method of solving problems, motivated from collective behavior of swarm such as group of insects, ants, bees, birds etc. The meaning of swarm is group of individuals which communicates with each other for their survival. Basically, swarm interacts with each other and also with surroundings in search of food or prey. The two main capabilities of swarms, which inspired SI, are group forming and self-organization. They live in group so that they can deal with several problems together like protection from predator, searching of food when there is scarcity of food. While, self-organization is the feature by which swarms interact with each other under the control of a central authority or external elements. The two main properties of SI are self-organization and division of labor, which are necessary for solving complex problems in nature.

1. Self-organization

   Self-organization is mechanism which provides results at global level because of interaction among low level components of system. SI has defined some set of rules for interaction among components. These rules ensure that information exchange between components must be on the basis of local data and there should be no connection between local and global data. Self-organization is the feature by which swarms interact with each other under no control of a central authority or external elements. The four basic rules of SI are discussed below:

   a) Positive feedback: This rule defines the activation of some popular actions, taken by individuals, so that other individuals can inspire from these actions and do same. Examples of positive feedback is trail lying by ants, waggle dance of honey bees.
   b) Negative feedback: If there is positive feedback, then there is need of negative feedback to control its effect. These both process helps to make system more stable. In this system, individuals avoid to follow same actions or behavior to maintain the stable equilibrium.
   c) Fluctuations and randomness: The process of fluctuations make the system jump out of local minima and hence, helps to avoid local stagnation problem. Randomness defines the diversity of system by generating new solutions in different directions.
   d) Multiple interactions: There should be minimum number of interactions among individuals at low level so that it can make system work at global level. It helps the individuals in communication with each other and also for their survival.
2. Division of labor: According to this concept, the task is divided and allocated to special individuals. These special individuals perform these tasks simultaneously in groups by cooperating each other. The work performed in simultaneous manner is efficient and better than sequential manner. The purpose behind division of task is to reach at global minima even if solution get stuck at local minima. The advantages of SI are given in Figure 1.6.

There are various optimization techniques under category of SI like PSO [22], ABC [23], ACO [21] and FA [28]. For better understanding of the concept of SI, one of the algorithm is discussed here which is firefly algorithm (FA). FA is derived from the firefly species found in nature.

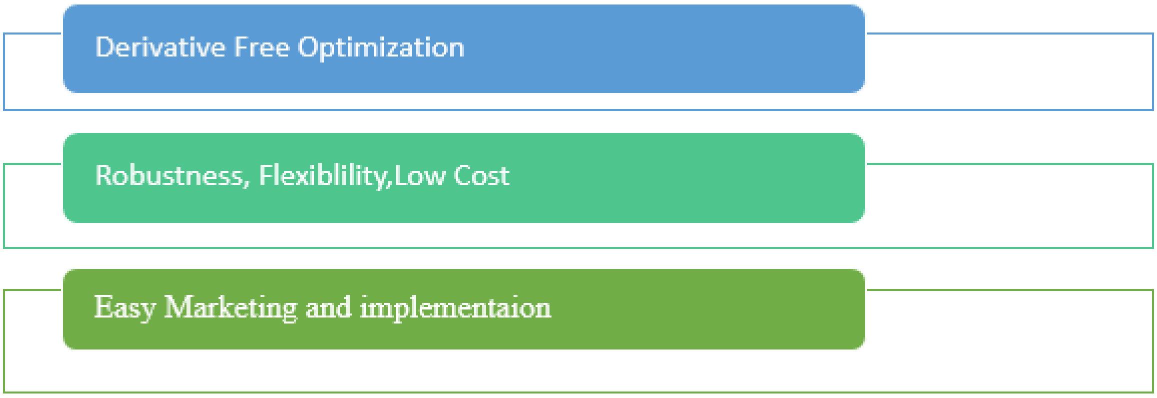


Figure 1.6 Advantages of Swarm Intelligence

Fireflies are winged insects which are mostly found in mild and tropical regions as shown in Figure 1.7 and 1.8. They produce small and periodic flashes of light due to bioluminescence process. Currently, there are approximately two thousand species of fireflies and most of them can produce light. They are usually active in nights. One interesting thing about this flashing is that each specie has their unique flashing behavior. This light is actually acts as a signal to find the prey, for communication purpose and to attract the mating partner. This flashing light also helps to warn other fireflies about predators. Furthermore, male fireflies try to attract females by sending signals and in response of that signals females produce continuous flashing lights. Mostly females favor in attracting towards brighter males.

Figure 1.7 Group of fireflies [29]

Figure 1.8 Firefly species [29]

According to swarm intelligence (SI), swarms live together in groups and communicate with each other for their survival. FA also follows the characteristics of SI and gives priority to reproduction instead of foraging. FA was introduced by Yang in 2010. It is population based meta-heuristic algorithm which mimics the social behavior of fireflies.

To implement FA, some characteristics are idealized by using basic three rules which are,

1) All the fireflies attract each other regardless of their sex. So, basically all the fireflies have same sex.
2) Fireflies attract each other on the basis of brightness and distance between them. Attractiveness is basically directly proportional to brightness so the less bright firefly will be attracted towards other brighter fireflies. Moreover, attractiveness is inversely proportional to distance. According to this, the fireflies which are separated by larger distance will less susceptible to change its position towards brighter firefly. If there is no brighter firefly than particular firefly, it will move randomly in search space.
3) Brightness can be determined by the nature of objective function. Suppose, for minimization problem, brightness can be reciprocal of the objective function. It describes that a brighter firefly has minimum objective value. On the other hand, for maximization problem brightness has direct relationship with objective function.

1.3.2 Evolutionary based algorithms

Evolutionary algorithms (EA) are usually used to solve high dimension and non-linear problems. They have ability to reach at global minima in less number of iterations for large scale optimization problems. EAs inspire from the natural or biological phenomenon like flocking of birds, foraging behavior, mating behavior. EA is stochastic population based search algorithm which mimics the natural process of evolution. The major property of EA is that it works with number of possible solutions called population. It will help to explore the search space efficiently and avoiding premature convergence. The purpose of these algorithms is to find the fittest individual among all solutions and pass it to the next generation. The reason behind this is that best solutions from previous generation have more chances to produce more optimized solutions. It also follows the survival of fittest theory because only fittest solutions get passed to next generation, other population die or try to re-initialize itself. The solutions passed to next generation after applying two operator recombination and mutation. Recombination operator is applied on two or more solutions, which is called parents, and come up with new one or more solutions. Mutation operator is applied on only one chosen solution and it results into one new solution with different properties. The new solutions produced after applying operators is called offspring. These offspring compete with previous solutions to make its place in next generation. This process keeps on repeating itself until certain termination criteria is satisfied. The two main processes carried out in EAs are:

- Selection: Selection is force pushing quality and it ensures that only better candidates are chosen and passed to next generation. It increases the chances of reaching at global minima in less time.

- Recombination: Recombination process generates variation in population Hence, maintains the diversity in population.

During selection, there are more chances of selecting better candidates but even then, there are few chance of selecting candidates with low fitness function. They also have opportunity to become parent or to survive in nature. The candidates chosen for mutation and recombination is purely random process that increase the randomness in EAs. Also, EA is the process of learning and adaption in environment. Fitness function is actually an expression for the environmental requirements. So, meeting these requirements is primarily task for the candidates. EA makes the population adapt to environment in better way. The examples of evolutionary techniques are evolutionary programming (EP) [30], biogeography based optimization (BBO) [31], ant lion optimizer (ALO) [32], evolutionary strategy (ES) [33] and moth flame optimization (MFO) [34]. The one of the evolutionary algorithm named moth flame optimization is discussed in this section to better understanding of evolutionary algorithms. MFO is latest algorithm which has been employed to various complex problems. Hence, in this work MFO is used to meet the desire goals. Basically, moths are winged insects and belong to the family of butterflies. There are almost 160,000 species of moths present in nature. There are two stages in the lifetime of moths namely larvae and adult. In initial stage, moth larvae live in cocoons and fully grow into adult moths. But some of them live under the earth for evolving process. During night, moths fly with the help of moon light. For flying purpose, they use special navigation method called transverse orientation. According to this method, a moth maintains the fixed angle with respect to moon while flying as shown in Figure 1.9. This method is helpful for travelling long distances and also ensures the straight flight towards moon. However, it has been observed that moths fly around the light in spiral manner instead of transverse orientation. This is because transverse orientation is effective only when light source is very far away from the moth. Moths show same behavior around the artificial lights and try to make similar angle with light so that they can travel in straight line. Since the light source is not as far as moon so this straight-line path is changed to spiral path which has been depicted in Figure 1.10. This behavior has been mathematically modeled and the model is called moth –flame optimization (MFO).

1.4 OBJECTIVES

Antenna arrays are the important part of communication system to increase the overall gain of system. Linear array is one of the types of antenna array. Linear array is simple geometrical configuration in which radiating elements are arranged in straight line. To get desired radiation pattern, there is need to optimize the parameters of antenna array such as amplitude, phase and position.

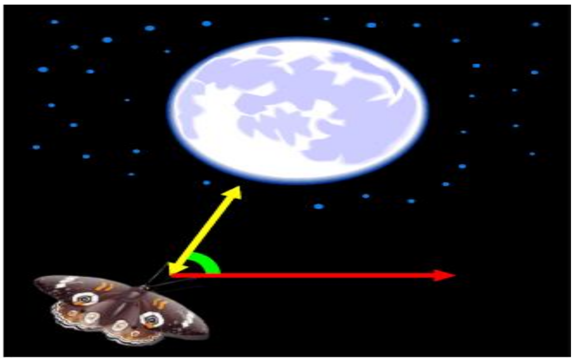


Figure 1.9 Transverse orientation [34]

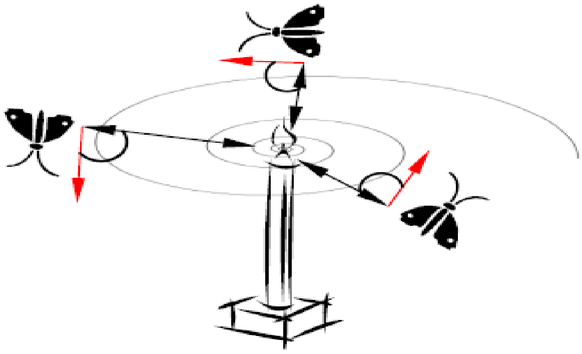


Figure 1.10 Spiral flying path [34]

In this present work, meta-heuristic technique is employed for the synthesis of linear antenna array (LAA) [35] by controlling amplitude only. There are two main objectives of this work one is reduction of side lobe levels and other is null placement in desired direction to avoid interference. Here, enhanced moth flame optimization (E-MFO) is employed for optimizing the amplitude excitation of linear antenna. MFO is latest evolutionary algorithm which has been employed in various real-world problems. But still, there is need of improvement in algorithm to enhance its performance. Main tasks of this thesis are:

a) Study of literature of LAA and MFO
b) Design of an enhanced version of MFO (E-MFO)
c) Implementation of E-MFO on various benchmark functions
d) Implementing E-MFO on linear antenna design problem

1.5 THESIS OUTLINE

This thesis work is divided into six chapters. The chapter description is as follows: In Chapter 2, literature survey is presented which includes various nature inspired algorithms arranged in chronological order. The several modifications and applications of nature inspired algorithms are also discussed. The survey about linear antenna arrays is also given. After that, Chapter 3 describes the basic MFO and its enhanced version. The chapter 4 describes the problem formulation of linear antenna array. Chapter 5 deals with experimental study and its results. It deals with experimental details of algorithm, its performance on various aspects and comparison with other algorithms. It also represents the implementation of proposed algorithm on antenna design problem. Finally, in chapter 6, final remarks are given according to results and future work is presented.

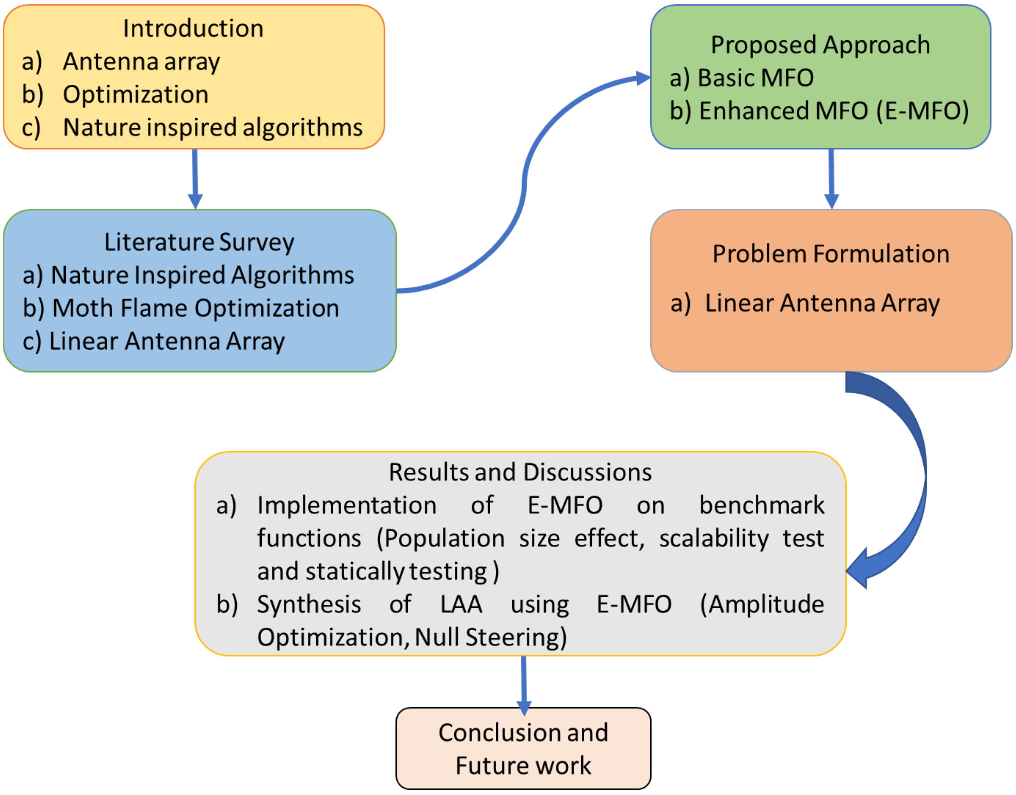


Figure 1.11 Thesis Outline

CHAPTER 2
LITERATURE REVIEW

2.1 LITERATURE REVIEW FOR NATURE INSPIRED ALGORITHMS

The first optimization algorithm was genetic algorithm (GA) [19] which was proposed by Holland in 1992. This evolutionary algorithm is based on Darwin's theory of evolution. The solution of problem is represented by string called chromosome and this chromosome further consists of genes which defines the dimension of problem. The fitness of chromosome is evaluated with help of objective function. Then, the best chromosomes are chosen among whole population and they combined their characteristic through crossover operator to form new offspring. There are different ways to choose best candidates like roulette wheel selection, tournament selection, Boltzmann selection and others. The random chromosome is chosen for mutation process and its one gene is mutated to produce new offspring. The fitness value of new offspring is evaluated and checked whether they improved or not. If their fitness is better from previous population then they are passed to next generation for evolution process. This process continues for number of iterations so that it will achieve global minima value. Population size, number of generations, crossover rate and mutation rate are main parameters of GA. To solve various real-world problems like scheduling problem [36], neural network problem [37], antenna design problem [38] and clustering problem [39], GA has proven its worth by showing better results. Also, there are some drawbacks of GA like it take too much time to give desired results and also lacks in diversity. Afterwards, differential evolution (DE) an evolutionary based algorithm was proposed by Storn and Price in 1997 [20]. DE is population based algorithm comprises of two methods which are mutation and selection. It uses the mutation operation to explore the search space and selection operator to exploit the best solution. It starts with randomly initialized population. A new solution is generated according to mutation and crossover operator and then its fitness is evaluated. The previous solution is compared with new generated solution and get replaced if it has lower fitness. DE also faces some problems like lacking in exploring due to insufficient selection pressure [40]. Hence, to improve its performance various modifications has been proposed like use of chaos theory [41], local search neighborhood [42], hybridization [43-44]. This algorithm has been applied on various tasks like optimal power flow [45], economic dispatch [46], capacitor placement [47]. Particle swarm optimization (PSO) technique was proposed by Kennady and Eberhart. in 1995. It is basically inspired from social behavior of flock of birds. In PSO, particles evolve their behavior and change its position towards destination. All the particles move in specific direction and by communication with each other, they locate the best particle among them.

Then, each particle moves towards best particle with some definite velocity which is dependent on its current position. From past few years, PSO has been successfully applied in wide range of applications [48-50]. This algorithm is popular because it has few parameters to tune and it perform well in various applications. Also, it obtains results in much better and cheaper way. The one of popular examples of SI is ant colony optimization (ACO) [21] which follows the metaphor of social behavior of ants. This algorithm was designed in 2006 by Dorgio. In search of food ants move randomly in search space and leave pheromones trails in their searching path. These trails help other ants to find food because they stop wandering on other paths and start following only trails left by earlier ants. 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 project scheduling problem [51], graph coloring problem [52], image segmentation [53] and routing problem [54]. ACO has proven its worth, by showing efficient and suitable results for benchmark functions and engineering tasks. Artificial bee colony (ABC) [23] is another swarm based technique inspired from foraging behavior of honey bees and proposed by D. Karaboga in 2007. In ABC, there are three types of bees employed bees, scout bees, and onlooker bees. These three group of bees perform different tasks according to rules of SI. ABC algorithm is considered as good algorithm because it has less parameters to tune. There are various modifications in original ABC to make balance between exploration and exploitation phases [55-57].

The above algorithms are very basic algorithms and suffers from many limitations. Therefore, various latest nature inspired algorithms are designed which belongs to either SI or EA like BBO [31], CS [58], FA [28], BA [59], GWO [24] and WOA [60]. Biogeography based optimization [31] is evolutionary algorithm based on distribution of different species on different islands. This algorithm mimics the migration and survival behavior of species. This algorithm has been applied to many optimization problems [61-62]. Cuckoo search (CS) [58] is one such population based algorithm introduced in the recent past. The algorithm uses brood parasitic behavior of cuckoos present in nature to find the solution of problem under test. The algorithm is found to be very efficient and literature regarding its application is expanding rapidly. This algorithm uses a balanced combination of global and local search which is controlled by the switching probability. Various modifications have been proposed to CS algorithms to enhance its exploration and exploitation ability [63-66]. This algorithm showed its effectiveness by solving many non-linear and complex problems [67-70]. Afterwards, firefly algorithm (FA) is introduced which mimics the flashing behavior of fireflies. The details of this algorithm are discussed in chapter one. FA though is a very competitive algorithm but the algorithm lacks in maintaining a proper balance between the exploration and exploitation capabilities making it less efficient. Also, the algorithm sometimes gets stuck in local optima. Thus, to improve its exploration and exploitation, few modifications has been proposed on the basis of step size. Many different methods have been proposed for adaptive step size like implementation of initial and final step size [71], variation according to iterations [72], levy distribution [73], self-adaptive [74] and based on best solution found so far [75]. The concept of opposite based learning [76], chaos theory [77] and memory element [78] has been introduced to improve the performance of FA. FA has been successfully used for solving various real-world task. Ashraf [79] proposed a method for designing circular antenna array using FA. The researchers have applied FA to digital image processing problems like image segmentation. Chenhang [80] combined 2-D otsu method for segmentation with improved FA and the results show the effectiveness of proposed approach in terms of image quality and processing time. Optimal reactive power dispatch (ORPD) [81] is non-convex, non-linear optimization problem comprises of both continuous and discrete decision variable. FA has also been implemented to solve job scheduling problem [82].

Moreover, after CS, Yang proposed one more algorithm named as bat algorithm (BA). It is a metaheuristic algorithm that is based upon the echolocation nature of micro bats developed by Yang in 2010 [59]. Bats possess the fascinating property of avoiding the spectacles and detecting the prey even in presence of complete darkness. Bat emits the loud sound pulse, intercepts the echo and then analyze it. The sound pulse is correlated by their hunting strategy depending on the prey. To overcome the problem of BA, many modifications has been proposed [83-86]. BA is also hybridized with other algorithms in order to enhance its performance. There are so many applications in which BA was used as optimization tool [87-89].

In last three years, Mirjalli *et al.* has been proposed many optimization algorithms like GWO, WOA etc. GWO algorithm [24] 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. The main phases of group hunting of grey wolf are encircling, hunting and attacking the prey. This algorithm suffers from exploration problem so several modifications has been employed like adaption of parameters [90], re-initialization of worst particle [91], hybridization [92] and binary GWO [93]. Along with these modification of GWO, there are many authors who have adopted the basic GWO to solve real world problems like power system [94], fuzzy controller [95] and neural networks [96]. Furthermore, whale optimization algorithm (WOA) [60] is a recently developed swarm intelligence based algorithm which is inspired from the social behavior of humpback whale. This algorithm mimics the bubble net hunting strategy of whales and has been applied to optimization problems [97-100].

2.2 LITERATURE SURVEY FOR MOTH FLAME OPTIMIZATION

MFO is latest population based algorithm proposed by Seydali Mirjalli in 2015. It is inspired from moth's special navigation methods. Due to its good performance, it has been used for solving many engineering design problems. In [101], authors proposed an improved version of MFO named as Levy flights based MFO (LMFO). To improve the diversity of algorithm, Levy function is used to update the position at the end of iteration. This will lead to rise in convergence speed due to its higher capability of avoiding local minima. Combination of MFO with Levy flights makes the adequate balance between exploration and exploitation. The proposed algorithm is tested on 19 benchmark functions. According to numerical analysis, LMFO outperforms basic MFO. So, this approach helps MFO in obtaining global minima and help in avoiding local minima. Furthermore, this LMFO algorithm is also applied on two real-world problems namely beam welding design and speed reducer design. Just like benchmark functions, LMFO algorithm performs effectively for these engineering problems also. The effect of spirals on performance of MFO algorithm has been discussed in [102]. In original paper logarithmic spiral is used, however in this article two different spirals are used, one is reciprocal spiral and another is Archimedes' spiral. The new version of MFO is applied on feature selection problem to predict the terrorist groups. Data taken from Global Terrorism Database is preprocessed with help of MFO and then its best feature set is chosen. The results are conducted on two classifier algorithms such as Random Forests (RF) ensemble and K-Nearest Neighbor algorithm. The results shown in paper conclude that new version of MFO gives competitive results to original MFO. Also, this algorithm provides more stability, validity and robustness. The new modification is proposed in [103] which helps to improve the exploration ability of MFO. Chaos theory and crossover processes are introduced in this algorithm which increases the randomness or diversity. Chaotic systems have properties like randomness, certainty and ergodicity which help the solution to jump out of local minima. The linear chaotic map is employed and is controlled by random parameter P whose value varies from 0 to 0.5. The updating equation of basic MFO is converted into:

$$M_i = S(M_i, F_i) + c * (2 * X_i - 1) \tag{2.1}$$

Here c specifies the constant whose value is equal to 1.75 and X_i is chaotic vector. Furthermore, this algorithm employs crossover process with crossover probability equal to 0.5. Moreover, adaptive flame number mechanism is also changed using sinusoidal calculation; it will help algorithm to make proper balance between exploration and exploitation. This modified MFO algorithm is used to solve satellite image thresholding problems. Dataset of 5 satellite images are taken to evaluate the performance of modified MFO. Clearly, it is shown in results that proposed approach provides better results in comparison to others. The MFO algorithm has been applied to various applications like it has been successfully applied in field of neural networks [104]. The main aim of applying this algorithm is to train the multilayer perceptron (MLP) with optimal values of weights and biases so that minimum error and high classification rate can be obtained. Here MFO is used to find the best possible values of weights which are used with training samples to train MLP. The mean square error is taken as key index to evaluate the performance. This problem is verified on five different datasets like XOR, heart, balloon, iris and breast cancer. Experimental analysis shows that MFO algorithm is very effective in training MLP and avoiding local minima problems. Moreover, it gives better results in comparison to other algorithms due to its high exploring and exploiting capabilities. MFO has proven its worth in electronics field by designing mathematical model for solar cells [105]. In this work, authors emphasized on the need of extracting or estimating the modal parameters by using meta-heuristic algorithms. Here MFO algorithm is used to achieve minimum root mean square error (RMSE) and mean bias error (MBS). It is clear from the results that MFO algorithm efficiently calculates the model parameter while optimizing the RMSE and MBS. In comparison to other algorithms MFO has high convergence rate which means it reaches the global minima in less number of iterations. Optimal power flow is challenging task in an electrical domain [106]. An interconnected power system is taken with various constraints to solve the optimal power flow problem with help of MFO. The objective of this problem is to minimize the fuel cost, rate of emission and improvement in voltage profile. MFO algorithm has been used to test the 59-bus test system which has 20 control variables. The results show the effectiveness and robustness of this algorithm in comparison to other state in art algorithms.

The modified version of MFO has been applied in field of image processing for feature extracting. According to this work [107], MFO algorithm is hybridized with rough set theory to detect tomato disease. This combination of MFO and rough set is used for feature selection phase. The fitness function of MFO depends on rough sets. The idea behind using rough set is to generate all possible feature reductions and then choose one with minimum cardinality. This proposed algorithm exploited the high exploration capability of MFO and high performance of rough sets for feature selection of real dataset of tomato disease. To verify the results, proposed algorithm is tested on six benchmark datasets and results showed that this algorithm is efficient in comparison to others in terms of accuracy, feature size and execution time.

The electrical problems are one of real-world and complex problems. In article [108] problem is to deal with elimination and minimization of harmonic distortions in multilevel inverters by using latest developed meta-heuristic algorithm MFO. To verify the effectiveness of MFO algorithm, this is tested on 7 and 11 level multilevel inverter. MFO is used to calculate the switching angles at different values of modulation index. The results showed that MFO is successful in solving harmonic problem very efficiently. In [109], MFO is employed to solve the optimal power flow problem (OPF). OPF is one of the necessities of any electrical network. Higher convergence rate and capability of finding good solution make MFO superior than other algorithms. In this work, MFO is implemented on IEEE 30 bus test system. The aim of this work is to reduce the fuel cost, minimize active and reactive power loss. The results show the worth of this algorithm with very good results over other nature inspired algorithms. MFO has also been employed for compter network problems like to find the shortest path between source and destination even if that shortest path may contain attacker node [110]. In other words, we can say that it deals with open shortest path first problem (OSPF). This problem is optimized by using nature inspired algorithms like MFO. The optimized OSPF can be used in huge networks for secure routing. Delay and consumption of energy are key parameters to evaluate the performance of optimized OSPF. For simulation, a large network area is considered with 9 blocks and results shows the wide difference from the original one.

2.3 LITERATURE REVIEW FOR ANTENNA ARRAY

The concept of antenna array is related to analogy of rain and buckets. Large buckets collect large amount of rain, just like that, in communication system antenna with large aperture receives large amount of electromagnetic energy. In this analogy buckets are described as antenna and rain is as energy of signals. In order to collect large energy, large antenna is replaced by many small antennas corresponds to replacing large buckets with small ones to collect maximum amount of rain. The advantages of using large bucket is to collect large amount of water in one direction whereas small buckets are easy to handle and to move from one place to another to collect water. This analogy lacks in terms of phase because rain does not have phase but electromagnetic waves does.

The first antenna array was designed 100 years ago [111]. Brown used two vertical antennas, separated by half a wavelength, to increase the overall gain of system [112]. These both antennas were working out of phase to each other. He concluded from the experiment that directivity was maximum in the plane of antenna. Afterwards, De Forest [113] and Marconi

[114] performed several experiments on multiple antenna to increase the overall gain of system. The early arrays were physically large and its main goal was to find the direction of signal arriving. The antenna arrays were first used in military applications to detect the enemy aircrafts [115]. In mid 1950s, scientist worked on suppressing sidelobes by varying the antenna parameters for linear and circular antenna arrays. They also created a model of antenna array that scans in azimuth and elevation angles. Afterwards, in 1960s a new array was designed in which large energy source is replaced by vacuum tube transistors [116]. But this idea could not work out. Further, microstrip patch antenna array was designed using small patches to adapt the curved shape instead of planer surface. The research is still going on microstrip antenna array. Antenna arrays are widely used in field of wireless communication, radar, mobile communication etc. This antenna technology is widely used in latest system known as MIMO system (Multi Input Multi Output). Different applications of antenna array are to broadcast, missile guidance, fire radar aircraft, weather research.

2.3.1 Linear antenna array

Linear array is array in which all elements are arranged in single line with same excitation level that's why its geometry is simple and easy to synthesize. LAA is complex and non-linear problem and hence many optimization techniques have been employed to synthesis LAA. Conventional optimization techniques are based on intial guess and if intial guess goes wrong then solution will be struck into local minima. Due to this, nature inspired algorithms have become a popular choice for the designing of LAA. Nature inspire algorithms have capability to find global minima without trapping into local minima.

Many meta-heuristic algorithms have been employed for synthesis of LAA. Merad *et al*. employed Tabu search (TS) for designing of linear array antenna (LAA) [117]. GA was also used for synthesis of LAA [118]. In this work, author used complex number to represent the array excitation and decimal linear crossover instead of binary crossover. To check the effectiveness of proposed approach it was tested on linear and circular broadside array with thirty isotropic antennas. In case of linear array SLL achieved was -36.02 dB and for circular array lowest SLL obtained was -29.16 dB. Further, in [119] authors discussed that PSO was better option for synthesis of linear arrays in comparison to GA and SA because PSO is easy to implement and provide better results. PSO was employed to adjust the spacing between elements to suppress SLL. Three examples were considered to verify the proposed approach which were 10, 28 and 32 element array antennas. It was observed from the results that PSO successfully optimize the locations of array elements. Comprehensive learning PSO is one of the variants of PSO. This is basically advance type of PSO which was used for SLL, null placement and for beam-width control [120].

An evolutionary based DE algorithm [121] was also applied to reduce SLL and interference of nulls with main lobe. In this work, DE was applied for the design and synthesis of uniformly excited array with equal and unequal phases. The impact of angle resolution on performance of LAA is also evaluated. A 32-element array was taken and results showed the improved array efficiency and reduced SLL. Multi objective DE was also used for reducing SLL and null placement [122]. However, it provides better trade-off between SLL and nulls in comparison to single objective DE. The comparison of self-adaptive differential algorithm (SADE) and taguchi's method (TM) for designing of LAA, was evaluated in [123]. Other variant of DE named as fitness adaptive DE (FiADE) [124] has also been employed for designing LAA. BBO [125] has been employed for designing LAA. Its main objective was to minimize the maximum sidelobe in neighbor and placement of nulls in desired direction. The three different examples were considered like 10, 20 and 24 element arrays. The results obtained from BBO were better in comparison to TS and PSO algorithms. The swarm based technique named ACO [126] has also been implemented for designing of non-linear LAA. The aim of the proposed approach was to find the element locations for desired radiation pattern. The five different objective functions were taken to synthesis the 10, 28 and 32 element arrays. The objective functions were based on SLL suppression, narrow beam-width and nulls in desired direction. The convergence curve and radiation patterns were drawn for each case. Clonal selection algorithm [127] is latest algorithm and provide comparable results with other nature inspired algorithms hence employed for synthesis of LAA. A initial chebyshev pattern was considered for 20 element arrays. The extensive study about placing nulls, in single, double and multiple directions, was also done. The maximum SLL and null depth length (NDL) were the performance index. These both factors were in terms of array factors and measured in decibels. Similar to clonal selection algorithm, an immune algorithm [128] has also been implemented for designing LAA by optimizing the amplitude excitation. For validation of proposed approach, linear array of 20 element was considered with uniform spacing. Twelve different examples were taken which describes the suppression of SLL and imposing nulls in different directions. Guney et al. proposed four different optimization algorithms for side lobe reduction and null placement in LAA by controlling amplitude only. These algorithms are bees algorithm (BA) [129], bacterial forging algorithm (BFA) [130], plant growth simulation algorithm (PGSA) [131] and modified touring ACO [132]. The performance analysis of these algorithms was conducted on 20 element antenna array with uniform spacing and 30-db Chebyshev pattern as initial radiation pattern.

An evolutionary based flower pollination algorithm has also been applied for designing LAA and controlling nulls at various positions [133]. The objectives were achieved by controlling amplitude only and phase only. The results obtained from FPA is better than other popular algorithms as given in above literature. The radiation patterns were drawn at different null positions for both of cases. Guney *et al.* proposed backtracking search algorithm (BSA) [134] to synthesis the linear antenna array with pre-defined nulls at interference direction. BSA is simple optimization technique because it has only one parameter to tune. The synthesis of LAA is done by controlling amplitude only, phase only and position only. Eleven different examples have been performed with chebyshev and uniform array as initial pattern. Harmony search algorithm (HSA) [135] has also been employed to perform the same task as above in BSA. The results obtained by HSA was better than other several optimization algorithms. The ant lion optimization (ALO) [136] has been presented for designing LAA with different cases like for 10, 12 and 24 elements. The experimental analysis shows that proposed approach performed better in comparison to other algorithms. GWO is latest algorithm and has been applied to many complex problems Hence, this algorithm has also been used for synthesis of LAA [137]. Many other latest nature inspired algorithms have been applied for the same task as modified spider monkey optimization (MSMO) [138], seeker optimization algorithm (SOA) [139].

The disadvantages of classical techniques are efficiently overcome in nature inspired algorithms. Various nature inspired algorithms have been discussed in literature survey but not all the algorithms are good for all the real-world applications so there is still need to discover new algorithms. Moth flame optimization is latest evolutionary algorithm with good convergence speed and has ability to reach at global optima. However, the literature survey shows that, this algorithm lacks in exploration ability. So, there is need to improve the drawbacks of this algorithm. Antenna arrays are the important part of communication system. To reduce the side lobe levels, the antenna parameters (amplitude, phase and positions) need to be optimize. Many classical as well as nature inspired techniques has been employed for the synthesis of linear antenna array. To make the design of LAA more robust, latest optimization techniques should be implemented.

CHAPTER 3

ENHANCED MOTH FLAME OPTIMIZATION ALGORITHM

In this chapter, the details of MFO and its enhanced version, is given. Firstly, we discussed about the basic MFO and then its modified version.

3.1 BASIC MFO

MFO is latest evolutionary algorithm which mimics the special behavior of moths called transverse orientation. The moths follow the spiral path to reach at the flames. The description of MFO given below.

3.1.1 Mathematical model

The key components of MFO algorithm is given below:

3.1.1.1 Moth component: The MFO algorithm consists of three parts which is used to find global minima for any given problem. It can be defined as follows:

$$MFO = (I, P, T) \tag{3.1}$$

The I component defines the initialization of randomly distributed moths in search space, P function is the movement of moths towards the flame and T function returns true when the termination condition is satisfied and false otherwise.

In MFO algorithm number of solutions represents the moths (n) and dimension (d) of problem defines the position of moth in search space. As we know, MFO is population based algorithm so moths can be represented in form of matrix.

$$M = \begin{bmatrix} m_{1,1} m_{1,2} m_{1,d} \\ m_{2,1} m_{2,2} m_{2,d} \\ . \\ . \\ . \\ m_{n,1} m_{n,2} m_{n,d} \end{bmatrix}$$

The fitness value of each solution is stored in array as shown below:

$$OM = \begin{bmatrix} OM_1 \\ OM_2 \\ . \\ . \\ . \\ OM_n \end{bmatrix}$$

Here n represents the number of moths. Basically, moth's position vector is passed to fitness function which returns the output called fitness value.

3.1.1.2 Flame component: To describe the flames, *F* matrix is considered same as moth matrix and its fitness value *OF* is also calculated in similar way. The moths and flames are both solutions and the only difference lies in their updation mechanism. Moreover, we can say that moths are the search agents which explore the search space and the flames are best solution obtained so far. Flames can be considered as flags which are dropped by moths while searching.

3.1.1.3 Transverse orientation of moths: To simulate the behavior of moth in mathematical model, the position of moth is updated around flame using equation (4.2)

$$M_i = S(M_i, F_j) \tag{3.2}$$

Here M_i represents the i^{th} moth, F_j represents the j^{th} flame and S is spiral function which defines how will moth updates its position "around" the flame not necessarily in the space between them. The logarithmic spiral is chosen for this paper. The chosen spiral must satisfy the below conditions:

1. The starting point of spiral must be a moth's position.
2. Spiral should end at flame.
3. Fluctuation of range of spiral should be within given search space.

The equation used for updating mechanism of MFO algorithm is given below:

$$S(M_i, F_j) = D_i . e^{bt} . Cos(2\pi t) + F_j \tag{3.3}$$

where $D_i$indicates the distance between the i^{th}moth and j^{th}flame. It can be calculated as:

$$D_i = |F_j - M_i| \tag{3.4}$$

Here *b* is any constant number that defines the shape of spiral and *t* is random number between 1 and -1. The *t* parameter decides the step size of moth's movement towards flame. In other words, this parameter describes that how much moth's next position will be close to flame. It is shown in Figure 3.1 that when *t=1* moth is far from the flame whereas *t=-1* indicates the closet position of moth around the flame.

4.1.2 Some points about MFO

1. The *t* parameter is key index, by changing its value moth can converge to any position around the flame. As the value of *t* decreases, distance between moth and flame also decreases. Moreover, for proper exploitation the parameter t is taken in the range of r to 1 where *r* is linearly decreasing function from -1 to -2 over course of iterations. This change will lead to more precise exploitation of corresponding flames by moths.

2. To avoid local minima, each moth is allowed to update its position towards only one flame. In every iteration, flames are updated and sorted based on fitness value. Afterwards, moths update its position with respect to their corresponding flames. If all moths are attracted toward only one flame, it will lead to stagnation. However, moving towards different flames causes high exploration. First moth will update its position according to first flame and last moth will update its position with respect to worst flame. The Figure 3.2 shows the connection between the moths and flames. To ensure the best results, best solutions obtained so far are considered as a flame and moths move towards the flame in hyper sphere path.

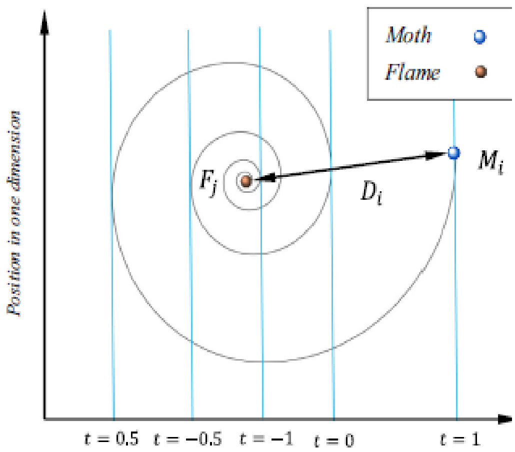


Figure 3.1 Logarithmic spiral [34]

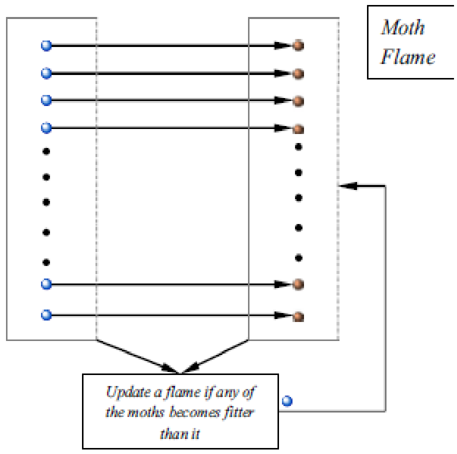


Figure 3.2 Connection between moth and flame [34]

3. Best solutions in each iteration get changed and so do the flames. So, moths are required to update their position towards newly updated flame. It will cause the moths to move around different flames. This rapid movement of moths in the search space promotes exploration.
4. The numbers of flames also keep on decreasing with increase in iterations. In initial stages the moths update its position around assigned flame and start moving towards the best flame in the final stages. This adaptive mechanism makes balance between the exploration and exploitation. The following relation is used to generalize this statement:

$$flameno = round\left((N - L) * \frac{N - 1}{T}\right) \tag{3.5}$$

where N is number of flames, L is current iteration and T is maximum iterations.

3.1.3 Drawbacks of MFO

It is clear from the presented literature survey in chapter 2 that MFO has been effectively applied on various optimization problems. Moreover, this is also true that original MFO suffers from local stagnation problem because of less diversity among search agents. The articles in literature present that various methods have been applied to improve its exploration ability such as chaos theory, crossover and levy distribution function. But, their work lacks in explanation and experimentation analysis. So, these drawbacks lead to the new enhanced version of MFO which is discussed in next section.

3.2 ENHANCED MOTH FLAME OPTIMIZATION

For any generalized optimization algorithm, exploration and exploitation are the two main aspects. Exploration refers to the global search and corresponds to whole of the search space whereas exploitation is defined as local search and corresponds to small areas inside the search space. Both these phenomena help the algorithm in avoiding local minima stagnation problem (from exploitation), provide better convergence and diversity in the solutions (from exploration). Another important feature is the balance between these two phenomena. Any algorithm which is able to achieve these three properties can be considered in the family of state-of-art algorithms. MFO is a recently introduced algorithm and it can be inferred from the literature that it suffers from the problem of poor global searching capability. Here it can be said that exploration part of MFO need to be enhanced to make it fit for high end problems. Also, improving only, the exploration and not the exploitation may not help the algorithm in maintaining a balance between the two. So, in present work, a new enhanced version of MFO namely E-MFO has been proposed by improving the local as well as global search capability of the original MFO algorithm. The details of proposed modifications are given below:

3.2.1 Division of iterations

Since exploration and exploitation are the two main aspects of any optimization algorithm and there should be proper balance between them to achieve the global solution. During initial iterations, it is required that search agents must move inside whole search area and try to explore it instead of getting trapped in some local minima. Hence search agents gather the information about best solutions during exploration process and towards the later stages, exploit that information and try to reach global minima. Here it should be noted that too much exploration may cause more randomness and degrade the performance of any algorithm, similar to this too much exploitation may lead to loss of diversity causing local optima stagnation. A large number of articles have been published in this context and it is still a matter of concern and more research is to be carried out to know how much exploration and exploitation is to be carried out. In present work, we have considered that 50% exploration and 50% exploitation as the basis and this has been achieved by dividing the total number of iterations into two equal parts. During first half, algorithm will tend to perform more exploration and in later half it will emphasizes on exploitation. This 50-50 balance is ensured by utilization of random attraction model and influence of best flame as discussed in following subsections. Random attraction model is used in first half iterations to improve the exploration capability and in other half iterations, influence of best flame is taken in order to enhance exploitation capability.

3.2.2 Exponential step size

The parameter *t* describes the next position of moth with respect to flame. In basic MFO *t* varies from r to 1 where *r* is linearly decreasing function from -1 to -2 proportional to iterations. According to Figure 4.3 when value of *t* is high, it means the position of moth is far from the flame so moth have to take large steps (exploration) to reach at the destination while when the value of *t* is less, step size is less (exploitation) and it will reach the destination in limited time. A proper *t*-value should be there to ensure balanced exploration and exploitation. So, in proposed approach exponential function helps to increase the exploration rate with respect to iterations in the initial stages and also exploitation in the later stages. In other words, using exponential function helps to maintain a proper balance between exploration and exploitation as shown in Figure 3.3. The value of *r* will change slowly instead of rapidly. The equation used for this is as follows:

$$r = \frac{-2}{1 + e^{\left(-6*\frac{L}{T}\right)}} \tag{3.6}$$

Where L is current iteration and T represents the maximum iterations. Here the value of *r* varies from -1 to -2.

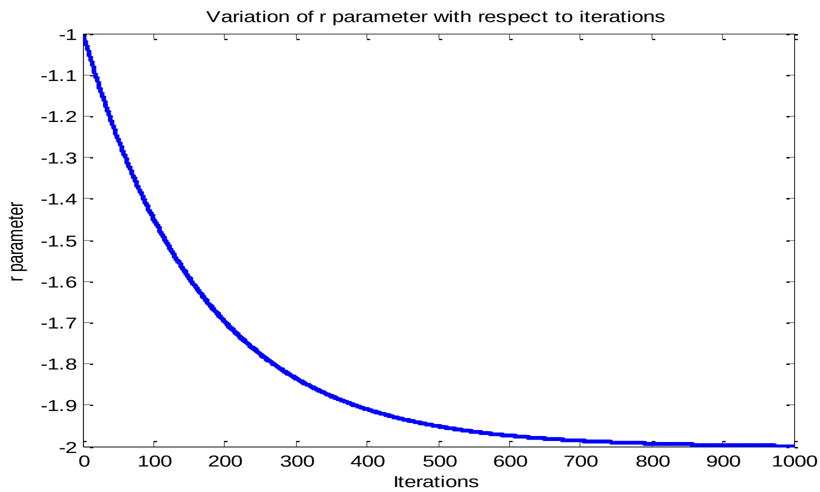


Figure 3.3 Exponential curve for step size

3.2.3 Random attraction model

According to MFO algorithm, during initial phase moth updates its position towards corresponding flame and in later phase they tend to move around only best flame. Further, basic MFO describes that the sequence of flames is assigned to the moths and each moth updates its position corresponding to that flame. The first moth moves with respect to best moth and last moth moves according to worst flame. Sometimes, following the particular sequence reduces the randomness in nature. In order to enhance the randomness and diversity in initial iterations, new modification is proposed which defines the random model of MFO algorithm given in Figure 4.4. A random model describes that one random flame is chosen by moth for its position updating. Now, instead of moving towards assigned flame, moth will tend to move around one random flame. Although moth will update its position according to random flame but in final iterations moth will move towards only best flame just like basic MFO. This randomness helps to increase the exploration capability of MFO algorithm. The Figure 3.4 shows the random attraction model.

$$Random\ Flame = 1 + fix(rand * N) \tag{3.7}$$

The random flame can be chosen by this equation where N specifies the number of flames in each iteration. So, the equation will be:

$$(M_i, F_j) = D_i . e^{bt} . Cos(2\pi t) + F_{Random\ Flame} \tag{3.8}$$

This modification is only applied for first half iterations so that requirement of more exploration can be fulfilled.

4.2.4 Influence of best flame

The basic paper of MFO states that each moth updates its position with respect to its corresponding flame. There is sequence of flames which are assigned to moths in each iteration. First moth will move for best flame and last will move towards worst flame. This is not the actual behavior of moths in environment. However, assigning each moth a flame promotes the exploration but at the same time it degrades the exploitation capability. In proposed approach, the influence of best flame is taken for all the moths. Here all the moths will update their position according to their corresponding flame and the best flame. The moths will move towards average position of best and its corresponding flame. Since the moths are moving toward the best flame, so this modification promotes the exploitation and helps in improving the convergence speed. The effect of best flame is taken only during second half of iterations where more exploitation is the requirement.

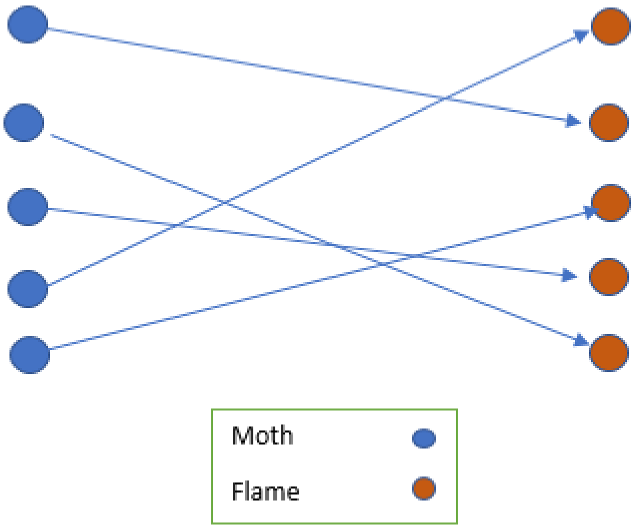


Figure 3.4 Random attraction between moths and flames

The new updated equation will become:

$$S(M_i, F_j) = D_i . e^{bt} . Cos(2\pi t) + \frac{a * F_j + (1-a) * bestflame}{2} \quad (3.9)$$

where

$$a = 0.6 - 0.4 * \frac{L}{T} \quad (3.10)$$

Here a parameter is linearly decreasing function and is used as controlling factor which controls the effect of best and corresponding flame. The value of a varies from 0.4 to 0.2 with respect to iterations. In initial stage, the value of a parameter is kept high which signifies that the effect of corresponding flame is more so that moth can explore the search space efficiently. After that in later stages, the value of a is decreased which helps the algorithm to move towards best flame and improve the convergence speed. The pseudo code and flow diagram for proposed approach is given in Figure 3.5 and 3.6.

```
Initialize the position of moths
While (iteration<= Maximum iteration)
    if (iteration <=Maximum iteration/2)
        Update flame Number using equation (3.5)
        OM=Fitness Function (M)
        If iteration==1
            F=sort (M)
            OF=sort (OM)
        Else
            F=sort (M_t − 1, M_t)
            OF=sort (M_{t−1}, M_t)
        End
        For i =1: n
            For j=1: d
                Change r and t
                Calculate D using (3.4)
                Evaluate M (i, j) using (3.8) and (3.1)
            End
        End
else
    Update flame Number using equation (3.5)
    OM=Fitness Function (M)
    If iteration==1
        F=sort (M)
        OF=sort (OM)
    Else
        F=sort (M_t − 1, M_t)
        OF=sort (M_{t−1}, M_t)
    End
    For i =1: n
        For j=1: d
            Change r and t
            Evaluate D using (3.4)
            Calculate M (i, j) using (3.9) and (3.1)
        end
    end
    end
end
```

Figure 3.5 Pseudo code for E-MFO

The below given flowchart is divided into four parts. First is initialization part shown by pink color and second part describes about the boundaries conditions. If moth population lies outside lower and upper bounds then it should firstly bring back into range by applying boundaries conditions. This part is represented by yellow color. Afterwards, flames updation part is shown by green color. In this phase moths are sorted according to fitness and sorted population of moths considered as flame population. In next phase, moth positions are updated according to number of iterations and this phase is shown by blue color. The last terminating phase is represented with orange color.

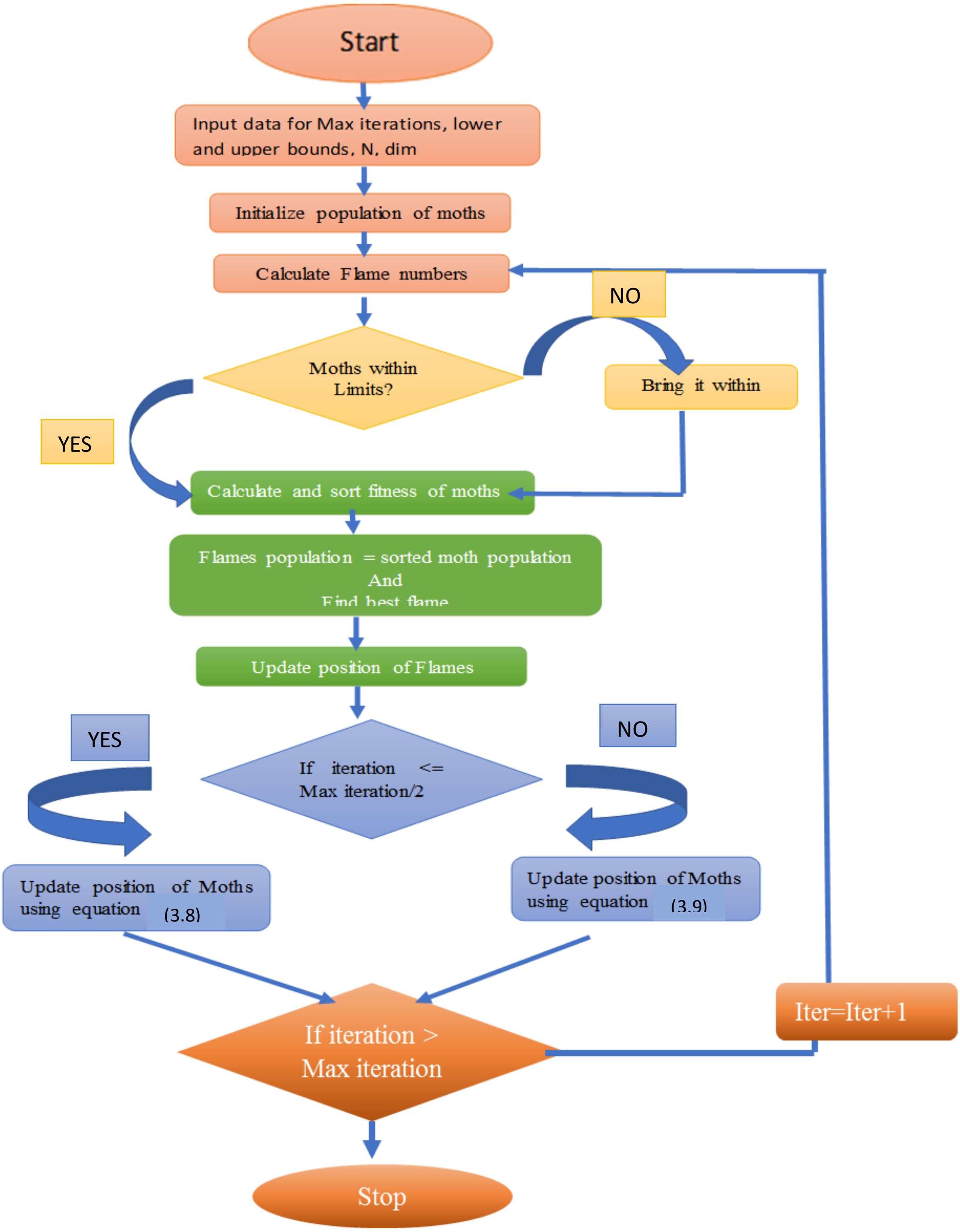


Figure 3.6 Flowchart for proposed approach

CHAPTER 4

PROBLEM FORMULATION

In this present work, the synthesis of linear antenna array is carried out using meta-heuristic technique named as enhanced moth flame optimization. Linear antenna array is chosen because it is easy to handle and has simple geometry. The main aim of this work is to reduce the side lobe levels and null steering. To meet the desired goals, there is need to optimize the antenna parameters. The amplitude of current excitation is varied while keeping other parameters constant. For optimization, nature inspired algorithm E-MFO provides optimal set of current excitations. This chapter describes the linear antenna array as an optimization problem and defines the fitness function in terms of array factor.

4.1 Linear antenna array

A linear array is assembly of N radiating elements placed in straight line. A Figure 4.1 depicts the arrangement of 2N elements along x-axis. The figure shows that elements are symmetric about origin means same number of elements lie on both side of origin. This symmetric arrangement is widely used in wireless communication because of less complex structure and symmetric radiating pattern. Here, only N elements are needed to be optimized because rest of N elements are just symmetric to that.

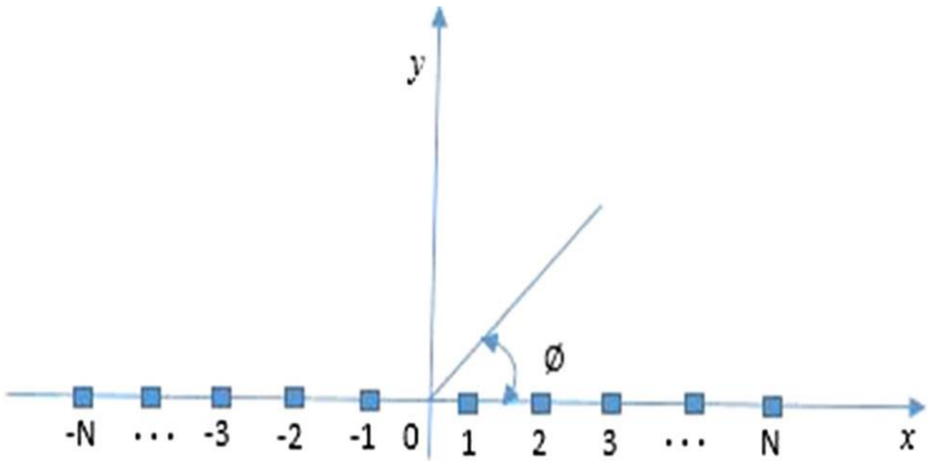


Figure 4.1 Linear array antenna [133]

The designing of LAA depends on various parameters like array factor, amplitude and phase excitation, wave number and its position. The array factor is given as:

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

The parameters I_n describes the amplitude excitation, ψ_n describes phase and x_n is position of n^{th} element. Also $\emptyset$ represents the azimuthal angle and k is wave number whose value is derived from 2 π / λ.

In this present work, synthesis of uniform LAA is performed. Uniform LAA means the inter-element spacing of radiating elements is constant and equal to λ/2. Only amplitude excitation is varied to reduce the SLL. The amplitude ranges between [0 1]. The phase excitation ψ_n is taken as zero. So, the simplified version of array factor is given as:

$$\mathrm{AF}(\emptyset) = 2\sum_{n=1}^{N} I_n \cos[(n - 0.5)\pi \cos(\emptyset)] \tag{4.2}$$

$$\mathrm{AF}(\emptyset) = 2\sum_{n=1}^{N} I_n \cos(k x_n \cos(\emptyset)) \tag{4.3}$$

The objective function for this aim is:

$$Fitness\ Function = k_1 f_{SL}(\emptyset) + k_2 f_{NS}(\emptyset) \tag{4.4}$$

This fitness function describes the two main goals one is side lobe reduction and other is null steering by controlling amplitude only. Here, k_1 and k_2 are weighting factors that decides the dominance of one particular aim whether it is sidelobe reduction or null replacement. The factors $f_{SL}(\emptyset)$ and $f_{NS}(\emptyset)$ are defined as:

$$f_{SL}(\emptyset) = \max[AF(\emptyset)] \tag{4.5}$$

$$f_{NS}(\emptyset) = \left[max_{k=1.....K}\ (AF(\emptyset_k^{null}))\right] \tag{4.6}$$

Here SL is required region for side lobes and $\emptyset_k^{null}$ is k^{th} null direction. AF defines the array factor in terms of decibels.

CHATER 5

RESULT AND DISCUSSION

In this chapter, we analyze the effectiveness of enhanced moth flame optimization using benchmark functions and antenna design problem. First section includes the description of test functions and its implementation on proposed approach. The details of function include its dimension, global optimum value and its equation. Also, the results of proposed approach are compared to basic algorithm as well as other meta-heuristic algorithms. Second section comprises of synthesis of linear antenna array with different cases. For performance evaluation results are taken at Windows7 having properties like x32-bit operating system, 4 GB RAM, Intel core i3-4200U processor and CPU@ 1.6GHZ. The MATLAB version used is R2007b.

5.1 TEST FUNCTIONS

Test functions are artificial problems which are used to evaluate and validate the optimization algorithm under constraints. These functions help to test the reliability, efficiency and effectiveness of any optimization algorithm. In literature [140], dozens of functions are provided with different properties. Mainly, these benchmark functions include unimodal functions, multimodal functions and multimodal functions with fixed dimension. Unimodal functions are those which have only one global minima on the other hand multimodal functions has number of local minima. Here it should be noted that unimodal functions help to check the exploitative ability of an algorithm whereas multimodal functions help in keeping a check on the explorative capabilities of an algorithm. In this case, nineteen functions are taken to evaluate the performs whose details are given below in Tables 5.1, 5.2 and 5.3:

Equation	Global minima	Dim (*d*)	Range
$f_1(x) = \sum_{i=1}^{D} x_i^2$	0	30	[-100 100]
$f_2(x)=\sum_{i=1}^{n} \lvert x_i \rvert+\prod_{i=1}^{n} \lvert x_i \rvert$	0	30	[−10,10]
$f_3(x)=\sum_{i=1}^{n}\left(\sum_{j-1}^{i} x_j\right)^2$	0	30	[−100, 100]
$f_4(x)= max_i\{\lvert x_i \rvert, 1 \leq i \leq n\}$	0	30	[−100, 100]
$f_5(x) = \sum_{i=1}^{N-1}(100\ (x_{i+1} - x_i)^2 + (x_i - 1)^2$	0	30	[-100 100]
$f_6(x)=\sum_{i=1}^{n} ix_i^4$+random [0,1)	0	30	[−1.28,1.28]
$f_7(x) = \left[\frac{1}{n-1}\sqrt{s_i}.(\sin\left(50.0s_i^{\frac{1}{5}}\right)+1)\right]^2 = \sqrt{x_i^2 + x_{i+1}^2}$	0	30	[-100 100]
$f_8(x) = x_0^2 + 10000\sum_{i=1}^{D} x_i^2$	0	30	[-10 10]

$f_9(x) = \sum_{i=1}^{D} (10^6)^{\frac{i-1}{D-1}} x_i^2$	0	50	[-100 100]
$f_{10}(x) = \sum_{i=1}^{D} x_i^{i+1}$	0	30	[-1 1]
$f_{11}(x) = \sum_{i=1}^{D} i x_i^2$	0	30	[-10 10]
$f_{12}(x) = \sum_{i=1}^{d} [(x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2$ $+(x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^2]$	0	30	[-4 5]

Table 5.1: Uni-modal functions [140]

Equation	**Global minima**	**Dim (*d*)**	**Range**
$f_{13}(x) = 10D + \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i)]$	0	30	[-5.12 5.12]
f_{14}(x)=$-20 \exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right) -$ $\exp\left(\frac{1}{n}\sum_{i=1}^{n} \cos(2\pi x_i)\right) + 20 + e$	0	30	[-100 100]
$f_{15}(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N} \cos(\frac{x_i}{\sqrt{i}}) + 1$	0	30	[-600 600]
$f_{16}(x) = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5 i x_i)^2 + (\sum_{i=1}^{n} 0.5 i x_i)^4$	0	30	[-5 10]
$f_{17}(x) = sin^2(\pi w_1) + \sum_{i=1}^{d-1} (w_i - 1)^2 [1 + 10 sin^2(\pi w_i + 1)] + (w_d - 1)^2 [1 + 10 sin^2(2\pi w_d)]$	0	30	[-10 10]

Table 5.2: Multimodal functions [140]

Equation	**Global minima**	**Dim (*d*)**	**Range**
f_{18} (x)= $\sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	0.00030	4	[−5,5]
f_{19} (x)= $4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	−1.0316	2	[−5,5]
f_{20}(x)=$[1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] * [30 + (2x_1 - 3x_2)^2 * (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3	2	[−2,2]

Table 5.3: Multi modal functions with fixed dimensions [140]

5.2 PERFORMANCE OF E-MFO ON BENCHMARK FUNCTIONS

The performance of proposed approach verified on nineteen functions whose details is given in above tables. The results of E-MFO are compared with FA, FPA, BA, BFP, DE. The various parameters of these algorithms are given below in Table 5.4. All the algorithms are run for 50 times and 500 iterations. The effect of population and dimension is also verified on proposed approach.

Algorithm	Parameter settings
BA	Loudness (A)=0.5, Pulse Rate(r)=0.5
FPA	Switch Probability (p)=0.8
DE	Control Parameter(F)=1.5, Crossover rate (CR)=0.8
BFP	Loudness(A)=0.5, Pulse rate(r)=0.5, Switch Probability (p)=0.8
FA	Randomization parameter(α)=0.25, Attractiveness (β_o)=1, Light absorption coffieicent (γ)=1

Table 5.4: Parameter settings for various algorithms

5.2.1 Effect of population

To investigate the effect of population on algorithm, all the algorithms including E-MFO, MFO, BA, BFP, DE, FA and FPA runs for 30, 60 and 100 population size. The analysis of their results is shown in Table 5.5, 5.6 and 5.7 for population size 30, 60 and 100 respectively.

Case 1: Population size 30

In first case, population size and dimension both are considered to be equal to 30. From the Table 5.5, it is clear that for function f_1, E-MFO outperforms other algorithms in terms of best, worst, average and standard deviation. A function f_2 provides better results in case of MFO but only in terms of best solution, rest of solutions like worst, average and standard deviation are good in E-MFO. In case of f_3 ,f_4 , f_5 and f_6 E-MFO outperforms other algorithms. Although, best value in case of f_7 is same for algorithms like MFO, E-MFO, DE and FPA but standard deviation of E-MFO is better. The functions f_8,f_9,f_{10},f_{11} and f_{12} come up with very good results in case of E-MFO. Their values are near to global optimum values and better than other algorithms. For f_{13} E-MFO provides very poor standard deviation but its best solution is equal to optimum solution. Still, in comparison to others, E-MFO consider as best algorithm because of its best solution value. In case of f_{14}, the best solution provided by all the algorithms are comparable, but MFO provides minimum standard deviation so considered as best algorithm. The function f_{15} provide exact global solution in case of E-MFO. Function f_{16} offer better results in terms of best, worst, average and standard deviation for E-MFO algorithm. Function f_{17} gives comparable results for FA and E-MFO in terms of standard deviation but FA performs better than other algorithms because its best solution is close to global minima. Moreover, for function f_{18}, E-MFO, FA, FPA provides highly competitive results but E-MFO outperforms in terms of best value and standard deviation. For function f_{19}, MFO and E-MFO provides same value for standard deviation and for this particular function BA and BFP algorithm's performance is very poor. MFO, DE, FPA provides highly competitive results for function f_{20}, however MFO algorithm considered as best due to its slightly higher standard deviation.

Function	Algorithm	Best	Worst	Average	Std. Dev
f_1	E-MFO	**2.36E-31**	**6.64E-28**	**5.57E-29**	**1.19E-28**
	MFO	4.62E-06	2.00E+04	2.80E+03	5.72E+03
	BA	6.09E+03	3.99E+04	1.95E+04	8.10E+03
	BFP	2.29E+04	8.42E+04	6.32E+04	1.21E+04
	DE	4.80E+03	8.38E+04	5.49E+04	2.18E+04
	FA	0.001627	0.0055095	0.0029205	7.93E-04
	FPA	3.45E+02	1.99E+03	8.97E+02	3.30E+02
f_2	E-MFO	**4.79E-17**	**2.59E-15**	**4.99E-16**	**4.79E-16**
	MFO	3.23E-20	20.0000	1.2000000	3.8545486
	BA	0.799159	54.363152	20.212749	14.369427
	BFP	12.89518	50.763038	33.154089	7.8308621
	DE	0.058768	16.608305	3.1239795	3.3459719
	FA	0.058768	0.0063620	0.0027869	7.27E-04
	FPA	0.264359	3.7517812	1.4495611	0.7239958
f_3	E-MFO	**1.38E-21**	**3.34E-17**	**2.54E-18**	**7.45E-18**
	MFO	1.60E-11	5.00E+03	4.00E+02	1.37E+03
	BA	2.45E+02	1.53E+04	5.36E+03	2.89E+03
	BFP	8.06E+03	3.53E+04	1.85E+04	5.65E+03
	DE	9.198866	7.69E+03	1.47E+03	1.92E+03
	FA	8.10E-05	8.50E-04	2.70E-04	1.60E-04
	FPA	0.031425	0.6695819	0.136197	0.1103161
f_4	E-MFO	**1.92E-13**	**4.20E-12**	**1.23E-12**	**9.23E-13**
	MFO	4.39E-06	21.81578	2.241167	4.5042567
	BA	14.50311	61.435179	35.658991	9.8208726
	BFP	48.09316	83.318482	65.699653	8.2987753
	DE	1.723137	64.270224	28.148822	17.241827
	FA	0.002353	0.0112001	0.0062217	0.0016856
	FPA	0.480584	2.6076867	1.3954512	0.4344367
f_5	E-MFO	**27.77914**	**28.70866**	**28.20277**	**0.248776**
	MFO	62.24768	9.97E+09	3.99E+08	1,97E+09
	BA	3.83E+08	1.36E+10	3.07E+09	2.28E+09
	BFP	8.16E+09	4.10E+10	2.78E+10	8.58E+09
	DE	2.68E+09	4.22E+10	3.08E+10	6.90E+09
	FA	27.43123	1.39E+04	1.45E+03	3.03E+03
	FPA	5.15E+05	3.80E+07	1.07E+07	8.91E+06
f_6	E-MFO	**8.68E-06**	**2.23E-03**	**3.81E-04**	**3.67E-04**
	MFO	0.002000	0.017547	0.0059	0.0266
	BA	0.054085	4.6075454	0.7965411	0.8246265
	BFP	0.692697	11.896073	5.5161543	2.2357915
	DE	0.009755	2.1225818	0.226336	0.4254076
	FA	0.003236	0.1490810	0.0298927	0.0282967
	FPA	0.004622	0.0302129	0.0298927	0.0058381
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.0153561	5.04E-04	0.002345

	BA	1.77E-13	0.3928647	0.1190811	0.1114218
	BFP	0.004275	0.4663736	0.2946212	0.128708
	DE	0.00E+00	0.2575325	0.0335426	0.0510467
	FA	2.29E-12	5.66E-10	1.20E-10	1.10E-10
	FPA	0.00E+00	9.60E-13	6.38E-14	1.69E-13
f_8	E-MFO	**3.44E-29**	**2.69E-26**	**3.12E-27**	**5.85E-27**
	MFO	0.001179	1.00E+08	2.07E+08	2.40E+07
	BA	8.55E+04	1.80E+06	5.64E+05	3.77E+05
	BFP	2.96E+06	7.98E+06	5.90E+06	1.10E+06
	DE	4.80E+05	7.75E+06	4.78E+06	2.29E+06
	FA	0.21042	1.29E+02	31.256325	37.390463
	FPA	3.83E+04	2.11E+05	8.25E+04	3.34E+04
f_9	E-MFO	**5.07E-28**	**8.85E-25**	**8.13E-26**	**1.40E-25**
	MFO	5.40E+05	3.74E+08	3.65E+07	5.84E+07
	BA	1.38E+08	2.31E+09	7.88E+08	4.89E+08
	BFP	9.42E+08	4.38E+09	2.37E+09	7.85E+08
	DE	6.66E+07	1.86E+09	6.37E+08	5.02E+08
	FA	1.29E+06	1.86E+07	6.24E+06	3.72E+06
	FPA	8.18E+05	9.68E+06	3.08E+06	1.88E+06
f_{10}	E-MFO	**1.15E-114**	**1.81E-103**	**6.67E-105**	**2.86E-104**
	MFO	2.96E-23	2.29E-14	7.23E-16	3.52E-15
	BA	1.25E-08	1.25E-07	5.00E-08	2.66E-08
	BFP	0.020192	0.6319118	0.2592852	0.1612362
	DE	0.288252	1.5353542	0.775902	0.294377
	FA	2.89E-08	1.18E-06	2.82E-07	2.45E-07
	FPA	6.09E-10	5.09E-07	4.01E-08	5.44E-07
f_{11}	E-MFO	**3.54E-32**	**7.23E-29**	**9.46E-30**	**1.64E-29**
	MFO	1.63E-06	2.40E+03	6.44E+02	6.59E+02
	BA	2.15E+02	3.44E+03	9.30E+02	5.48E+02
	BFP	5.10E+03	1.15E+04	8.99E+03	1.37E+03
	DE	8.24E+02	1.23E+04	7.10E+03	3.63E+03
	FA	0.007319	1.8620043	0.273306	0.3845407
	FPA	44.46101	2.72E+02	1.13E+02	52.064633
f_{12}	E-MFO	**1.19E-30**	**8.40E-28**	**1.01E-28**	**1.58E-28**
	MFO	0.02636	3.85E+03	8.89E+02	1.12E+03
	BA	0.037852	2.24E+02	34.225633	50.957114
	BFP	4.34E+03	2.32E+04	1.11E+04	4.00E+03
	DE	3.67E+02	1.15E+04	2.64E+03	2.22E+03
	FA	0.1822	6.5718834	2.1574333	1.4794017
	FPA	4.952664	77.383768	24.34227	13.517725
f_{13}	E-MFO	**0.00E+00**	**1.95E+02**	**0.00E+00**	**71.49277**
	MFO	69.64709	2.30E+02	1.53E+02	37.32309
	BA	39.80093	2.27E+02	99.100651	45.018746
	BFP	2.83E+02	4.23E+02	3.66E+02	32.738443
	DE	1.56E+02	4.93E+02	3.62E+02	1.12E+02

	FA	23.88061	75.618224	47.779669	13.571661
	FPA	98.87448	1.70E+02	1.26E+02	13.248671
f_{14}	E-MFO	2.00E+01	2.00E+01	2.00E+01	3.57E-04
	MFO	**19.9999**	**19.9999**	**19.9999**	**5.36E-07**
	BA	19.9709	20.0000	19.9999	0.00425
	BFP	20.9921	21.2619	21.1331	0.07481
	DE	20.1399	20.5298	20.364	0.10511
	FA	0.04561	20.0058	19.5999	2.82185
	FPA	20.9167	21.1306	21.0462	0.04743
f_{15}	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	1.68E-05	1.81E+02	16.31648	39.52865
	BA	94.00835	4.19E+02	1.94E+02	65.96575
	BFP	3.26E+02	7.47E+02	5.84E+02	93.30520
	DE	2.70E+02	7.48E+02	5.82E+02	1.06E+02
	FA	0.002136	0.0150722	0.0055429	0.002858
	FPA	3.450933	17.305742	8.8112189	2.953524
f_{16}	E-MFO	**1.72E-17**	**3.76E-13**	**1.07E-14**	**5.33E-14**
	MFO	32.42624	6.01E+02	3.02E+02	1.18E+02
	BA	18.90162	4.69E+04	3.81E+03	9.12E+03
	BFP	4.13E+02	6.37E+08	3.48E+07	1.21E+08
	DE	7.97E+02	6.06E+04	1.26E+04	1.12E+04
	FA	9.239959	1.18E+02	46.929186	28.13372
	FPA	1.45E+02	8.33E+02	3.30E+02	1.91E+02
f_{17}	E-MFO	1.722771	2.389673	2.1147778	0.140332
	MFO	10.08956	88.30436	30.11427	12.76732
	BA	7.054811	46.566024	17.888414	8.837656
	BFP	1.21E+02	2.53E+02	1.80E+02	30.08757
	DE	31.31569	2.38E+02	92.044334	46.56949
	FA	**7.86E-06**	**0.9089115**	**0.0855539**	**0.191229**
	FPA	2.573172	22.556051	9.5783313	4.728633
f_{18}	E-MFO	3.09E-04	1.49E-03	5.21E-04	3.13E-04
	MFO	7.04E-04	0.0203633	0.0022678	0.004781
	BA	3.12E-04	0.0907032	0.0063442	0.0150543
	BFP	0.001482	0.1449052	0.0353342	0.0345246
	DE	0.001026	0.0024567	0.0013172	3.44E-04
	FA	3.46E-04	0.0024475	8.49E-04	3.84E-04
	FPA	**9.21E-04**	**0.0012438**	**0.0010333**	**4.48E-05**
f_{19}	E-MFO	-1.03163	-1.0316285	-1.031628	2.24E-16
	MFO	**-1.03163**	**-1.0316285**	**-1.031628**	**2.24E-16**
	BA	-1.03163	-1.0316285	-0.966335	0.2236679
	BFP	-1.03163	-1.0316285	-0.616376	0.4709543
	DE	-1.03163	-1.0316285	-1.031628	8.63E-09
	FA	-1.03163	-1.0316285	-1.031628	1.39E-09
	FPA	-1.03163	-1.0316285	-1.031628	2.23E-10
f_{20}	E-MFO	**3.0000000**	**3.0000000**	**3.0000000**	**1.62E-15**

	MFO	3.0000000	3.0000000	3.0000000	2.08E-15
	BA	3.0000000	84.000000	10.560000	22.517259
	BFP	3.0000000	96.485886	15.342473	19.364906
	DE	3.0000000	3.0000000	3.0000000	6.94E-15
	FA	3.0000000	3.0000000	3.0000000	1.45E-08
	FPA	3.0000000	3.0000000	3.0000000	3.60E-15

Table 5.5: E-MFO compared with other algorithms for population size 30

Case 2: Population size 60

As seen the Table 5.6, the E-MFO algorithm outperform other algorithms for functions f_1,f_2,f_3 and f_4. The best solution provided by this algorithm is very close to global minima. For function f_5, best solution provided by MFO is better whereas standard deviation is better in case of E-MFO but overall E-MFO algorithm considered as best for this function. E-MFO again outperforms other algorithms in case of f_6. For f_7, MFO and E-MFO both gives optimum solution. Further in case of f_8, f_9, f_{10}, f_{11} and f_{12} E-MFO is efficiently capable of avoiding local stagnation problem by providing results close to global optima. For function f_{13} only E-MFO provides exact optimum value but its standard deviation is poor. For f_{14}, best solution provided by all algorithm is comparable but standard deviation of MFO is good so MFO considered as best algorithm. In case of function f_{15}, E-MFO achieves exact global minima whereas other algorithms get stuck in some local minima. For function f_{16}, E-MFO is capable of achieving global minima in less number of iterations. Further in case of f_{17}, E-MFO outperforms other algorithms except FA. All the algorithms show competitive results in terms of best solution for function f_{18} but FPA attains somewhat better standard deviation so is taken as best algorithm. For the functions f_{19} and f_{20}, all the algorithms except BFP achieve global minima. For function f_{19}, E-MFO and MFO provides competitive results. For f_{20} the results of E-MFO are superior than that of other algorithms

Function	Algorithm	Best	Worst	Average	Std. Deviation
f_1	E-MFO	**3.10E-32**	**2.07E-29**	**2.76E-30**	**4.40E-30**
	MFO	5.86E-07	2.00E+04	1.60E+03	4.22E+03
	BA	4.69E+03	2.19E+04	1.11E+04	3.80E+03
	BFP	3.30E+04	7.30E+04	5.80E+04	9.96E+03
	DE	2.54E+04	7.63E+04	5.96E+04	1.22E+04
	FA	0.001599	0.005057	0.0029780	8.77E-04
	FPA	4.08E+02	1.31E+03	8.26E+02	2.26E+02
f_2	E-MFO	**3.65E-17**	**1.72E-15**	**2.17E-16**	**2.60E-16**
	MFO	2.47E-15	20.0000	1.20000	3.85454
	BA	0.001851	40.1189	7.71867	10.88550
	BFP	18.55163	49.0700	32.2295	7.62291

	DE	0.073607	6.47732	1.22106	1.19288
	FA	0.001318	0.01428	0.00293	0.00178
	FPA	2.30964	5.55647	3.94546	0.91879
f_3	E-MFO	**3.96E-21**	**2.77E-17**	**1.33E-18**	**4.19E-18**
	MFO	5.03E-09	5.00E+02	5.00E+02	5.00E+02
	BA	2.99E+02	1.10E+04	3.66E+03	2.12E+03
	BFP	5.75E+03	1.94E+04	1.34E+04	3.51E+03
	DE	16.13605	4.54E+03	8.75E+02	9.75E+02
	FA	7.92E-05	8.24E-04	2.91E-04	1.58E-04
	FPA	0.24565	2.33319	0.97460	0.45213
f_4	E-MFO	**5.09E-14**	**1.93E-12**	**4.56E-13**	**3.71E-13**
	MFO	3.47E-07	9.36E-05	1.23E-05	1.90E-05
	BA	7.264344	44.34353	25.06512	8.340469
	BFP	40.90571	75.04778	59.20485	6.672909
	DE	5.097891	37.07507	13.85543	5.956982
	FA	0.003688	0.010373	0.006439	0.001576
	FPA	1.795477	5.389310	3.285244	0.718641
f_5	E-MFO	**27.87864**	**28.64255**	**28.07047**	**0.101667**
	MFO	17.83941379	1.00E+06	2.41E+05	4.31E+05
	BA	2.56E+08	3.64E+09	1.06E+09	7.18E+08
	BFP	9.48E+09	3.45E+10	2.54E+10	6.50E+09
	DE	1.17E+10	3.79E+10	2.76E+10	6.02E+09
	FA	26.93427133	7.65E+03	1.07E+03	2.16E+03
	FPA	1.40E+06	2.49E+07	9.33E+06	4.54E+06
f_6	E-MFO	**7.50E-06**	**1.22E-03**	**2.18E-04**	**2.19E-04**
	MFO	6.63E-04	0.001834	6.63E-04	0.001834
	BA	0.047906	0.868799	0.033036	0.192910
	BFP	0.999476	8.414058	4.550350	1.855117
	DE	0.007706	0.552526	0.096286	0.104484
	FA	0.001542	0.075146	0.018333	0.018002
	FPA	0.006577	0.033036	0.016295	0.006246
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	BA	2.69E-14	0.2979932	0.0450544	0.0618240
	BFP	2.35E-04	0.4242079	0.2258614	0.1134229
	DE	0.00E+00	0.0373844	0.0027551	0.0065351
	FA	1.15E-12	0.0031266	2.50E-04	8.57E-04
	FPA	4.44E-16	9.37E-12	1.01E-12	1.95E-12
f_8	E-MFO	**1.99E-30**	**1.28E-27**	**1.72E-28**	**2.65E-28**
	MFO	1.58E-04	1.00E+06	1.00E+05	3.03E+05
	BA	0.1364456	5.35E+04	8.88E+03	1.64E+04
	BFP	3.26E+06	6.58E+06	5.02E+06	9.54E+05
	DE	1.49E+06	7.74E+06	5.21E+06	1.79E+06

	FA	0.2390960	1.01E+02	15.417394	19.176159
	FPA	3.44E+04	1.08E+05	6.89E+04	1.70E+04
f_9	E-MFO	**2.69E-29**	**7.27E-26**	**7.80E-27**	**1.39E-26**
	MFO	7.77E+05	1.53E+08	2.78E+07	2.82E+07
	BA	4.34E+07	1.05E+09	3.78E+08	2.04E+08
	BFP	8.71E+08	3.05E+09	1.91E+09	5.99E+08
	DE	4.93E+07	1.53E+09	3.67E+08	3.20E+08
	FA	9.09E+05	9.44E+06	3.80E+06	1.83E+06
	FPA	4.20E+06	1.33E+07	7.14E+06	2.04E+06
f_{10}	E-MFO	**5.02E-118**	**6.21E-107**	**1.27E-108**	**8.77E-108**
	MFO	4.61E-27	1.24E-20	1.06E-21	2.78E-21
	BA	5.04E-09	2.33E-08	1.21E-08	4.64E-09
	BFP	4.15E-04	0.159422	0.0505439	0.039257
	DE	0.191838	1.114176	0.6971725	0.228109
	FA	9.06E-09	3.40E-07	1.01E-07	7.03E-08
	FPA	1.47E-08	1.78E-05	2.29E-06	3.24E-06
f_{11}	E-MFO	**5.96E-33**	**6.93E-30**	**5.64E-31**	**1.16E-30**
	MFO	4.17E-07	2.10E+03	2.90E+02	4.63E+02
	BA	2.24E-04	1.90E+02	18.547169	35.36918
	BFP	2.91E+03	1.03E+04	7.40E+03	1.52E+03
	DE	2.03E+03	1.02E+04	6.52E+03	2.30E+03
	FA	0.001985	0.6926388	0.158906	0.1851881
	FPA	58.69568	1.68E+02	1.03E+02	23.640397
f_{12}	E-MFO	**8.38E-32**	**1.74E-28**	**1.13E-29**	**2.68E-29**
	MFO	0.010302	3.71E+03	7.03E+02	9.75E+02
	BA	0.005754	0.230388	0.017592	0.0324171
	BFP	2.34E+03	1.29E+04	6.49E+03	2.32E+03
	DE	4.33E+02	8.35E+03	3.27E+03	1.90E+03
	FA	0.090063	6.315860	1.364417	1.149367
	FPA	12.57818	90.08843	33.40407	15.63032
f_{13}	E-MFO	**00.E+00**	**2.04E+02**	**0.00E+00**	**56.39479**
	MFO	55.71759	2.26E+02	1.38E+02	37.70682
	BA	28.85562	1.31E+02	74.90292	23.84936
	BFP	1.91E+02	3.69E+02	2.94E+02	35.09971
	DE	1.52E+02	4.67E+02	3.82E+02	76.49085
	FA	19.90106	70.64365	34.92485	10.60092
	FPA	1.20E+02	1.72E+02	1.42E+02	11.85016
f_{14}	E-MFO	2.00E+01	2.00E+01	2.00E+01	3.26E-04
	MFO	**19.99999**	**19.99999**	**19.99999**	**6.98E-08**
	BA	19.97241	20.00002	19.99911	0.003929
	BFP	20.54088	21.00324	20.81227	0.096971
	DE	20.17702	20.50006	20.31751	0.088286
	FA	0.032016	20.00210	14.78980	8.829073
	FPA	20.85078	21.09232	21.01417	0.053304

f_{15}	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	3.28E-06	1.80E+02	19.90594	41.97550
	BA	31.39265	2.03E+02	1.17E+02	35.87847
	BFP	3.02E+02	6.90E+02	5.39E+02	90.43190
	DE	1.13E+02	6.96E+02	5.39E+02	1.09E+02
	FA	0.002745	0.012703	0.004987	0.001993
	FPA	5.2551690	13.89611	8.4289974	1.839532
f_{16}	E-MFO	**5.85E-18**	**6.65E-14**	**3.32E-15**	**1.03E-14**
	MFO	4.508310074	5.00E+02	2.71E+02	1.21E+02
	BA	3.89E-05	3.76E+03	5.68E+02	7.81E+02
	BFP	3.50E+02	6.10E+06	2.03E+05	8.87E+05
	DE	7.88E+02	2.50E+04	5.55E+03	5.26E+03
	FA	7.427830225	80.463224	29.941753	16.61655
	FPA	2.71E+02	1.33E+03	6.62E+02	2.50E+02
f_{17}	E-MFO	1.833941	2.351627	2.121151	0.106055
	MFO	11.72771	98.66733	26.84347	13.80313
	BA	4.261282	38.35813	11.37968	5.724066
	BFP	89.35431	1.85E+02	1.36E+02	25.96749
	DE	36.03824	2.12E+02	94.43658	42.49181
	FA	**9.60E-06**	**0.089539**	**0.004246**	**0.017819**
	FPA	8.179656	29.76226815	18.87273	4.652033
f_{18}	E-MFO	3.10E-04	1.35E-03	5.24E-04	3.44E-04
	MFO	4.64E-04	0.001655	0.001030	3.31E-04
	BA	3.07E-04	0.013934	0.001907	0.002103
	BFP	3.09E-04	0.126919	0.027299	0.034561
	DE	0.001007	0.001794	0.001134	2.32E-04
	FA	3.07E-04	0.001594	6.71E-04	2.03E-04
	FPA	**9.03E-04**	**0.001048**	**0.001019**	**2.70E-05**
f_{19}	E-MFO	**-1.031628**	**-1.031628**	**-1.031628**	**2.24E-16**
	MFO	-1.031628	-1.031628	-1.031628	2.24E-16
	BA	-1.031628	-0.215463	-1.015305	0.115423
	BFP	-1.031628	-0.215463	-0.928243	0.268649
	DE	-1.031628	-1.031628	-1.031628	2.62E-16
	FA	-1.031628	-1.031628	-1.031628	1.24E-09
	FPA	-1.031628	-1.031628	-1.031628	1.35E-10
f_{20}	E-MFO	**3.0000000**	**3.0000000**	**3.0000000**	**1.33E-15**
	MFO	3.0000000	3.0000000	3.0000000	1.66E-15
	BA	3.0000000	30.00000011	4.080000024	5.344613
	BFP	3.0000000	84.00000924	7.860001612	14.10892
	DE	3.0000000	3.0000000	3.0000000	3.16E-15
	FA	3.0000000	3.0000000	3.0000000	1.63E-08
	FPA	3.0000000	3.0000000	3.0000000	7.19E-15

Table 5.6: E-MFO compared with other algorithms for population size 60

Case 3: Population size 100

In this case, all the algorithms are run for 100 population size to evaluate the effect of change of population size. Starting from functions f_1,f_2,f_3 and f_4, it is observed from Table 5.7 that E-MFO performs better in comparison to other algorithms. For function f_5, E-MFO and FA provides competitive results for best solution but E-MFO provides better standard deviation so taken as best algorithm for this particular function. The function f_6 gives better results in case of E-MFO. In case of f_7, the results provided by MFO and E-MFO are same as well as better than others. Also, both algorithms have achieved global optimum values. For functions f_8, f_9, f_{10}, f_{11} and f_{12} provides best results in case of E-MFO. Function f_{13} gives exact optimum results in E-MFO whereas other algorithms get stuck in local minima. But, its standard deviation is poor in comparison to others. The results provided by functions f_{14} and f_{15} are very close to optimum values in case of MFO and E-MFO respectively. For f_{16}, E-MFO performs better in comparison to all other given algorithms. Although for f_{17}, E-MFO is better in comparison to MFO but FA gives better results in this case. For function f_{18}, E-MFO, MFO, DE, FA, FPA give highly competitive results but standard deviation is lower for FPA. Further, for f_{19} E-MFO and MFO both provides sane value for standard deviation. For function f_{20}, E-MFO, MFO, DE and FPA provides competitive results but E-MFO considered as best.

Function	Algorithm	Best	Worst	Average	Std. Deviation
f_1	E-MFO	**3.51E-33**	**8.82E-30**	**7.33E-31**	**1.52E-30**
	MFO	1.35E-06	1.00E+04	8.00E+02	2.74E+03
	BA	6.71E+02	1.32E+04	5.87E+03	2.80E+03
	BFP	3.45E+04	6.73E+04	5.53E+04	7.65E+03
	DE	3.62E+04	6.90E+04	5.91E+04	8.05E+03
	FA	0.0012483	0.006041	0.003129	0.0010461
	FPA	6.97E+02	2.79E+03	1.46E+03	3.66E+02
f_2	E-MFO	**8.37E-18**	**2.65E-16**	**1.09E-16**	**7.64E-17**
	MFO	5.52E-15	10.00000	0.400000	1.979486
	BA	0.001311	43.92925	2.575453	8.486372
	BFP	13.82175	51.00618	30.53601	7.196034
	DE	0.357783	4.386435	1.828253	0.947587
	FA	0.001459	0.004149	0.002642	5.48E-04
	FPA	2.630346	6.822717	5.131647	0.965264
f_3	E-MFO	**9.52E-22**	**2.70E-18**	**3.38E-19**	**5.72E-19**
	MFO	5.13E-11	5.00E+03	2.00E+02	9.90E+02
	BA	1.37E+02	5.08E+03	1.77E+03	1.20E+03
	BFP	4.50E+03	1.86E+04	1.06E+04	3.23E+03
	DE	70.24188	5.63E+03	1.32E+03	1.10E+03
	FA	1.03E-04	8.86E-04	2.91E-04	1.61E-04

	FPA	0.655377	3.720407	2.016670	0.690653
f_4	E-MFO	**4.34E-14**	**9.05E-13**	**1.74E-13**	**1.41E-13**
	MFO	1.21E-07	1.73E-05	2.68E-06	2.95E-06
	BA	3.826875	32.38125	17.13029	6.872499
	BFP	41.07376	70.95114	55.61738	6.886787
	DE	6.480239	25.90295	15.26117	5.192262
	FA	0.003825	0.011335	0.006487	0.001515
	FPA	2.769895	6.735046	4.724890	0.866823
f_5	E-MFO	**27.46182**	**28.12800**	**28.04558**	**0.107252**
	MFO	14.17205	1.00E+06	1.82E+05	3.87E+05
	BA	1.84E+07	5.73E+08	2.46E+08	1.47E+08
	BFP	6.30E+09	3.18E+10	2.15E+10	5.05E+09
	DE	1.05E+10	3.46E+10	2.50E+10	5.80E+09
	FA	26.83713	6.43E+03	5.52E+02	1.37E+03
	FPA	6.46E+06	6.12E+07	3.04E+07	1.33E+07
f_6	E-MFO	**4.72E-06**	**5.08E-04**	**1.46E-04**	**1.26E-04**
	MFO	3.32E-04	0.004967	0.002474	9.85E-04
	BA	0.022020	0.597634	0.193514	0.1246816
	BFP	0.476258	7.449452	3.448125	1.4499283
	DE	0.011643	0.194339	0.097394	0.0424953
	FA	1.89E-04	0.040243	0.011021	0.0091707
	FPA	0.00350197	0.040243	0.015715	0.0070141
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	BA	4.44E-16	0.1482220	0.019866	0.0261309
	BFP	3.62E-07	0.3811390	0.150564	0.1050442
	DE	0.00E+00	0.0133423	4.28E-04	0.0019433
	FA	8.73E-12	0.0080897	4.04E-04	0.0015098
	FPA	1.55E-15	9.84E-12	6.30E-13	1.50E-12
f_8	E-MFO	**1.71E-31**	**2.29E-28**	**3.76E-29**	**5.23E-29**
	MFO	21.56645	2.00E+06	5.40E+05	7.06E+05
	BA	0.371804	40.63813	5.995335	8.215169
	BFP	3.63E+06	1.08E+07	8.76E+06	1.53E+06
	DE	9.17E+06	1.31E+07	1.14E+07	8.62E+05
	FA	4.519641	70.57293	20.10809	11.73150
	FPA	2.69E+05	7.04E+05	4.91E+05	1.14E+05
f_9	E-MFO	**8.58E-29**	**1.30E-26**	**1.16E-27**	**2.14E-27**
	MFO	2.53E+05	1.60E+08	2.29E+07	3.21E+07
	BA	2.10E+07	4.48E+08	1.23E+08	7.94E+07
	BFP	3.38E+08	2.35E+09	1.50E+09	4.19E+08
	DE	5.05E+07	1.07E+09	3.04E+08	2.22E+08
	FA	5.04E+05	6.49E+06	2.05E+06	1.27E+06
	FPA	7.78E+06	3.07E+07	1.70E+07	4.23E+06

f_{10}	E-MFO	**4.59E-119**	**2.19E-112**	**9.68E-114**	**3.37E-113**
	MFO	3.71E-17	6.83E-11	2.85E-12	1.21E-11
	BA	1.50E-09	1.07E-08	5.08E-09	2.14E-09
	BFP	1.78E-04	0.109785	0.009185	0.017004
	DE	0.305766	1.213377	0.772623	0.200543
	FA	5.73E-09	2.57E-07	5.63E-08	5.43E-08
	FPA	3.19E-06	2.18E-04	4.96E-05	4.74E-05
f_{11}	E-MFO	**2.88E-33**	**5.14E-31**	**6.62E-32**	**1.01E-31**
	MFO	0.078572	5.80E+03	2.25E+03	1.80E+03
	BA	0.001845	4.06E+01	0.969132	5.753952
	BFP	8.92E+03	2.78E+04	2.06E+04	4.24E+03
	DE	2.10E+04	3.17E+04	2.75E+04	2.28E+03
	FA	0.646657	18.873705	4.0483788	3.461136
	FPA	6.56E+02	1.88E+03	1.11E+03	2.47E+02
f_{12}	E-MFO	**3.53E-32**	**7.41E-29**	**4.84E-30**	**1.26E-29**
	MFO	2.926888	9.41E+03	2.57E+03	2.41E+03
	BA	0.013428	0.029649	0.021994	0.003142
	BFP	5.32E+03	2.04E+04	1.25E+04	3.49E+03
	DE	7.62E+03	4.00E+04	2.03E+04	7.25E+03
	FA	0.908233	13.161332	5.8621089	3.0729248
	FPA	1.82E+02	6.25E+02	3.80E+02	1.14E+02
f_{13}	E-MFO	**0.00E+00**	**2.24E+02**	**0.00E+00**	**46.250770**
	MFO	1.34E+02	3.30E+02	2.45E+02	42.858611
	BA	50.750472	1.93E+02	1.06E+02	33.180211
	BFP	3.27E+02	5.42E+02	4.66E+02	57.456312
	DE	6.49E+02	7.92E+02	7.49E+02	32.709280
	FA	28.879126	77.686828	48.778460	11.100252
	FPA	2.70E+02	3.62E+02	3.14E+02	20.987951
f_{14}	E-MFO	2.00E+01	2.00E+01	2.00E+01	4.14E-04
	MFO	**19.99999**	**19.99999**	**19.99999**	**8.83E-08**
	BA	19.95783	20.00002	19.99831	0.006709
	BFP	20.21452	20.73213	20.47791	0.105519
	DE	20.13510	20.65240	20.39644	0.133967
	FA	0.025757	19.99658	4.823835	8.595871
	FPA	20.868585	21.08081	20.983183	0.048196
f_{15}	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.2747030	2.71E+02	49.291682	70.975315
	BA	82.67836	2.35E+02	1.48E+02	33.989769
	BFP	3.98E+02	1.15E+03	9.48E+02	1.79E+02
	DE	8.38E+02	1.21E+03	1.03E+03	85.277153
	FA	0.007460	0.019512	1.19E-02	0.0024736
	FPA	25.53438	61.80807	43.94408	8.6594561
	E-MFO	**4.15E-18**	**1.16E-14**	**1.63E-15**	**2.58E-15**

f_{16}	MFO	2.71E+02	8.65E+02	6.55E+02	1.35E+02
	BA	0.0012761	1.33E+04	1.35E+03	2.04E+03
	BFP	9.31E+02	9.97E+07	3.09E+06	1.44E+07
	DE	2.78E+03	7.27E+04	1.70E+04	1.72E+04
	FA	61.229569	1.95E+02	1.18E+02	30.766435
	FPA	7.83E+02	4.86E+03	2.58E+03	9.32E+02
f_{17}	E-MFO	1.6541929	2.322980	2.134220	0.113337
	MFO	19.573907	1.20E+02	55.74949	20.04667
	BA	6.9738610	35.13761	14.43145	5.508628
	BFP	1.24E+02	2.89E+02	2.20E+02	38.75242
	DE	2.19E+02	4.37E+02	3.66E+02	49.72360
	FA	**3.14E-04**	**0.502528**	**0.028050**	**0.077812**
	FPA	34.210730	78.65889	56.33422	9.203493
f_{18}	E-MFO	3.13E-04	0.001488	6.07E-04	4.23E-04
	MFO	3.07E-04	0.001655	9.31E-04	3.54E-04
	BA	3.07E-04	0.012662	0.001551	0.001837
	BFP	5.69E-04	0.096422	0.010643	0.018388
	DE	9.84E-04	0.001794	0.001157	2.68E-04
	FA	3.07E-04	9.52E-04	6.18E-04	1.42E-04
	FPA	**8.51E-04**	**0.001039**	**0.001019**	**3.30E-05**
f_{19}	E-MFO	**-1.031628**	**-1.031628**	**-1.031628**	**2.24E-16**
	MFO	-1.031628	-1.031628	-1.031628	2.24E-16
	BA	-1.031628	-1.031628	-1.031628	2.66E-10
	BFP	-1.031628	-0.215463	-0.966335	0.223667
	DE	-1.031628	-1.031628	-1.031628	2.24E-16
	FA	-1.031628	-1.031628	-1.031628	9.53E-10
	FPA	-1.031628	-1.031628	-1.031628	9.56E-11
f_{20}	E-MFO	**3.000000**	**3.000000**	**3.000000**	**1.87E-15**
	MFO	3.000000	3.000000	3.000000	2.00E-15
	BA	3.000000	3.000000	3.000000	1.03E-08
	BFP	3.000000	84.00000576	8.940000794	14.72802462
	DE	3.000000	3.000000	3.000000	3.87E-15
	FA	3.000000	3.000000	3.000000	9.23E-09
	FPA	3.000000	3.000000	3.000000	6.50E-15

Table 5.7: Results of E-MFO and other algorithms for population size 100

Inference: This test is carried out on 20 benchmark functions for three different population sizes 30, 60 and 100. In first two cases, MFO is better in two functions, E-MFO is better in seventeen functions, FPA and FA is in one function. In third case, DE obtained good results in one function and for rest of functions same pattern is followed as in first two cases. The performance of E-MFO is almost same for all population sizes. On other hand, with increase in population size the results of MFO gets better but they are not as good as E-MFO. FA algorithm also provides good and comparable results to MFO as population size increase. BA fails to achieve global minima in all cases. Out of all given algorithms BFP is proven to be a worst algorithm. After all this discussion, it is concluded that E-MFO gives almost similar and comparable results on all populations and also outperforms in comparison to other algorithms. The population size 30 is taken for further experimentation to avoid the increase of number evaluations.

5.2.2 Effect of dimension (Scalability Study)

The performance of proposed approach is also evaluated on different dimension sizes. Five different dimensions (30, 50, 100, 200 and 500) are considered to study its effect on MFO and E-MFO. The results of MFO and E-MFO are compared with other meta-heuristic algorithms like BA, BFP, DE, FA and FPA. The performance of MFO and E-MFO for different dimension size is evaluated only on unimodal and multimodal functions which are given in Table 5.2 and 5.3. Out of twenty functions, there are three fixed dimension functions whose dimension can't be changed. Hence, performance is evaluated on seventeen functions. During this experimentation, population size is taken as 30. The results for dimension size 30 has already been discussed in section 4.3 under population size of 30. From Table 5.5, it is clear that E-MFO performs better in sixteen functions and FA is good in only one function (f_{17}) for dimension size 30. In case of dimension size 50, E-MFO achieves global minima for all the functions except f_{14} and f_{17} as shown in Table 5.8. For function f_{14}, MFO provides better results and for f_{17}, FA provides better results. Table 5.9 shows simulation results for dimension size 100 and results show that E-MFO outperforms in all functions except f_{14} (good in MFO). Simulation results for dimension size 200 is given in Table 5.10. The results show that BA comes up with good results in f_{14} and for rest of functions, E-MFO give superior performance. Table 5.11 depicts the results for dimension size 500. Again, E-MFO provides better results for all the functions except f_{14}. For f_{14} function, all the algorithms provide competitive results in terms of best, worst, average and standard deviation. But, FPA considered as best due to better standard deviation.

Function	Algorithm	Best	Worst	Average	Std. Dev
f_1	E-MFO	**3.54E-24**	**2.49E-21**	**3.54E-22**	**5.03E-22**
	MFO	3.908591	3.00E+04	6.77E+03	7.83E+03
	BA	1.71E+04	5.73E+04	3.28E+04	9.33E+03
	BFP	6.28E+04	1.49E+05	1.13E+05	2.14E+04
	DE	1.05E+05	1.44E+05	1.27E+05	1.02E+04
	FA	0.012525	3.62E-02	0.021249	0.005907
	FPA	1.80E+03	5.03E+03	3.22E+03	7.93E+02
f_2	E-MFO	**2.01E-13**	**1.26E-11**	**2.18E-12**	**2.78E-12**

	MFO	20.13997	1.40E+02	70.54696	30.87216
	BA	28.56251	1.34E+13	4.18E+11	2.14E+12
	BFP	1.00E+10	8.77E+19	2.58E+18	1.27E+19
	DE	88.01279	3.60E+02	1.67E+02	63.16034
	FA	0.35757	3.511229	0.908903	0.704378
	FPA	29.58089	67.71294	41.87777	7.817614
f_3	E-MFO	**3.68E-15**	**2.15E-11**	**1.49E-12**	**3.65E-12**
	MFO	1.51E+04	1.25E+05	4.84E+04	2.51E+04
	BA	4.02E+04	3.32E+05	1.56E+05	6.56E+04
	BFP	1.63E+05	7.75E+05	3.72E+05	1.54E+05
	DE	5.54E+04	3.05E+05	1.27E+05	5.19E+04
	FA	6.79E+03	2.98E+04	1.32E+04	4.12E+03
	FPA	2.29E+03	7.71E+03	4.59E+03	1.27E+03
f_4	E-MFO	**1.57E-10**	**9.36E-09**	**1.71E-09**	**1.82E-09**
	MFO	69.23615	93.2834	82.31819	6.015157
	BA	34.85881	86.54522	56.76027	9.872661
	BFP	81.19559	96.1857	90.47538	3.569025
	DE	85.35093	95.74333	92.60194	2.274813
	FA	0.286266	8.790009	2.68246	1.774635
	FPA	14.99628	27.35923	20.16288	2.760949
f_5	E-MFO	**47.76318**	**48.863067**	**48.37563**	**0.323467**
	MFO	6.52E+04	9.98E+09	9.97E+08	3.01E+09
	BA	2.00E+09	1.39E+10	5.53E+09	2.93E+09
	BFP	2.25E+10	7.35E+10	5.65E+10	1.06E+10
	DE	4.08E+10	7.73E+10	6.33E+10	7.89E+09
	FA	48.23219	2.05E+05	1.58E+04	4.51E+04
	FPA	1.76E+07	1.19E+08	5.41E+07	2.46E+07
f_6	E-MFO	**2.13E-05**	**1.60E-04**	**4.33E-04**	**3.36E-04**
	MFO	0.247644	85.65464	18.16018	22.83136
	BA	4.751723	30.74124	14.02752	5.432146
	BFP	1.51E+02	3.96E+02	2.72E+02	60.57708
	DE	2.96E+02	5.26E+02	4.04E+02	57.48182
	FA	0.022205	0.221239	0.064656	0.035351
	FPA	0.256331	1.099411	0.625464	0.204166
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.001125	2.89E-05	1.65E-04
	BA	4.73E-14	0.343255	0.077117	0.087457
	BFP	0.015626	0.475936	0.287786	0.125548
	DE	5.90E-06	0.257498	0.042654	0.05665
	FA	7.04E-13	3.31E-10	1.06E-10	9.15E-11
	FPA	0.00E+00	3.42E-12	4.08E-13	7.59E-13
f_8	E-MFO	**8.40E-21**	**6.17E-17**	**3.84E-18**	**1.03E-17**
	MFO	1.65E+05	4.00E+06	7.42E+05	9.84E+05
	BA	5.87E+05	3.68E+06	1.68E+06	7.20E+05
	BFP	6.25E+06	1.37E+07	1.10E+07	1.88E+06

	DE	1.02E+07	1.39E+07	1.23E+07	1.88E+06
	FA	7.223926	2.20E+02	47.11834	46.68087
	FPA	1.65E+05	4.77E+05	3.13E+05	6.86E+04
f_9	E-MFO	**2.48E-28**	**4.10E-25**	**5.77E-26**	**8.78E-26**
	MFO	3.75E+05	3.00E+08	4.21E+07	5.88E+07
	BA	9.82E-55	1.82E+09	7.62E+08	4.20E+08
	BFP	1.01E+09	4.61E+09	2.53E+09	8.34E+08
	DE	3.72E+07	1.85E+09	5.53E+08	4.19E+08
	FA	1.38E+06	1.44E+07	6.14E+06	3.08E+06
	FPA	5.96E+05	1.15E+07	3.34E+06	2.08E+06
f_{10}	E-MFO	**1.7E-110**	**4.96E-97**	**9.93E-99**	**7.02E-98**
	MFO	1.00E+02	7.40E+03	2.74E+03	1.94E+03
	BA	1.63E+03	8.26E+03	4.04E+03	1.63E+03
	BFP	1.53E+04	3.36E+04	2.77E+04	3.92E+03
	DE	8.41E+03	3.59E+04	2.96E+04	5.49E+03
	FA	0.227668	37.98557	7.492614	7.847924
	FPA	3.87E+02	1.37E+03	7.19E+02	2.02E+02
f_{11}	E-MFO	**4.60E-25**	**2.12E-21**	**1.94E-22**	**4.38E-22**
	MFO	1.10E-13	3.00E-06	7.86E-08	4.31E-07
	BA	1.80E-08	1.35E-07	6.16E-08	2.70E-08
	BFP	0.034787	0.783276	0.384903	0.199591
	DE	0.549786	1.873582	1.09182	0.316775
	FA	5.59E-08	8.93E-07	3.33E-07	2.22E-07
	FPA	1.17E-09	4.20E-07	6.07E-08	8.43E-08
f_{12}	E-MFO	**1.66E-24**	**1.84E-21**	**1.54E-22**	**2.96E-22**
	MFO	84.99353	8.08E+03	3.42E+03	2.41E+03
	BA	13.16207	1.31E+03	2.54E+02	2.87E+02
	BFP	7.41E+03	4.59E+04	2.83E+04	8.52E+03
	DE	3.28E+03	5.04E+04	1.91E+04	1.28E+04
	FA	3.98546	31.06265	14.88969	6.463814
	FPA	60.48988	4.43E+02	2.09E+02	88.57409
f_{13}	E-MFO	**0.00E+00**	**429.4064**	**248.5947**	**159.8702**
	MFO	2.18E+02	4.45E+02	3.05E+02	50.9097
	BA	58.71291	3.02E+02	1.46E+02	50.342
	BFP	5.94E+02	7.56E+02	6.79E+02	33.99931
	DE	5.95E+02	8.47E+02	7.78E+02	41.85033
	FA	55.91358	1.78E+02	96.22701	23.74788
	FPA	2.27E+02	3.23E+02	2.77E+02	20.44704
f_{14}	E-MFO	20.0000	2.0608	20.0020	0.008654
	MFO	**20.0000**	**20.0000**	**20.00000**	**2.54E-07**
	BA	19.99671	20.00007	19.99983	5.12E-04
	BFP	21.17698	21.3594	21.28038	0.042162
	DE	20.36133	20.83655	20.56707	0.086475
	FA	20.00797	20.03609	20.02193	0.006991
	FPA	21.05664	21.28213	20.02193	0.053717

f_{15}	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.883172	3.61E+02	68.30557	79.01692
	BA	1.45E+02	5.80E+02	3.24E+02	1.00E+02
	BFP	6.17E+02	1.27E+03	1.04E+03	1.64E+02
	DE	9.08E+02	1.36E+03	1.13E+03	85.53582
	FA	0.00539	0.020282	0.011161	0.002371
	FPA	18.49829	39.19273	27.77052	5.2091
f_{16}	E-MFO	**8.94E-12**	**3.73E-08**	**4.25E-09**	**7.63E-09**
	MFO	1.43E+02	1.13E+03	8.12E+02	1.40E+02
	BA	1.94E+02	4.58E+05	1.73E+04	6.58E+04
	BFP	1.01E+03	6.36E+10	3.11E+09	1.11E+10
	DE	8.62E+02	4.13E+05	6.29E+04	8.22E+04
	FA	1.43E+02	5.41E+02	3.16E+02	80.22963
	FPA	1.43E+02	7.83E+03	2.12E+03	1.16E+03
f_{17}	E-MFO	3.482678	4.370867	3.901378	0.147334
	MFO	43.82359	1.24E+02	70.74052	21.06246
	BA	10.08134	97.85139	38.2227	17.55698
	BFP	1.84E+02	4.74E+02	3.56E+02	60.09763
	DE	1.35E+02	5.26E+02	3.78E+02	92.99365
	FA	**5.37E-04**	**2.729747**	**0.384985**	**0.503965**
	FPA	5.974415	2.729747	20.72206	7.788164

Table 5.8: Results of E-MFO compared with others in case of dimension size 50

Function	Algorithm	Best	Worst	Average	Std. Dev
f_1	E-MFO	**2.97E-18**	**1.01E-14**	**8.47E-16**	**2.08E-15**
	MFO	5.24E+03	7.91E+04	3.69E+04	1.69E+04
	BA	3.89E+04	1.58E+05	7.41E+04	2.53E+04
	BFP	1.36E+05	2.95E+05	2.42E+05	4.47E+04
	DE	2.42E+05	2.98E+05	2.71E+05	1.39E+04
	FA	0.130716	2.98E+05	0.19235	0.031247
	FPA	6.05E+03	1.35E+04	9.80E+03	1.50E+03
f_2	E-MFO	**2.70E-11**	**4.40E-09**	**8.45E-10**	**1.00E-09**
	MFO	76.70737	2.89E+02	1.76E+02	54.33873
	BA	1.54E+02	7.67E+33	1.53E+32	1.08E+33
	BFP	2.87E+21	2.54E+47	5.50E+45	3.59E+46
	DE	2.64E+02	5.54E+50	1.20E+49	7.83E+49
	FA	4.601386	50.97501	20.94172	10.98618
	FPA	76.70737	1.09E+02	90.12284	9.384296
f_3	E-MFO	**6.39E-11**	**0.021467**	**4.34E-04**	**0.003090**
	MFO	1.00E+05	3.30E+05	2.02E+05	5.85E+04
	BA	2.67E+05	1.27E+06	6.00E+05	2.84E+05
	BFP	6.75E+05	3.58E+06	1.44E+06	5.34E+05
	DE	6.75E+05	1.14E+06	5.17E+05	2.16E+05
	FA	5.14E+04	1.06E+05	7.53E+04	1.40E+04
	FPA	1.23E+04	3.88E+04	2.12E+04	5.72E+03
f_4	E-MFO	**7.73E-08**	**2.23E-06**	**6.04E-07**	**4.99E-07**

	MFO	87.53573	96.59618	93.14778	2.081455
	BA	45.44762	81.25122	64.15183	8.183842
	BFP	86.47633	97.31797	94.97631	2.055199
	DE	94.40296	97.93031	96.33653	2.055199
	FA	94.40296	33.05645	25.59123	4.133269
	FPA	20.38937	30.56896	25.05148	2.139599
f_5	E-MFO	**97.88309**	**98.80912**	**98.447167**	**0.301765**
	MFO	6.56E+08	2.53E+10	8.06E+09	6.03E+09
	BA	1.78E+08	3.80E+10	1.24E+10	7.40E+09
	BFP	3.24E+10	1.65E+11	1.19E+11	3.84E+10
	DE	1.15E+11	1.73E+11	1.47E+11	1.06E+10
	FA	1.24E+02	1.85E+05	1.61E+04	3.62E+04
	FPA	9.98E+07	5.01E+08	2.74E+08	8.31E+07
f_6	E-MFO	**4.49E-05**	**1.50E-04**	**6.51E-04**	**4.21E-04**
	MFO	16.41162	4.54E+02	1.49E+02	1.04E+02
	BA	15.0506	60.23618	30.2305	9.421689
	BFP	3.65E+02	1.93E+03	1.31E+03	4.30E+02
	DE	1.51E+03	2.30E+03	1.91E+03	1.64E+02
	FA	0.089564	0.293075	0.193614	0.044121
	FPA	1.744613	8.760175	3.8938	1.365825
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	BA	2.71E-14	0.345929	0.091339	0.086893
	BFP	2.03E-11	0.48589	0.280384	0.137033
	DE	1.15E-10	0.269149	0.045804	0.055261
	FA	2.42E-13	3.83E-10	9.44E-11	7.93E-11
	FPA	0.00E+00	2.04E-12	1.69E-13	4.08E-13
f_8	E-MFO	**2.23E-14**	**8.39E-11**	**8.31E-12**	**1.47E-11**
	MFO	2.69E+05	5.95E+06	3.06E+06	1.33E+06
	BA	1.40E+06	1.30E+07	5.16E+06	2.26E+06
	BFP	1.19E+07	2.84E+07	2.39E+07	4.19E+06
	DE	2.13E+07	2.84E+07	2.65E+07	1.62E+06
	FA	5.81E+02	1.20E+03	8.14E+02	1.54E+02
	FPA	6.13E+05	1.34E+06	9.55E+05	1.78E+05
f_9	E-MFO	**1.21E-27**	**6.96E-25**	**1.34E-25**	**1.76E-25**
	MFO	1.07E+06	4.79E+08	4.51E+07	7.34E+07
	BA	8.90E+07	1.69E+09	6.68E+08	3.50E+08
	BFP	6.10E+08	3.79E+09	2.24E+09	7.36E+08
	DE	4.19E+07	2.11E+09	7.01E+08	5.31E+08
	FA	1.27E+06	1.42E+07	6.21E+06	3.03E+06
	FPA	5.21E+05	1.03E+07	2.83E+06	1.79E+06
f_{10}	E-MFO	**2.41E-106**	**1.02E-94**	**2.78E-96**	**1.50E-95**
	MFO	9.91E-07	0.002301	8.21E-04	0.002301
	BA	2.11E-08	3.59E-08	6.33E-08	3.59E-08
	BFP	0.013344	0.300021	0.569071	0.300021

	DE	0.630326	0.397031	1.498911	0.397031
	FA	3.62E-08	3.95E-07	4.88E-07	3.95E-07
	FPA	1.36E-09	6.40E-07	1.74E-07	6.40E-07
f_{11}	E-MFO	**9.21E-20**	**1.69E-15**	**1.97E-16**	**3.10E-16**
	MFO	4.44E+03	4.62E+04	1.75E+04	8.26E+03
	BA	1.18E+04	4.45E+04	2.26E+04	7.57E+03
	BFP	6.65E+04	1.48E+05	1.14E+05	2.21E+04
	DE	1.20E+05	1.48E+05	1.34E+05	6.84E+03
	FA	19.98107	2.35E+02	87.81589	49.89147
	FPA	2.86E+03	6.76E+03	4.46E+03	8.10E+02
f_{12}	E-MFO	**4.47E-19**	**2.63E-15**	**1.34E-16**	**4.11E-16**
	MFO	1.31E+03	3.00E+04	1.23E+04	6.64E+03
	BA	99.34773	4.80E+03	1.74E+03	1.17E+03
	BFP	3.66E+04	1.25E+05	8.45E+04	2.02E+04
	DE	6.23E+04	1.34E+05	1.02E+05	1.74E+04
	FA	32.68077	1.38E+02	83.28036	25.51051
	FPA	6.16E+02	1.69E+03	9.65E+02	2.28E+02
f_{13}	E-MFO	**0.00E+00**	**972.5861**	**217.2754**	**373.5995**
	MFO	5.43E+02	9.39E+02	7.52E+02	86.87307
	BA	6.10E-46	4.47E+02	2.65E+02	58.4988
	BFP	1.23E+03	1.58E+03	1.47E+03	88.92216
	DE	1.49E+03	1.74E+03	1.64E+03	58.77398
	FA	1.93E+02	4.24E+02	2.90E+02	51.20952
	FPA	6.45E+02	7.52E+02	6.93E+02	26.56881
f_{14}	E-MFO	20.0320	20.2378	20.1031	0.0488
	MFO	**20.0000**	**21.42653**	**20.00001**	**3.78E-05**
	BA	19.9904	20.00016	19.9995	0.002006
	BFP	21.35306	21.48368	21.41729	0.030592
	DE	20.68334	21.0389	20.85937	0.095717
	FA	20.06513	20.15414	20.10049	0.019191
	FPA	21.32538	21.42653	21.38318	0.022966
f_{15}	E-MFO	**0.00E+00**	**5.21E-15**	**4.26E-16**	**1.10E-15**
	MFO	62.48609	6.01E+02	2.77E+02	12.73271
	BA	3.33E+02	1.60E+03	7.16E+02	2.51E+02
	BFP	1.19E+03	2.66E+03	2.21E+03	4.00E+02
	DE	2.10E+03	2.71E+03	2.47E+03	1.29E+02
	FA	0.026208	0.079403	0.041701	0.010145
	FPA	69.37639	1.14E+02	91.97533	12.73271
f_{16}	E-MFO	**1.52E-06**	**0.1E-02**	**1.00E-03**	**2.1E-03**
	MFO	4.31E+02	1.04E+03	7.53E+02	1.53E+02
	BA	59.82885	3.87E+04	1.68E+03	5.53E+03
	BFP	4.46E+02	1.42E+07	3.08E+05	2.01E+06
	DE	6.51E+02	5.38E+05	3.89E+04	7.79E+04
	FA	2.06E+02	3.82E+02	2.88E+02	33.10453
	FPA	2.18E+02	2.07E+03	7.47E+02	4.32E+02

f_{17}	E-MFO	**8.079067**	**8.861085**	**8.392088**	**0.161567**
	MFO	1.42E+02	4.19E+02	2.13E+02	53.5671
	BA	53.14262	2.14E+02	94.81253	30.86755
	BFP	4.50E+02	1.00E+03	7.91E+02	1.43E+02
	DE	8.24E+02	1.10E+03	9.63E+02	6.12E+01
	FA	2.05E+00	18.36023	8.602651	4.377323
	FPA	33.70335	97.69666	61.72317	14.48923

Table 5.9: Results of E-MFO compared with others in case of dimension size 100

Function	Algorithm	Best	Worst	Average	Std. Dev
f_1	E-MFO	**2.00E-15**	**1.05E-11**	**1.40E-12**	**2.53E-12**
	MFO	1.37E+05	2.71E+05	1.83E+05	3.01E+04
	BA	8.82E+04	2.63E+05	1.50E+05	3.84E+04
	BFP	2.63E+05	6.04E+05	4.88E+05	9.79E+04
	DE	5.36E+05	6.19E+05	5.79E+05	2.08E+04
	FA	0.763145	1.97E+00	1.10E+00	0.235442
	FPA	2.04E+04	3.28E+04	2.48E+04	2.74E+03
f_2	E-MFO	**3.28E-10**	**1.72E-07**	**1.80E-08**	**8.00E-07**
	MFO	4.55E+02	6.97E+02	5.49E+02	53.00042
	BA	1.99E+03	9.03E+62	1.81E+61	1.28E+62
	BFP	9.19E+46	1.70E+101	9.26E+99	3.64E+100
	DE	1.81E+31	2.17E+104	1.82E+103	5.18E+103
	FA	1.23E+02	3.12E+02	1.96E+02	46.32443
	FPA	1.64E+02	2.32E+02	2.00E+02	16.43607
f_3	E-MFO	**2.59E-06**	**2.95E+05**	**4.68E+04**	**9.22E+04**
	MFO	4.81E+05	1.09E+06	7.12E+05	1.51E+05
	BA	7.11E+05	5.26E+06	2.75E+06	1.07E+06
	BFP	2.25E+06	1.15E+07	5.92E+06	1.92E+06
	DE	7.77E+05	4.08E+06	2.21E+06	8.10E+05
	FA	2.50E+05	4.50E+05	3.48E+05	4.36E+04
	FPA	4.92E+04	1.37E+05	8.75E+04	1.86E+04
f_4	E-MFO	**6.45E-06**	**48.9307**	**2.4070**	**9.1517**
	MFO	93.08323	98.52615	96.96512	1.063848
	BA	54.03602	79.41252	68.64042	6.496662
	BFP	78.84859	98.34887	96.44807	3.068331
	DE	96.04051	98.97741	98.15073	0.570873
	FA	42.07106	55.97332	48.4488	3.387015
	FPA	23.74176	36.10634	29.24717	2.63306
f_5	E-MFO	**1.98E+02**	**1.99E+02**	**1.99E+02**	**0.16893**
	MFO	4.44E+10	1.25E+11	7.91E+10	1.86E+10
	BA	1.20E+10	9.91E+10	3.31E+10	1.73E+10
	BFP	6.42E+10	3.69E+11	2.53E+11	9.45E+10
	DE	2.76E+11	3.49E+11	3.26E+11	1.56E+10
	FA	2.25E+03	1.44E+06	1.44E+05	2.84E+05
	FPA	5.47E+08	1.32E+09	8.49E+08	1.97E+08
f_6	E-MFO	**3.60E-05**	**1.80E-04**	**7.97E-04**	**4.56E-05**

	MFO	1.08E+03	2.90E+03	1.86E+03	4.51E+02
	BA	38.92616	1.17E+02	67.6422	20.08162
	BFP	1.59E+03	8.92E+03	6.65E+03	1.82E+03
	DE	7.09E+03	9.14E+03	8.43E+03	5.41E+02
	FA	1.562429	5.726941	3.016414	0.91736
	FPA	9.256986	36.29072	21.15751	5.207852
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.003127	6.46E-05	4.42E-04
	BA	3.13E-13	0.341554	0.101385	0.101006
	BFP	0.00817	0.473123	0.293907	0.129029
	DE	0.00E+00	0.249703	0.061709	0.060307
	FA	1.16E-13	4.15E-10	9.23E-11	9.15E-11
	FPA	0.00E+00	5.66E-11	1.38E-12	7.99E-12
f_8	E-MFO	**1.73E-12**	**1.07E-07**	**1.50E-08**	**2.76E-08**
	MFO	1.26E+07	2.28E+07	1.78E+07	2.60E+06
	BA	5.56E+06	1.93E+07	1.11E+07	2.95E+06
	BFP	2.60E+07	6.11E+07	4.57E+07	1.13E+07
	DE	5.17E+07	6.11E+07	5.76E+07	2.22E+06
	FA	4.33E+03	1.05E+04	6.46E+03	1.28E+03
	FPA	1.92E+06	2.99E+06	2.45E+06	2.70E+05
f_9	E-MFO	**6.38E-28**	**1.02E-23**	**3.19E-25**	**1.44E-24**
	MFO	2.47E+06	2.45E+08	5.32E+07	5.67E+07
	BA	1.80E+08	2.64E+09	8.23E+08	5.10E+08
	BFP	6.70E+08	3.83E+09	2.30E+09	8.01E+08
	DE	2.25E+07	2.90E+09	9.50E+08	7.15E+08
	FA	1.53E+06	1.62E+07	5.66E+06	2.89E+06
	FPA	1.02E+06	9.69E+06	3.34E+06	1.84E+06
f_{10}	E-MFO	**1.1E-105**	**4.06E-89**	**8.14E-91**	**5.74E-90**
	MFO	1.43E-04	0.994845	0.059233	0.174417
	BA	1.70E-08	1.38E-07	5.70E-08	2.94E-08
	BFP	0.080006	1.845873	0.770675	0.43964
	DE	1.059435	2.630292	1.85E+00	0.419724
	FA	4.49E-08	2.35E-06	7.90E-07	5.14E-07
	FPA	7.24E-09	2.73E-05	9.47E-07	3.96E-06
f_{11}	E-MFO	**5.49E-15**	**1.06E-10**	**6.28E-12**	**1.75E-11**
	MFO	1.04E+05	2.30E+05	1.63E+05	2.66E+04
	BA	4.51E+04	2.16E+05	1.08E+05	3.54E+04
	BFP	2.44E+05	5.93E+05	5.15E+05	9.43E+04
	DE	4.90E+05	6.21E+05	5.68E+05	2.89E+04
	FA	7.65E+02	3.17E+03	1.58E+03	5.40E+02
	FPA	1.67E+04	2.97E+04	2.32E+04	3.37E+03
f_{12}	E-MFO	**1.83E-16**	**3.24E-11**	**8.12E-13**	**4.58E-12**
	MFO	2.33E+04	6.19E+04	3.93E+04	7.77E+03
	BA	1.70E+03	2.21E+04	8.02E+03	4.25E+03
	BFP	6.38E+04	2.77E+05	2.05E+05	5.66E+04

	DE	1.93E+05	3.19E+05	2.55E+05	2.40E+04
	FA	1.58E+02	5.08E+02	3.31E+02	87.21924
	FPA	2.04E+03	3.80E+03	2.89E+03	3.83E+02
f_{13}	E-MFO	**0.00E+00**	**1.68E-10**	**7.21E-12**	**2.47E-11**
	MFO	1.74E+03	2.15E+03	1.95E+03	89.58739
	BA	3.15E+02	9.29E+02	5.00E+02	1.45E+02
	BFP	2.48E+03	3.37E+03	3.13E+03	2.07E+02
	DE	3.21E+03	3.53E+03	3.41E+03	69.91233
	FA	6.24E+02	9.51E+02	7.79E+02	81.36572
	FPA	1.51E+03	1.68E+03	1.60E+03	40.86178
f_{14}	E-MFO	20.1485	20.4737	20.3551	0.0671
	MFO	20.00000	20.09454	20.01554	0.023989
	BA	**19.99425**	**20.00086**	**20.00017**	**0.001098**
	BFP	21.46023	21.55383	21.50665	0.020649
	DE	20.89783	21.24759	21.07666	0.077417
	FA	20.16807	20.28109	20.22424	0.026702
	FPA	21.43279	21.5232	21.49185	0.018635
f_{15}	E-MFO	**0.00E+00**	**1.85E-11**	**1.29E-12**	**3.27E-12**
	MFO	1.27E+03	2.16E+03	1.66E+03	2.13E+02
	BA	6.86E+02	2.33E+03	1.41E+03	3.16E+02
	BFP	1.81E+03	5.56E+03	4.61E+03	1.07E+03
	DE	4.51E+03	5.56E+03	5.16E+03	2.15E+02
	FA	0.14324	0.306529	0.207294	0.031399
	FPA	1.54E+02	3.25E+02	2.18E+02	28.88742
f_{16}	E-MFO	**2.8E-03**	**19.60556**	**1.573378**	**3.167623**
	MFO	5.63E+03	7.16E+03	6.22E+03	3.78E+02
	BA	1.77E+04	2.14E+08	5.74E+06	3.09E+07
	BFP	6.40E+04	4.17E+15	3.22E+14	8.44E+14
	DE	7.47E+04	7.37E+07	9.55E+06	1.26E+07
	FA	3.07E+03	6.30E+03	4.57E+03	6.68E+02
	FPA	2.08E+04	2.92E+05	8.67E+04	5.87E+04
f_{17}	E-MFO	**17.12754**	**17.66778**	**17.59107**	**0.201477**
	MFO	5.48E+02	1.01E+03	7.60E+02	1.02E+02
	BA	1.28E+02	5.63E+02	3.03E+02	93.40166
	BFP	7.78E+02	2.29E+03	1.83E+03	3.64E+02
	DE	1.87E+03	2.32E+03	2.11E+03	1.08E+02
	FA	30.04823	1.26E+02	69.30709	19.67717
	FPA	1.05E+02	1.96E+02	1.52E+02	22.08205

Table 5.10: Results of E-MFO compared with other algorithms in case of dimension size 200

Function	Algorithm	Best	Worst	Average	Std. Dev
f_1	E-MFO	**4.87E-14**	**1.05E-08**	**1.01E-09**	**2.11E-09**
	MFO	8.62E+05	1.03E+06	9.50E+05	3.53E+04
	BA	2.07E+05	7.80E+05	3.82E+05	1.08E+05
	BFP	7.65E+05	1.59E+06	1.33E+06	2.68E+05
	DE	1.44E+06	1.59E+06	1.53E+06	3.27E+04

	FA	3.13E+04	8.22E+04	4.82E+04	1.07E+04
	FPA	6.04E+04	8.35E+04	6.95E+04	6.56E+03
f_2	E-MFO	**1.63E-10**	**5.66E-06**	**2.17E-07**	**8.00E-07**
	MFO	2.01E+03	2.48E+03	2.25E+03	97.77479
	BA	3.54E+08	1.10E+208	2.20E+206	Inf
	BFP	9.20E+140	5.60E+268	1.12E+267	Inf
	DE	2.06E+148	3.72E+270	1.47E+269	Inf
	FA	9.44E+02	1.45E+03	1.16E+03	89.70169
	FPA	4.36E+02	5.86E+02	5.13E+02	34.11001
f_3	E-MFO	**0.023467**	**2.12E+06**	**7.31E+05**	**7.33E+05**
	MFO	2.30E+06	5.50E+06	3.66E+06	8.38E+05
	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
f_4	E-MFO	**48.81717**	**51.10278**	**49.49483**	**0.39221**
	MFO	98.22465	99.4224	98.95586	0.289448
	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	39.6108	6.61E+01	4.83E+01	5.29E+00
f_5	E-MFO	**4.98E+02**	**4.99E+02**	**4.99E+02**	**0.085523**
	MFO	4.34E+11	5.59E+11	4.99E+11	2.83E+10
	BA	3.07E+10	1.97E+11	8.51E+10	3.88E+10
	BFP	2.02E+11	9.28E+11	7.20E+11	2.42E+11
	DE	7.76E+11	9.27E+11	8.76E+11	3.16E+10
	FA	1.11E+09	7.73E+09	3.98E+09	1.44E+09
	FPA	1.48E+09	4.32E+09	2.85E+09	5.82E+08
f_6	E-MFO	**7.81E-05**	**0.004267**	**1.20E-04**	**8.55E-04**
	MFO	2.50e+04	3.75e+04	3.06e+04	2.40e+03
	BA	2.00E+02	3.11E+03	9.46E+02	5.95E+02
	BFP	1.59E+02	6.09E+04	7.07E+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
f_7	E-MFO	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	MFO	0.00E+00	0.002964	5.93E-05	4.19E-04
	BA	5.13E-14	0.342796	0.081967	0.094435
	BFP	0.043544	0.481134	0.305825	0.120609
	DE	0.00E+00	0.355643	0.059003	0.074542
	FA	3.27E-12	4.96E-10	7.87E-11	9.22E-11
	FPA	0.00E+00	6.13E-12	3.07E-13	9.32E-13
f_8	E-MFO	**1.60E-09**	**2.90E-04**	**2.38E-05**	**5.51E-05**

	MFO	8.89E+07	1.06E+08	9.59E+07	3.89E+06
	BA	1.80E+07	6.24E+07	3.33E+07	1.04E+07
	BFP	7.27E+07	1.59E+08	1.37E+08	2.59E+07
	DE	1.45E+08	1.58E+08	1.54E+08	2.83E+06
	FA	2.95E+06	6.93E+06	4.80E+06	9.82E+05
	FPA	5.86E+06	8.71E+06	6.91E+06	6.09E+05
f_9	E-MFO	**2.32E-28**	**8.79E-25**	**1.24E-25**	**1.98E-25**
	MFO	7.24E+05	3.67E+08	4.41E+07	6.58E+07
	BA	1.73E+08	1.86E+09	8.07E+08	4.72E+08
	BFP	1.33E+09	3.89E+09	2.53E+09	6.65E+08
	DE	9.27E+06	3.59E+09	1.26E+09	1.03E+09
	FA	1.05E+06	1.22E+07	5.58E+06	2.56E+06
	FPA	6.82E+05	7.11E+06	2.77E+06	1.48E+06
f_{10}	E-MFO	**8.38E-99**	**2.59E-82**	**5.28E-84**	**3.67E-83**
	MFO	0.038239	1.363739	0.336057	0.290076
	BA	1.24E-08	1.67E-07	6.22E-08	3.99E-08
	BFP	0.121382	1.916926	0.967956	0.533964
	DE	1.525914	3.777572	2.575341	0.529162
	FA	6.10E-07	0.94647	0.106643	0.159758
	FPA	5.43E-09	6.24E-06	7.76E-07	1.21E-06
f_{11}	E-MFO	**1.13E-11**	**6.81E-18**	**4.28E-09**	**1.15E-15**
	MFO	1.93E+06	2.48E+06	2.19E+06	1.17E+05
	BA	3.37E+05	1.24E+06	8.14E+05	2.02E+05
	BFP	1.12E+06	3.93E+06	3.31E+06	6.95E+05
	DE	3.56E+06	3.99E+06	3.80E+06	9.44E+04
	FA	9.43E+04	1.77E+05	1.31E+05	2.19E+04
	FPA	1.37E+05	2.33E+05	1.73E+05	1.93E+04
f_{12}	E-MFO	**1.13E-13**	**5.65E-10**	**9.20E-11**	**1.36E-10**
	MFO	1.61E+05	2.85E+05	1.95E+05	2.26E+04
	BA	1.05E+04	1.19E+05	3.83E+04	2.19E+04
	BFP	1.41E+05	7.78E+05	5.59E+05	2.09E+05
	DE	6.33E+05	8.38E+05	7.48E+05	4.59E+04
	FA	3.12E+03	9.48E+03	5.81E+03	1.29E+03
	FPA	6.67E+03	1.13E+04	9.05E+03	1.12E+03
f_{13}	E-MFO	**0.00E+00**	**1.59E-08**	**7.93E-10**	**2.41E-09**
	MFO	6.03E+03	6.80E+03	6.47E+03	1.68E+02
	BA	1.14E+03	2.49E+03	1.65E+03	2.84E+02
	BFP	6.62E+03	8.65E+03	8.14E+03	6.29E+02
	DE	8.49E+03	8.98E+03	8.78E+03	1.16E+02
	FA	2.78E+03	3.37E+03	3.09E+03	1.47E+02
	FPA	4.42E+03	4.78E+03	4.58E+03	77.79587
f_{14}	E-MFO	20.4956	20.7528	20.6658	0.0533
	MFO	20.00000	21.59477	20.13839	0.151432
	BA	20.11317	20.25831	20.18865	0.038652
	BFP	21.54223	21.61217	21.58754	0.013446

	DE	21.24674	21.38999	21.32366	0.034576
	FA	20.48476	20.62969	20.54637	0.029272
	FPA	**21.55964**	**21.59476**	**21.58079**	**0.008826**
f_{15}	E-MFO	**0.00E+00**	**8.54E-10**	**9.47E-11**	**1.65E-10**
	MFO	7.69E+03	9.56E+03	8.67E+03	3.79E+02
	BA	2.03E+03	6.78E+03	3.82E+03	9.95E+02
	BFP	6.06E+03	1.43E+04	1.27E+04	2.24E+03
	DE	1.32E+04	1.42E+04	1.38E+04	2.31E+02
	FA	2.33E+02	6.16E+02	4.32E+02	87.71168
	FPA	5.19E+02	8.14E+02	6.40E+02	63.67595
f_{16}	E-MFO	**8.745768**	**2.57+03**	**252.4219**	**408.4412**
	MFO	1.55E+04	7.31E+02	1.75E+04	7.31E+02
	BA	3.00E+04	1.22E+08	3.19E+07	1.22E+08
	BFP	1.03E+10	1.95E+18	1.15E+18	1.95E+18
	DE	5.50E+06	6.34E+08	4.42E+08	6.34E+08
	FA	1.25E+04	1.66E+03	1.66E+04	1.66E+03
	FPA	1.50E+05	1.27E+06	1.22E+06	1.27E+06
f_{17}	E-MFO	**44.30506**	**45.28934**	**44.83048**	**0.246678**
	MFO	3.10E+03	3.89E+03	3.47E+03	1.80E+02
	BA	6.74E+02	2.05E+03	1.11E+03	3.05E+02
	BFP	2.52E+03	5.81E+03	4.89E+03	9.82E+02
	DE	5.40E+03	6.11E+03	5.76E+03	1.55E+02
	FA	3.51E+02	6.76E+02	5.21E+02	79.57762
	FPA	3.45E+02	5.82E+02	4.62E+02	49.42061

Table 5.11: Results of E-MFO compared with others in case of dimension size 500

Inference: The effect of dimension is evaluated on five different sizes which are 30, 50, 100, 200 and 500. It is concluded from results that E-MFO outperforms for all dimension sizes. In case of 50-dimension size, E-MFO performed better in fifteen functions, FA and MFO in only one. For 30 and 100-dimension size, E-MFO outperforms in all functions except of one function. That one function provides good results in FA. In case of 200 dimension, E-MFO provides good results in sixteen functions and BA in one function. The same trend is follow for dimension 500 except for BA, its one function come up with good results for FPA. The results show that with increase in dimension size results degrade but even then, E-MFO manages to perform better than other algorithms.

5.2.3 Statistical testing

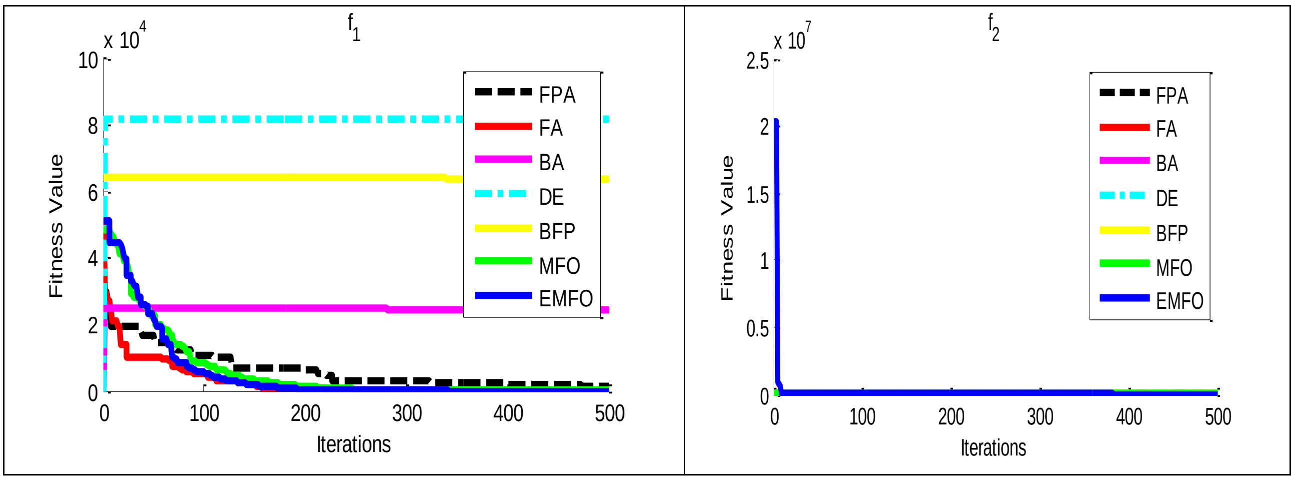
The statistical testing includes the Wilcoxon's rank-sum test [141] which tests the performance of two different algorithms. Here, the performance E-MFO is checked with respect to other nature inspired algorithms. Basically, this test is used to find the difference among two algorithms and at the end of this test we get p-value as an output. This p-value signifies the impact factor of algorithm under test. If p-value is <0.05, then an algorithm is considered as statically significant. The results show in Table 9 that E-MFO performs better in seventeen functions, FA in one and MFO in two functions. The NA value of any algorithm shows the algorithm's superiority over others and other algorithm finds its p-value with respect to that algorithm. Even statistical results prove that E-MFO is better algorithm than others.

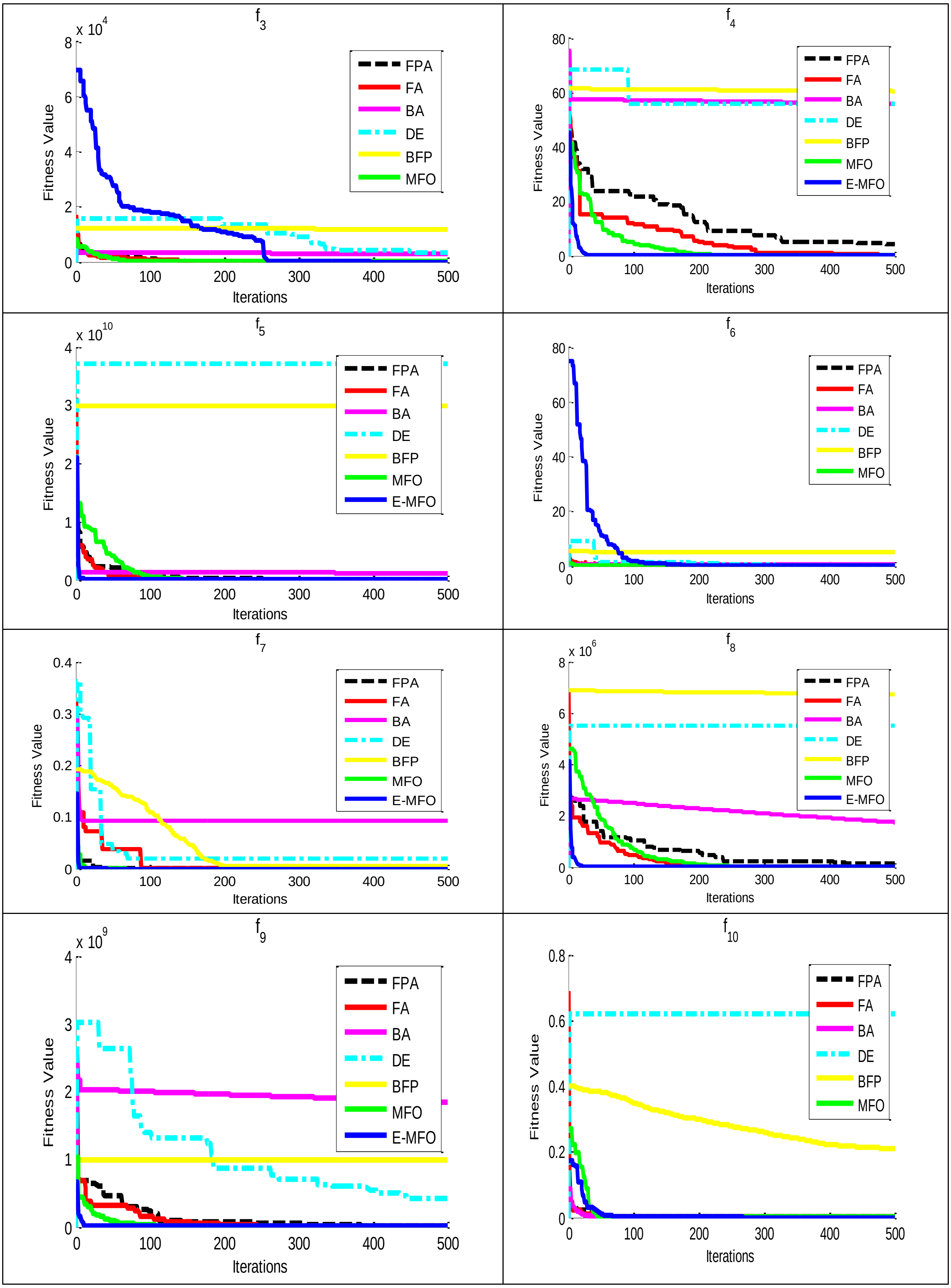
Function	BA	BFP	DE	FA	FPA	MFO	E-MFO
f_1	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_2	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	5.55e-012	NA
f_3	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_4	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_5	7.06e-018	7.06e-018	7.06e-018	5.05e-012	7.06e-018	7.06e-018	NA
f_6	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.96e-018	NA
f_7	3.31e-020	3.31e-020	1.25e-019	3.31e-020	5.96e-018	0.0065	NA
f_8	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_9	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_{10}	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_{11}	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_{12}	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_{13}	0.0408	6.62e-018	3.72e-016	1.49e-004	0.1818	0.0066	NA
f_{14}	0.5510	7.06e-018	7.06e-018	0.0044	7.06e-018	0.1691	NA
f_{15}	3.31e-020	3.31e-020	3.31e-020	3.31e-020	3.31e-020	3.31e-020	NA
f_{16}	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	7.06e-018	NA
f_{17}	7.06e-018	7.06e-018	7.06e-018	NA	7.06e-018	7.06e-018	7.06e-018
f_{18}	4.49e-016	7.06e-018	3.52e-017	1.80e-016	1.34e-016	5.62e-017	NA
f_{19}	3.31e-020	3.31e-020	4.24e-009	3.31e-020	3.31e-020	NA	3.31e-020
f_{20}	5.52e-018	5.52e-018	0.1540	5.52e-018	4.64e-006	NA	0.3336

Table 5.12 Statistical testing

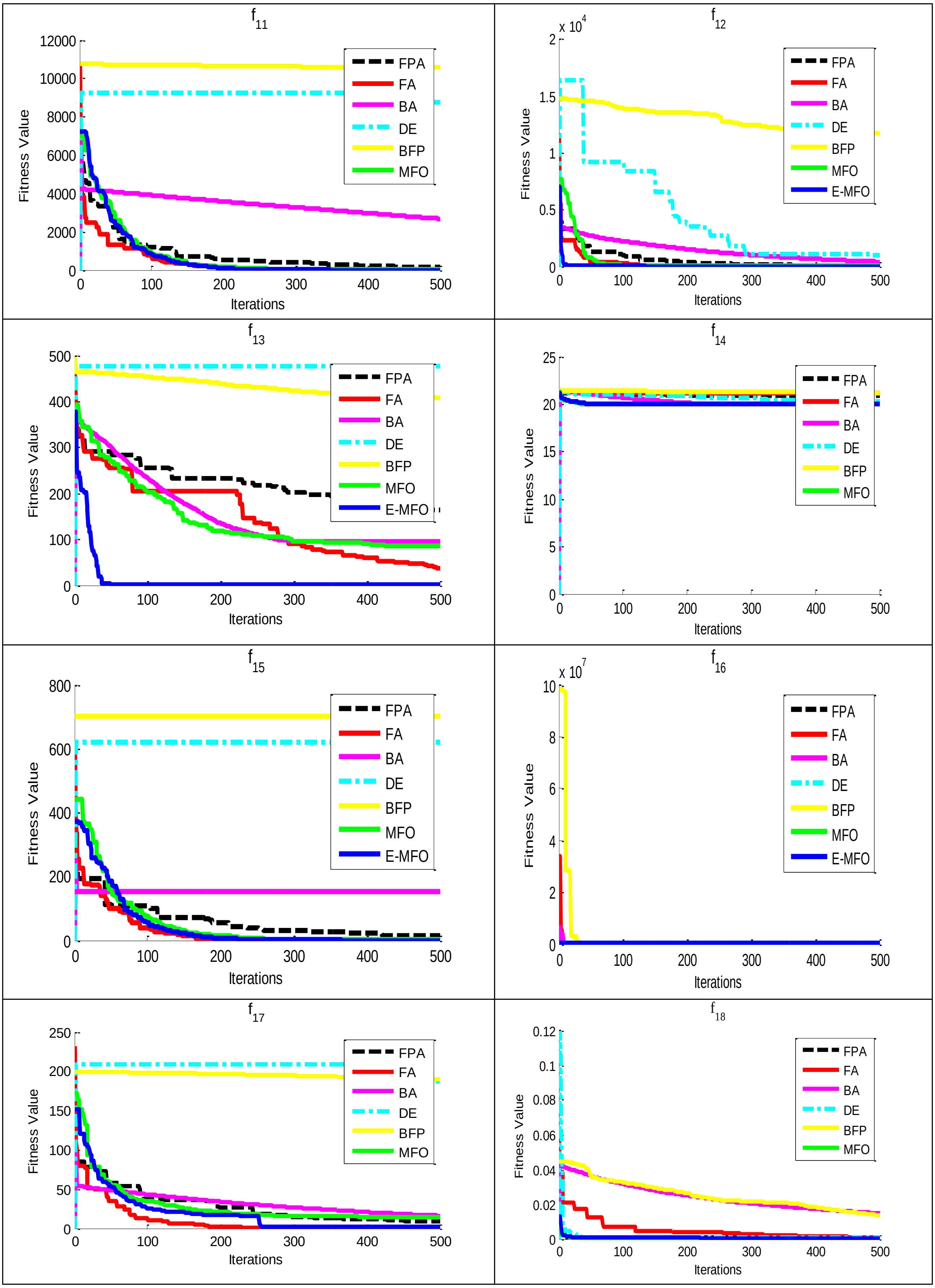
5.2.4 Convergence graphs

The convergence graphs for all algorithms are drawn and compared to E-MFO in order to check the speed of algorithm. It is observed from Figure 5.1 that E-MFO is able to achieve global minima in less number of iterations and has fast convergence rate in comparison to other algorithms.





f3
x 10^4
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
f4
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
E-MFO
f5
x 10^10
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
E-MFO
f6
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
f7
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
E-MFO
f8
x 10^6
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
E-MFO
f9
x 10^9
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO
E-MFO
f10
Fitness Value
Iterations
FPA
FA
BA
DE
BFP
MFO


f11
Fitness Value
Iterations
12000
10000
8000
6000
4000
2000
0
100
200
300
400
500
FPA
FA
BA
DE
BFP
MFO
f12
x 10^4
2
1.5
1
0.5
E-MFO
f13
f14
25
20
15
10
5
f15
800
600
400
200
f16
x 10^7
f17
250
150
50
f18
0.12
0.1
0.08
0.06
0.04
0.02

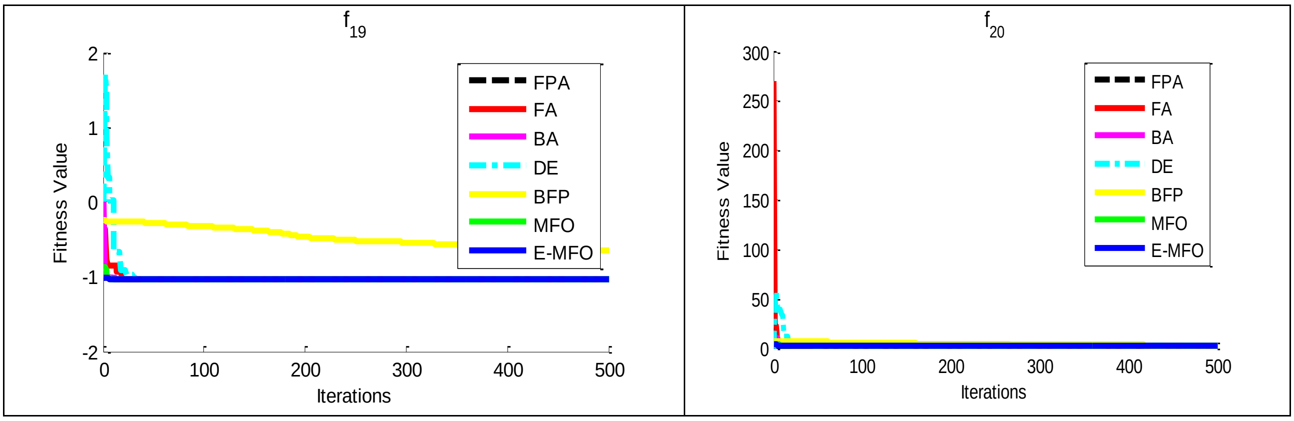


Figure 5.1 Convergence graphs for all benchmark functions

5.3 SYNTHESIS OF ANTENNA ARRAY WITH E-MFO

For synthesizing the linear antenna array with E-MFO, population size is taken as 100, maximum iterations equal to 500 and number of runs are 20. Various cases have been considered to analyze the performance of proposed approach on antenna array.

5.3.1 Optimization of amplitude excitation

For designing LAA, three cases are considered such as 10, 16 and 24 element arrays. For each case, optimized amplitude excitations and SLL is measured and compared with other algorithms. The radiation pattern is also drawn to check efficiency of proposed approach in comparison to others.

Case 1: 10-element array

In first example, a 10-element array is taken (2N=10). Hence, only five elements need to be optimized. Here, only amplitude excitations are varied, other antenna parameters are kept constant. The aim of optimization is to reduce SLL only. So, in fitness function (4.4), k_1=1 and k_2=0 are taken, to achieve maximum side lobe reduction. The SLL is reduced in the range of [104° 180°] and amplitude excitations are allowed to vary in range of 0 to 1. The optimized current amplitudes are shown in Table 5.13. From Table 5.14, it is clear that E-MFO provides minimum side lobes in comparison to other algorithms. The peak SLL provided by proposed approach is -26.66 dB which is -13.43 dB lower than the conventional method. The results of E-MFO are compared with results provided by other algorithms like SADE [123], PSO [119], TM [123], GWO [137], Chebyshev pattern [142] and ALO [136]. The radiation pattern is given in Figure 5.2 which shows the comparison of E-MFO with PSO, ALO and GWO. The normalized amplitude distribution for the optimized antenna is shown in Figure 5.3.

Optimization Method	Optimized Current Amplitudes				
E-MFO	1.0000	0.8918	0.7036	0.4786	0.3398
Conv	1.0000	1.0000	1.0000	1.0000	1.0000
ALO [136]	1.0000	0.8959	0.6957	0.4935	0.2966
GWO [137]	1.0000	0.8962	0.6963	0.4935	0.2964
PSO [119]	1.0000	0.9010	0.7255	0.5120	0.4088
Cheby [142]	1.0000	0.9010	0.7255	0.5119	0.4088
TM [123]	1.0000	0.8999	0.7228	0.5077	0.3994
SADE [123]	1.0000	0.9028	0.7277	0.5153	0.4158

Table 5.13 Optimized amplitude excitation for 10-element LAA

Optimization Algorithms	Conv	SADE [123]	PSO [119]	Cheby [142]	TM [123]	GWO [137]	ALO [136]	E-MFO
SLL (dB)	-13.23	-24.41	-24.62	-24.62	-24.88	-26.05	-26.08	-26.66

Table 5.14 Comparison of peak SLL for 10-element array

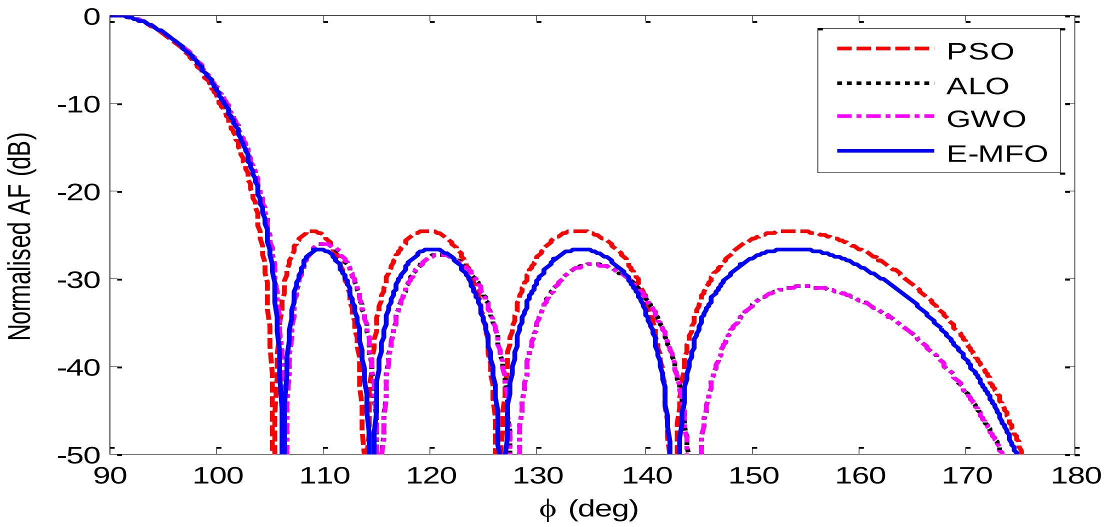


Figure 5.2 Optimized radiation pattern of 10- element antenna array

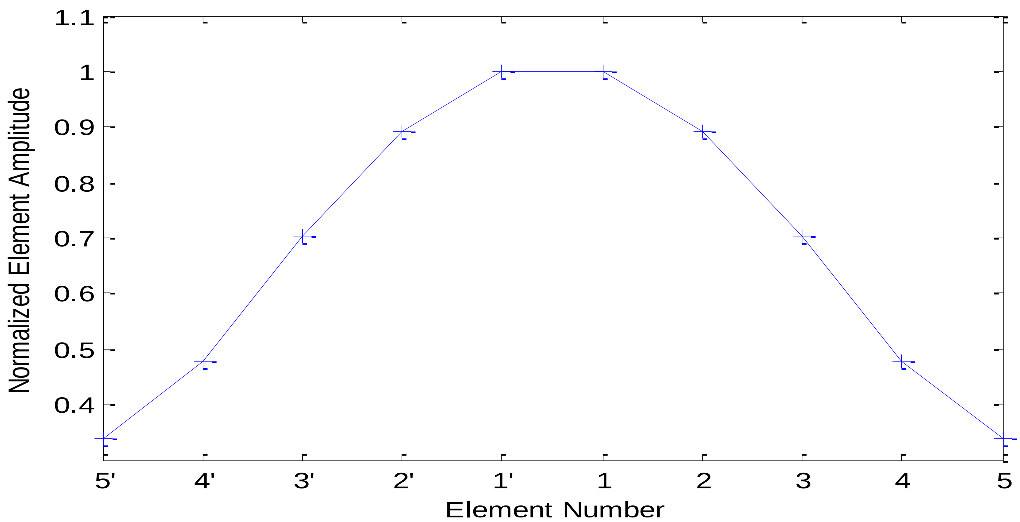


Figure 5.3 Normalized amplitude distribution for 10-element LAA

Case 2: 16-element array

In this example, a 16-element linear antenna array is taken for optimization for the same aim as in the above example. Due to symmetric nature of antenna array, only eight elements to be optimized. The amplitude coefficients and peak SLL comparison of E-MFO with ALO [136], TS [117], PSO [119], SADE [123], TM [123], MSMO [138] and Chebyshev pattern [142] are given in Table 5.15 and 5.16 respectively. The results show that E-MFO provides minimum SLL in comparison to other algorithms. E-MFO provides peak SLL value of -34.95 dB. This is -21.72 dB lower than uniform array. The Figure 5.4 shows the plot of optimized amplitude distribution for 16- element antenna array. The radiation pattern of E-MFO is compared with PSO, ALO and SADE as shown in Figure 5.5

Optimization Method	**Optimized Current Amplitudes**							
E-MFO	1.0000	0.9423	0.8399	0.6961	0.5454	0.3836	0.2514	0.1729
Conv	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
ALO [136]	1.0000	0.9344	0.8521	0.7044	0.6000	0.4000	0.3003	0.2002
TS [117]	1.0000	0.9627	0.8766	0.7560	0.6105	0.4833	0.2957	0.3426
PSO [119]	1.0000	0.9521	0.8605	0.7372	0.5940	0.4465	0.3079	0.2724
Cheby [142]	1.0000	0.9515	0.8602	0.7364	0.5933	0.4457	0.3069	0.2713
SADE [123]	1.0000	0.9515	0.8586	0.7333	0.5889	0.4404	0.3020	0.2616
TM [123]	1.0000	0.9500	0.8575	0.7317	0.5861	0.4381	0.2988	0.2552
MSMO [138]	1.0000	0.9613	0.7249	0.8346	0.5556	0.3977	0.2842	0.1844

Table 5.15 Optimized amplitude excitation for 16-element LAA

Optimization Method	**Conv**	**TS [117]**	**PSO [119]**	**Cheby [142]**	**ALO [136]**	**SADE [123]**	**TM [123]**	**MSMO [138]**	**E-MFO**
SLL (dB)	-13.23	-26.2	-30.7	-30.7	-30.87	-31.06	-31.21	-33.24	-34.95

Table 5.16 Comparison of peak SLL for 16-element array

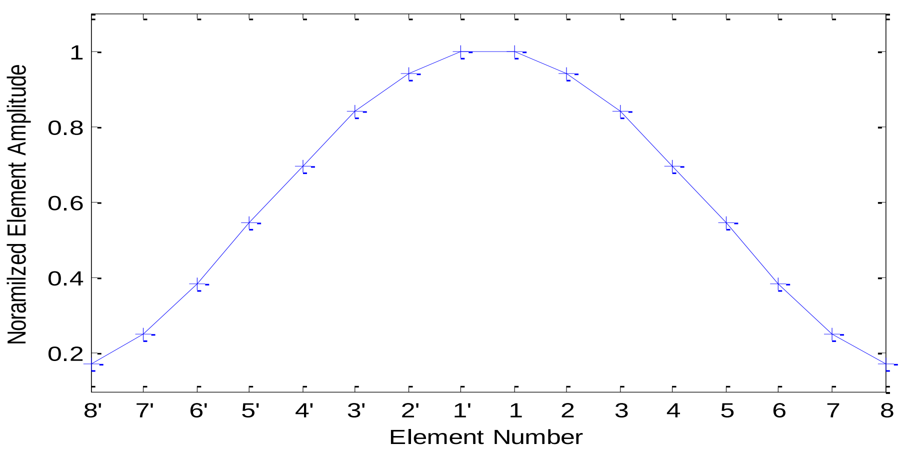


Figure 5.4 Amplitude distribution for 16-element LAA

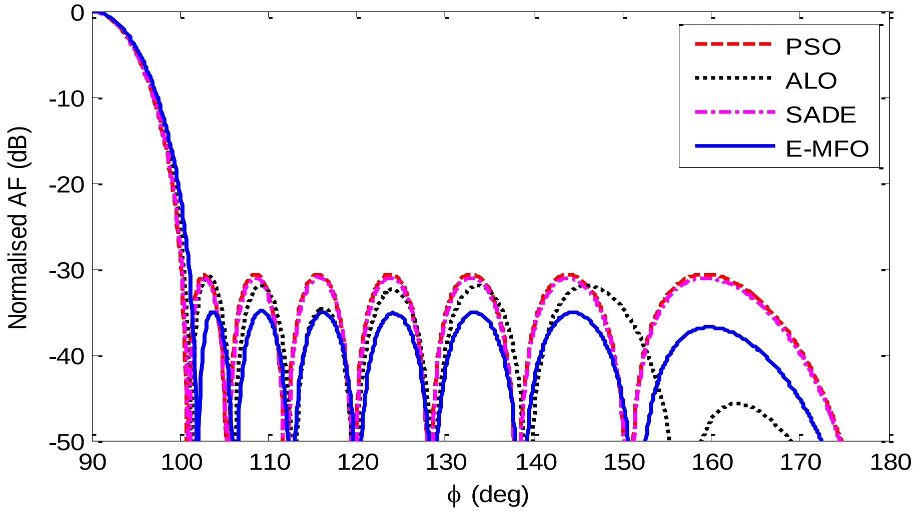


Figure 5.5 Optimized radiation pattern for 16-element LAA

Case 3: 24-element antenna arrays

In this example, geometry configuration is 2N=24 and only 12 elements are to be optimized with help of E-MFO. The optimized current amplitudes and peak SLL are given in Table 5.17 and 5.18 respectively. E-MFO provides minimum SLL at -36.75 dB which is -23.52 dB lower than uniform array. The TM [123] method and the SADE method [123] are competitive to E-MFO but E-MFO outperforms other algorithms (TS [117], PSO [119], CS [143], Chebyshev [142]). Table 5.17 shows the optimized excitation amplitude obtained by E-MFO and by other algorithms. The radiation patterns of E-MFO, CS [143], PSO [119] and SADE [123] are shown in Figure 5.6. Figure 5.7 shows the normalized amplitude pattern for 24- element antenna array.

Method	**Optimized Current Amplitudes**					
E-MFO	1.0000	0.9841	0.9115	0.8629	0.7399	0.6579
	0.5361	0.4356	0.3383	0.2381	0.1703	0.1187
Conv	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
CS [143]	1.0000	0.9773	0.9281	0.8573	0.7753	0.6854
	0.5767	0.4684	0.3836	0.2749	0.2227	0.2000
TS [117]	1.0000	0.9811	0.9373	0.8850	0.7883	0.7294
	0.5984	0.5319	0.4051	0.3381	0.2123	0.3197
PSO [119]	1.0000	0.9712	0.9226	0.8591	0.7812	0.6807
	0.5751	0.4768	0.3793	0.2878	0.2020	0.2167
Cheby	1.0000	0.9758	0.9289	0.8619	0.7789	0.6839
[142]	50.824	0.4794	0.3795	0.2870	0.2049	0.2225
SADE	1.0000	0.9747	0.9272	0.8584	0.7735	0.6775
[123]	0.5743	0.4712	0.3701	0.2781	0.1972	0.2053
TM [123]	1.0000	0.9731	0.9283	0.8585	0.7745	0.6758
	0.5772	0.4686	0.3719	0.2764	0.1995	0.2026
BBO [125]	1.0000	0.9765	0.9270	0.8581	0.7749	0.6750
	0.5757	0.4671	0.3716	0.2720	0.2003	0.2001

Table 5.17 Optimized amplitude excitation for 24-element LAA

Optimization Method	Conv	TS [117]	TM [123]	PSO [119]	SADE [123]	CS [143]	Cheby [142]	E-MFO
SLL (dB)	-13.23	-27.54	-35.25	-34.50	-35.21	-34.83	-34.50	-36.75

Table 5.18 Comparison of peak SLL for 24-element array

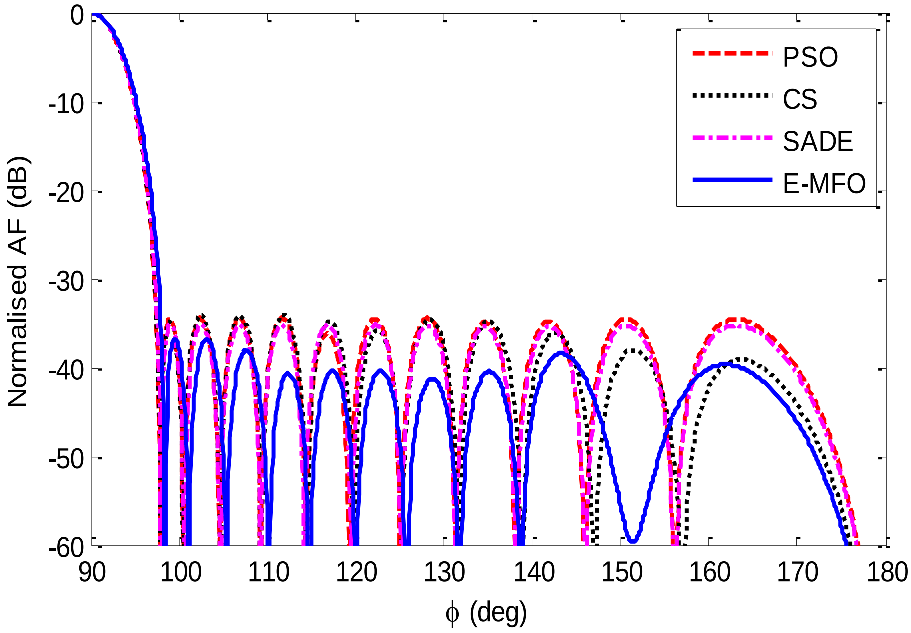


Figure 5.6 Optimized radiation pattern for 24-element LAA

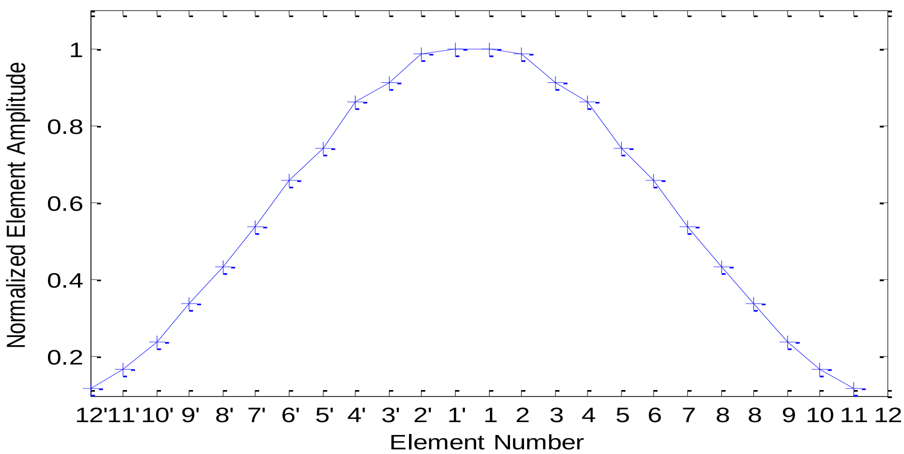


Figure 5.7 Normalized amplitude pattern for 24- element LAA

5.3.2 Optimization for null placement

The E-MFO algorithm is also employed for steering single, double and triple nulls in desired direction. In this work, 20-element antenna array is considered with uniform inter-element spacing. This experiment is conducted only by controlling the amplitude excitations. The values of k_1 and k_2 in the fitness equation (4.4) are taken as 20 and 1 respectively, as is taken in [133].

Case 1: Unidirectional null steering

In this example, a single null is imposed in 104° direction. The optimum amplitude excitations are shown in Table 5.19. The results of E-MFO are compared with algorithms in Table 5.20. The null depth level (NDL) and SLL are the key parameters to check the effectiveness of proposed algorithm. The results show that E-MFO provide better SLL in comparison to HSA [135], BA [129], MTACO [132], PGSA [131], BSA [134], SOA [139] and BFA [130]. The NDL provided by E-MFO, in direction of 104°, is -195.33 dB which is minimum among all algorithms. Any value lesser than -60 dB is considered as null. The Figure 5.8 shows the radiation patterns of E-MFO, BSA and HSA for LAA with null at 104°.

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
E-MFO	0.9856	1.0000	0.9603	0.8679	.7040	0.6039	0.4667	0.3292	0.2293	0.2101
HSA	1.0000	0.9782	0.9828	0.8716	0.7547	0.6296	0.5000	0.3425	0.2427	0.2713
BSA	1.0000	0.9871	0.9498	0.8816	0.7712	0.6394	0.4829	0.3306	0.2588	0.2950

Table 5.19 Optimized amplitude excitations for 20-element LAA with null at 104°

Parameters	**MSL (dB)**	**NDL (dB)**
HSA [135]	-29.14	-122.5
MTACO [132]	-29.10	-88.1
BA [129]	-28.80	-100.4
PGSA [131]	-28.03	-118.3
BSA [134]	-29.22	-134.2
BFA [130]	-28.11	-113.6
SOA [139]	-28.08	-103.90
E-MFO	-30.24	-195.33

Table 5.20 Comparison of different algorithms for 20-element LAA with null at 104°

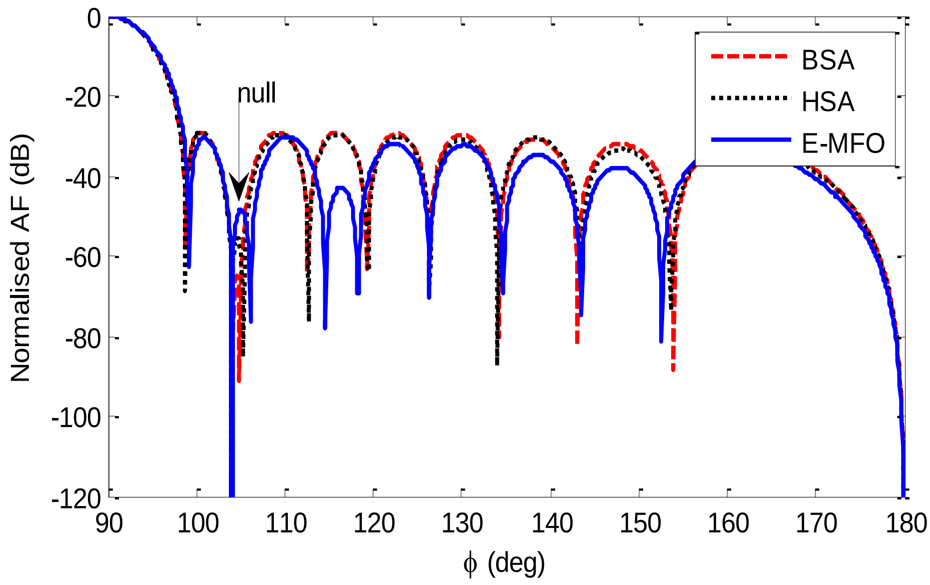


Figure 5.8 Radiation pattern comparison of E-MFO, BSA and HAS

Case 2: Bidirectional null steering

In this case, two nulls are imposed in different directions for 20-element array. The desired directions for null placement are 104° and 116°. Table 5.21 shows that SLL provided by E-MFO is better than other algorithms. But, E-MFO provides higher value of null depth level. In comparison to BSA, SLL provided by E-MFO is -1.45 dB lower. On the other hand, NDL provided by BSA is -33 dB lower than E-MFO. There is always a trade-off between NDL and MSL. To get desired results, antenna designer must sacrifice one of the parameter. But, any value less than -60 dB is considered as a null. Hence the results of E-MFO are better in terms of SLL and NDL. The radiation pattern of E-MFO along with BSA [134] and HSA [135] is shown in Figure 5.9. The optimized antenna excitations are shown in Table 5.22.

Parameter	BSA [134]	HSA [135]	BBO [125]	SOA [139]	E-MFO
NDL (104°)	-131.2	-125	-101.1	-95.54	-98.73
NDL (116°)	-130.7	-135	-101	-134.40	-91.01
MSL (dB)	-28.55	-28.34	-26.8	-27.51	-30

Table 5.21 Comparison of different algorithms for 20-element LAA with null at 104° and 116°

Element	1	2	3	4	5	6	7	8	9	10
E-MFO	0.9827	1.0000	0.9270	0.8428	0.6986	0.5894	0.4819	0.3415	0.1746	0.2052
HSA	1.0000	0.9874	1.023	0.8813	0.7222	0.6510	0.5091	0.3805	0.2070	0.2694
BSA	0.8437	0.8543	0.8507	0.7423	0.6098	0.5508	0.4405	0.3053	0.1743	0.2239

Table 5.22 Optimized amplitude excitations with nulls at 104° and 116°

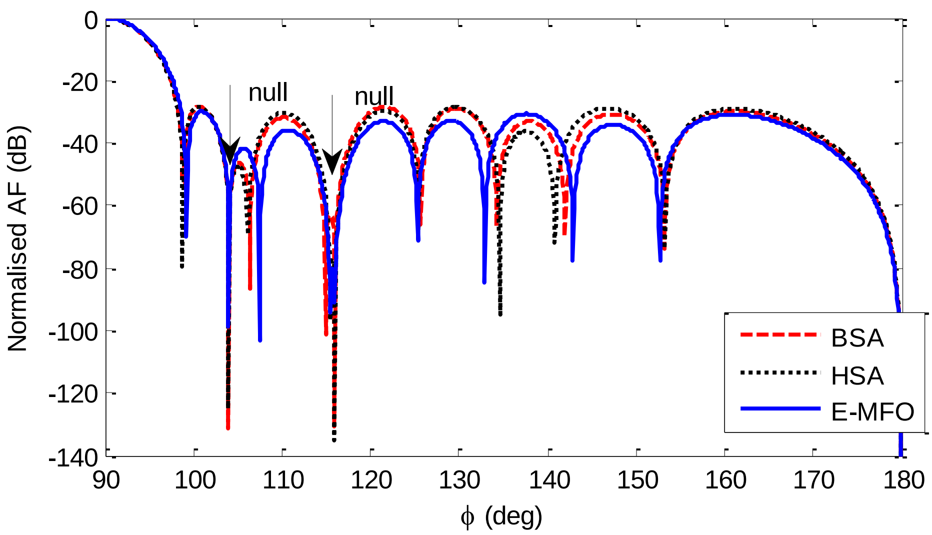


Figure 5.9 Radiation pattern for LAA where multiple nulls are imposed at 104° and 116°

Case 3: Multidirectional null steering

In this example, E-MFO is used to impose in directions 104°, 116° and 123° for a 20-element linear antenna array. Figure 5.10 shows the radiation pattern for triple nulls. The SLL and NDL comparison of various algorithms is done in Table 5.23. The amplitude excitations of E-MFO, BSA and HSA are given in Table 5.24. It is clear from the Figure 5.10 that all the desired nulls are lower than -60 dB.

Parameters	**BSA [134]**	**HSA [135]**	**SOA [139]**	**E-MFO**
SLL (dB)	-27.14	-27.37	-25.03	-28.73
NDL (104°)	-128.64	-144.18	-110.60	-98.57
NDL (116°)	-119.11	-128.09	-116.30	-96.67
NDL (123°)	-120.24	-124.69	-100.90	-99.10

Table 5.23 NDL and SLL for 20-element LAA with multiple nulling

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
E-MFO	1.0000	0.9724	0.8467	0.8524	0.6514	0.5746	0.4994	0.2994	0.1822	0.1211
HSA	1.0000	0.9729	0.9702	0.9106	0.7288	0.5826	0.5304	0.3921	0.2140	0.2265
BSA	0.9875	1.0000	0.9900	0.8505	0.7458	0.6033	0.3613	0.4650	0.2064	0.1512

Table 5.24 Optimized amplitude coffieicent for 20 element LAA with multiple nulls

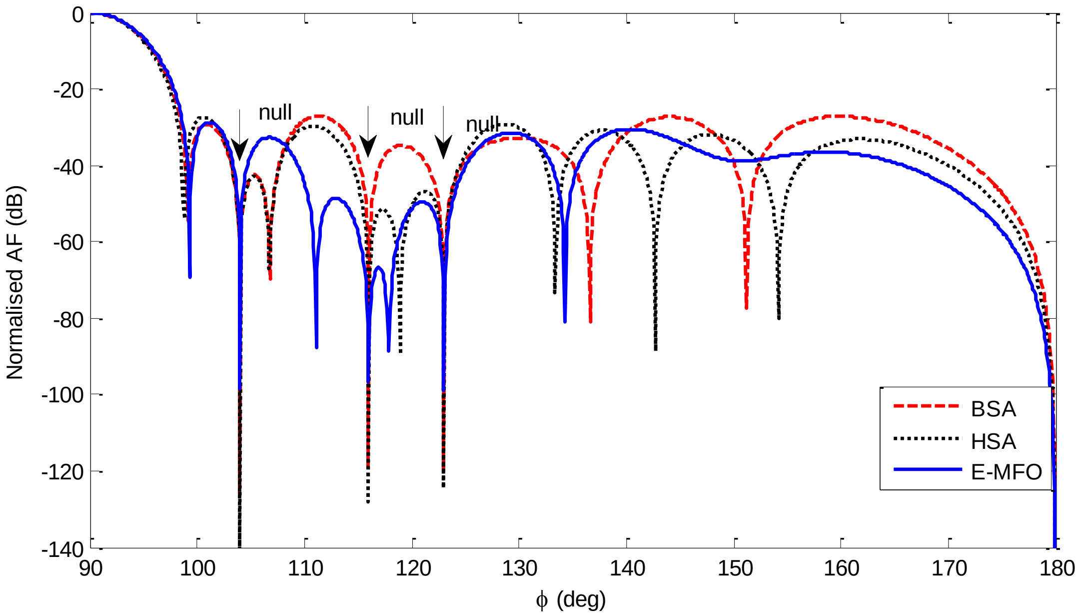


Figure 5.10 Radiation pattern of LAA with nulls at 104°, 116° and 123°

Case 4: Broader nulling

The synthesis of LAA with broad null is done in the last example. The broad nulls are used where the direction of arrival of interference is not known prior or it may change with time. Sometimes, continuous steering is required to get sharp null and desired signal to noise ratio. In this example, broad null is imposed at 120° with $\Delta\emptyset = 5°$, So, the null will be imposed in range of 117.5° and 122.5°. The SLL and NDL parameters of E-MFO are compared to HSA and BSA in Table 5.25. Table 5.26 shows the amplitude excitations for 20-element antenna array with broad nulling. The Figure 5.8 shows that in the range of interest, nulls are deeper than -60 dB. This result shows the effectiveness of proposed approach in imposing multiple nulls.

Parameters	**HSA [135]**	**BSA [134]**	**E-MFO**
SLL (in dB)	-29.28	-28.92	-31.00
NDL (dB) (minimum)	-86.54	-113.2	-93.45
NDL (dB) (maximum)	-52.18	-59.57	-63.79

Table 5.25 NDL and SLL for 20-element with broad null steering

Element	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
E-MFO	0.9859	1.0000	0.9977	0.7961	0.7444	0.5510	0.4947	0.3807	0.2413	0.1055
HSA	1.0000	0.9684	0.9304	0.8438	0.6963	0.5546	0.5293	0.4359	0.2296	0.2162
BSA	1.0000	0.9479	0.9315	0.8852	0.6322	0.6086	0.4987	0.4163	0.2450	0.1921

Table 5.26 Amplitude excitations for 20 element LAA with broad nulling

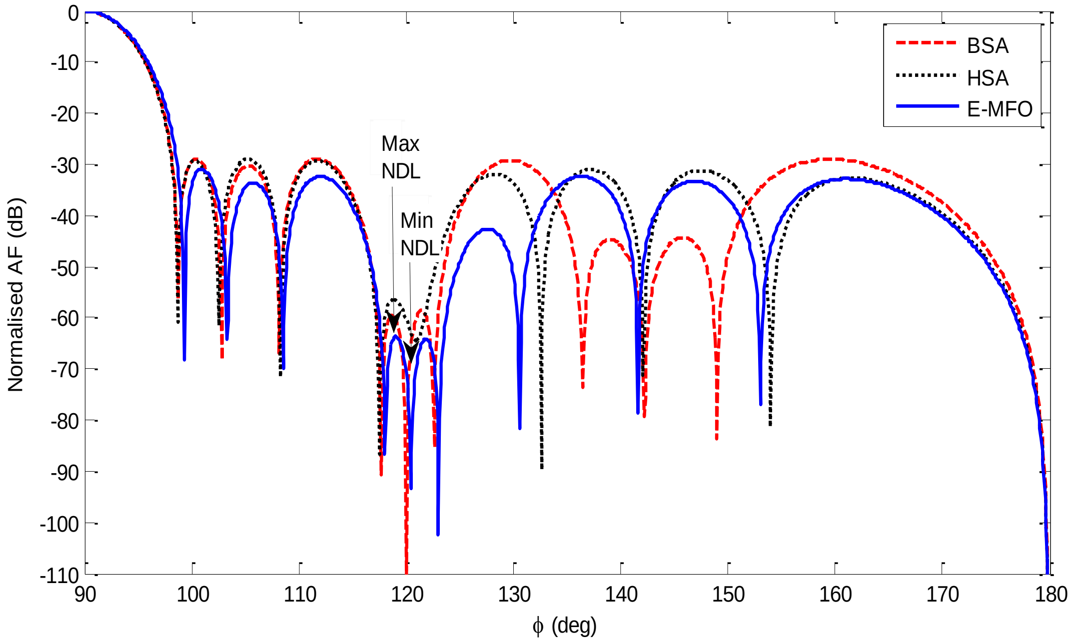


Figure 5.8 Radiation pattern of LAA with broad nulls at 120° with $\Delta\emptyset = 5°$

CHAPTER 6

CONCLUSIONS AND FUTURE SCOPE

6.1 CONCLUSION

Nature inspired algorithms are popular in real world for solving complex, non-linear and non-differentiable problems. Up till now, number of algorithms have been proposed which mimics the natural phenomenon. Every new algorithm tries to overcome the drawbacks of previous algorithm. In this work, MFO is employed for solving real world problem. After, extensive study of MFO and its applications, it is found that MFO lacks in exploration capabilities. Hence, to improve the exploration, an enhanced version of E-MFO is proposed. These modifications ensure the balance of exploration and exploitation, increase in diversity and improvement in convergence rate of E-MFO. Division of iterations, exponential step size, random attraction model and influence of best flame has been added to improve the performance of MFO. Basically, in E-MFO, iterations are divided into two groups, first group emphasizes on exploration and second group on exploitation. Further, instead of using linear decreasing step size, E-MFO used exponential step size to make proper balance between exploration and exploitation. A random attraction model is employed which increase the diversity of algorithm and helps the solutions to jump out of local minima effectively. This property makes E-MFO faster and robust in comparison to MFO. E-MFO includes one controlling parameter which controls the effect of best flame over corresponding flame. This modification leads to more exploitation in algorithm. The effectiveness of E-MFO is checked on twenty benchmark functions and it is clear from the results that it provides better results in comparison to other state-of-the-art algorithms like BA, BFP, DE, FA and FPA. The convergence curves are also drawn which shows that E-MFO has capability to reach the global minima in less number of iterations. To analyze the effect of population on MFO and E-MFO, three different population sizes i.e. 30, 60 and100 are considered and results shows that there is no significant change in performance of E-MFO with increase in population size but MFO performs better as population size increases. The E-MFO is also tested for different dimension sizes of benchmark functions which are 30, 50, 100, 200 and 500 and it is clear from the results that E-MFO performed better at all dimension sizes. The statistical testing has also been performed to show the worth of proposed algorithm.

Due to its faster convergence and better response E-MFO has been applied for synthesis of linear antenna array. The aim of synthesis of linear antenna array is to reduce the side lobe levels and null placement in desired direction. In the first instance, three different antenna arrays are optimized for achieving minimum SLL using E-MFO. The results show that the E-MFO achieves minimum SLL than MFO CS, TS, PSO, SADE, TM, and BBO. In the second instance, 20-element antenna is considered and the aim is to reduce SLL and place single, double, triple and broad null in different directions. Again, E-MFO outperforms other methods such as BSA, HSA and BBO.

Hence, from the results of benchmark functions and antenna design problem, it is concluded that proposed approach is better than original MFO algorithm and also can be generalized for other real-world problems. This works helped to understand the other algorithms like FA, BBO, DE and others as well.

6.2 FUTURE WORK

For future work, there are so many directions in which E-MFO can be efficiently recommended. This algorithm can be applied in various engineering problems like digital filter design, wireless sensor network, knapsack problems and others. The work can be extended into the field of medical science for wrapper based medical segmentation and RNA structure prediction. In the field of neural networks, proposed approach can be used to optimal design of feed forward and back propagation networks. The work can be extended in field of computer science by solving travelling salesman problem, vehicle routing problem etc. Digital image processing operations can also be implemented with proposed algorithm. Also, binary and multi-objective versions can be developed to solve practical optimization problems. Further, to enhance its performance it can be hybridized with other meta-heuristic algorithms like MFO/DE, MFO/BBO, MFO/PSO and others. More research can be done by exploiting the parameters of MFO like number of flames and step size. New modifications can be proposed to improve the performance of MFO.

REFERENCES

[1] L. Coe, *Wireless Radio: A History*. New York: McFarland & Company, 2006.

[2] L. W. Alvarez, *Alvarez: Adventures of a Physicist*. New York: Basic Books, 1987.

[3] Fallahi R and Roshandel M (2007). Effect of mutual coupling and configuration of concentric circular array antenna on the signal to interference performance in CDMA systems, *Progress in Electromagnetics Research*, 76, 427–447.

[4] Chan SC and Pun CKS (2002). On the design of digital broadband beamformer for uniform circular array with frequency invariant characteristics, *International Symposium on Circuits and systems* [2th: Arizona, USA:2002], pp. 693-696.

[5] Chrisman BP (1990). Planar array antenna design analysis. *Proceedings of the Tactical Communications Conference (Challenges of the 1990's)* [USA: 1990], pp. 705-731.

[6] Phased array antenna basics and types, Available at http://www.rfwireless-world.com/Terminology/phased-array-antenna-basics-and-types.html (Accessed on 5th July 2017)

[7] Liao WP (1999). Array pattern synthesis with null steering using genetic algorithms by controlling only the current amplitudes, *International journal of electronics*, 86(4), 445-457.

[8] Chaipanya P (2015). Null steering using low profile weights for base station antenna, *International Symposium* on *Microwave, Antenna, Propagation, and EMC Technologies (MAPE)* [6th: Shanghai, China: 2015], pp. 179-183.

[9] Ibrahim HM (1991). Null steering by real-weight control-a method of decoupling the weights, *IEEE transactions on antennas and propagation*, 39(11), 1648-1650.

[10] Cardil R (2010). Kepler: The Volume of a Wine Barrel, *Matematics Visuales*, 8.

[11] Andersen K (1983). The mathematical technique in Fermat's deduction of the law of refraction, *Historia mathematica*, 10(1), 48-62.

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

[13] Fraser C (1984). The origin of Euler's Variational Calculus, *Archive for History of Exact Science*, 47, 103-140.

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

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

[16] Nelder JA and Mead R (1965). A simplex method for function minimization, *The computer journal*, *7*(4), 308-313.

[17] Rao SS and Rao SS. *Engineering optimization: theory and practice*. New York: Wiley and Sons, 2009.

[18] Yang XS, Deb S and Fong S (2014). Metaheuristic algorithms: optimal balance of intensification and diversification, *Applied Mathematics*, 8(3), 1-7.

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

[20] Price K and Storn RM and Lampinen JA. *Differential evolution: a practical approach to global optimization*, Springer Science & Business Media, 2006.

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

[22] Eberhart R and Kennedy J (1995). A new optimizer using particle swarm theory, *International Symposium on Micro Machine and Human Science* [6th : Nagoya, Japan:1995], pp. 39-43.

[23] Karaboga D and Basturk B (2007). Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems, *International Fuzzy Systems Association World Congress*, 789-798

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

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

[26] Beni G and Wang J (1993). Swarm intelligence in cellular robotic systems. In *Robots and Biological Systems: Towards a New Bionics?* 703-712.

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

[28] Yang XS (2010). Firefly algorithm, stochastic test functions and design optimization. *International Journal of Bio-Inspired Computation*, 2(2), 78-84.

[29] Firefly (Enigmatic, Enchanting, Endangered). Available at http://www.firefly.org/firefly-pictures.html (Accessed on 29th June 2017).

[30] Sinha, N., Chakrabarti, R., & Chattopadhyay, P. K. (2003). Evolutionary programming techniques for economic load dispatch, *IEEE Transactions on evolutionary computation*, *7*(1), 83-94.

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

[32] Mirjalili S (2015). The ant lion optimizer, *Advances in Engineering Software*, 83, 80-98.

[33] Alavi M and Henderson JC (1981). An evolutionary strategy for implementing a decision support system, *Management Science*, 27(11), 1309-1323.

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

[35] Murthy P and Kumar A (1976). Synthesis of linear antenna arrays, *IEEE Transactions on Antennas and Propagation*, 24(6), 865-870.

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

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

[38] 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.

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

[40] Sutton AM, Lunacek M and Whitley LD (2007). Differential evolution and non-separability: using selective pressure to focus search. In *Proceedings of the annual conference on Genetic and evolutionary computation*, [9th: London, England: 2007], pp. 1428-1435.

[41] Liang W, Zhang L and Wang M (2011). The chaos differential evolution optimization algorithm and its application to support vector regression machine, *Journal of Software*, 6(7), 1297-1304.

[42] Wang H, Wu Z and Rahnamayan S. Differential evolution enhanced by neighborhood search, *IEEE Congress on Evolutionary Computation (CEC),* 2010, pp. 1-8.

[43] Wang L *et al.* (2015). A hybrid backtracking search optimization algorithm with differential evolution, *Mathematical Problems in Engineering, 2015.*

[44] Xue, J., Wang, K. J., Nguyen, P. H., & Wu, S. Y. A new algorithm hybridizing differential evolution with particle swarm optimization.

[45] El Ela AA, Abido MA and Spea SR (2010). Optimal power flow using differential evolution algorithm, *Electric Power Systems Research,* 80(7), 878-885.

[46] Noman N and Iba H (2008). Differential evolution for economic load dispatch problems, *Electric Power Systems Research*, 78(8), 1322-1331.

[47] Chiou JP, Chang CF and Su CT (2004). Ant direction hybrid differential evolution for solving large capacitor placement problems, *IEEE Transactions on power systems*, 19(4), 1794-1800.

[48] Shi Y (2001). Particle swarm optimization: developments, applications and resources, *Proceedings of the 2001 Congress* in *Evolutionary Computation,* 1, 81-86.

[49] Del Valle Y *et al.* (2008). Particle swarm optimization: basic concepts, variants and applications in power systems, *IEEE Transactions on evolutionary computation*, 12(2), 171-195.

[50] Pandey S *et al.* (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments, *IEEE international conference on Advanced information networking and applications* [24th : Parth, Austraila: 2010], pp. 400-407.

[51] 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.

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

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

[54] 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.

[55] Zhu G and Kwong S (2010). Gbest-guided artificial bee colony algorithm for numerical function optimization, *Applied mathematics and computation*, 217(7), 3166-3173.

[56] Gao WF and Liu SY (2012). A modified artificial bee colony algorithm, *Computational and Operations Research*, 39(3), 687–697.

[57] Gao WF, Liu SY and Huang LL (2013). A novel artificial bee colony algorithm with Powell's method, *Applied Soft Computing*, 13(9), 3763-3775.

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

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

[60] Mirjalili S and Lewis A (2016). The whale optimization algorithm, *Advances in Engineering Software*, 95, 51-67.

[61] Bhattacharya A and Chattopadhyay PK (2010). Biogeography-based optimization for different economic load dispatch problems, *IEEE transactions on power systems*, 25(2), 1064-1077.

[62] Rarick R *et al.* (2009, October). Biogeography-based optimization and the solution of the power flow problem. *IEEE International Conference on Systems, Man and Cybernetics* [USA: 2009], pp. 1003-1008.

[63] Walton S *et al.* (2011). Modified cuckoo search: a new gradient free optimization algorithm, *Chaos, Solitons & Fractals*, 44(9), 710-718.

[64] Zhang Y, Wang L and Wu Q (2012). Modified Adaptive Cuckoo Search (MACS) algorithm and formal description for global optimization, *International Journal of Computer Applications in Technology*, 44(2), 73-79.

[65] Zhao P and Li H (2012, October). Opposition-based Cuckoo search algorithm for optimization problems, *International Symposium on Computational Intelligence and Design* [5th: China: 2012], pp. 344-347.

[66] Yang XS and Deb S (2013). Multiobjective cuckoo search for design optimization, *Computers & Operations Research*, 40(6), 1616-1624.

[67] Khodier M (2013). Optimisation of antenna arrays using the cuckoo search algorithm, *IET Microwaves, Antennas & Propagation*, 7(6), 458-464.

[68] Gangopadhyaya M *et al.* (2016, October). Resonant frequency optimization of coaxially fed rectangular Microstrip Antenna using Cuckoo Search algorithm, *Electronics and Mobile Communication Conference in Information Technology* [7th: Canada: 2016], pp. 1-4.

[69] Bindu AH and Reddy MD (2013). Economic load dispatch using cuckoo search algorithm, *International Journal of Engineering Research and Applications*, 3(4), 498-502

[70] Dey N *et al.* (2013). Optimization of scaling factors in electrocardiogram signal watermarking using cuckoo search, *International Journal of Bio-Inspired Computation*, 5(5), 315-326.

[71] Yu S *et al.* (2015). A variable step size firefly algorithm for numerical optimization, *Applied Mathematics and Computation*, *263*, 214-220.

[72] Shafaati M and Mojallali H (2012). Modified firefly optimization for IIR system identification, *Journal of Control Engineering and Applied Informatics*,14(4), 59–69.

[73] Olamaei J *et al*. (2013). A new adaptive modified firefly algorithm to solve optimal capacitor placement problem. *Conference on Electrical Power Distribution Network*s [18th: Kermanshah, Iran: 2013]

[74] Yang XS (2010). Firefly algorithm, Levy flights and global optimization, *In Research and Development in Intelligent Systems.* Verlag berlin Heidelberg: Springer London,2010, 209–218.

[75] Yu S, Yang S and Su S (2013). Self-adaptive step firefly algorithm, *Journal of Applied Mathematics*, 8.

[76] Verma OP, Aggarwal D and Patodi T (2016). Opposition and dimensional based modified firefly algorithm, *Expert Systems with Applications*, 44, 168-176.

[77] Gandomi A *et al.* (2013) Firefly algorithm with chaos, *Communications in Nonlinear Science and Numerical Simulation*, 18(1), 89–98.

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

[79] Sharaqa A and Dib N (2014). Circular antenna array synthesis using firefly algorithm, *International Journal of RF and Microwave Computer‐Aided Engineering*, 24(2), 139-146.

[80] Zhou C *et al.* (2015). A method of Two-Dimensional Otsu image threshold segmentation based on improved Firefly Algorithm, *IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems* [China, 2015], pp. 1420-1424.

[81] Rajan A and Malakar T (2015). Optimal reactive power dispatch using hybrid Nelder–Mead simplex based firefly algorithm, *International Journal of Electrical Power & Energy Systems*, 66, 9-24.

[82] Khadwilard A *et al*. (2012). Application of firefly algorithm and its parameter setting for job shop scheduling, *Journal of Industrial Technology*, 8(1).

[83] Yılmaz S *et al.* (2014). Modified bat algorithm,.*Elektronika ir Elektrotechnika*, *20*(2), 71-78.

[84] Alihodzic A. and Tuba M (2014). Improved bat algorithm applied to multilevel image thresholding, *The Scientific World Journal*, *2014*.

[85] Gandomi, AH and Yang XS (2014). Chaotic bat algorithm. *Journal of Computational Science*, 5(2), 224-232.

[86] Kora P and Kalva SR (2015). Improved Bat algorithm for the detection of myocardial infarction, *Springer Plus*, *4*(1), 666.

[87] Komarasamy G and Wahi A (2012). An optimized K-means clustering technique using bat algorithm, *European Journal of Scientific Research*, 84(2), 263-273.

[88] Lemma TA and Bin Mohd HF (2011). Use of fuzzy systems and bat algorithm for exergy modelling in a gas turbine generator, *IEEE Colloquium on Humanities, Science and Engineering* [Malaysia: 2011], pp. 305–310.

[89] Lin JH *et al.* (2012). A chaotic Levy flight bat algorithm for parameter estimation in nonlinear dynamic biological systems, *Journal of Computer and Information Technology,* 2(2), 56–63.

[90] 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* [Canada: 2016], pp. 3116-3123.

[91] 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.

[92] 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.

[93] Emary E, Zawbaa HM and Hassanien AE (2016). Binary grey wolf optimization approaches for feature selection, *Neurocomputing,* 172, 371-381.

[94] 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.

[95] Lal DK, Barisal AK 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.

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

[97] Mostafa A *et al*. (2017). Liver segmentation in MRI images based on whale optimization algorithm, *Multimedia Tools and Applications*, 1−24.

[98] Zhou Y, Ling Y and Luo, Q. (2017). Lévy Flight Trajectory−Based Whale Optimization Algorithm for Global Optimization, *IEEE Access*, 5, 6168-6186.

[99] Trivedi IN *et al.* (2016, December). An emission constraint environment dispatch problem solution with microgrid using Whale Optimization Algorithm, *IEEE National Power Systems Conference,* [India: 2016], pp. 1−6.

[100] Hassanien AE *et al.* (2016, October). Historic handwritten manuscript binarization using whale optimization, *IEEE International Conference on Systems, Man, and Cybernetics* [Hungary: 2016], pp. 003842−003846.

[101] Li Z *et al.* (2016). Lévy-flight moth-flame algorithm for function optimization and engineering design problems, *Mathematical Problems in Engineering, 2016.*

[102] Soliman GM, Khorshid MM and Abou-El-Enien TH (2016). Modified moth-flame optimization algorithms for terrorism prediction, *International Journal of Application or Innovation in Engineering and Management*, 5(7), 47-58.

[103] Muangkote N, Sunat K and Chiewchanwattana S (2016). Multilevel thresholding for satellite image segmentation with moth-flame based optimization, *International Joint Conference on Computer Science and Software Engineering* [13th: Khon Kaen, Thailand: 2016], pp. 1-6.

[104] Yamany W *et al.* (2015). Moth-flame optimization for training multi-layer preceptrons, *IEEE International Computer Engineering Conference* [11th: Cairo, Egypt:2015] , pp. 267-272.

[105] Allam D, Yousri DA and Eteiba MB (2016). Parameters extraction of the three-diode model for the multi-crystalline solar cell/module using Moth-Flame Optimization Algorithm, *Energy Conversion and Management*, 123, 535-548.

[106] Bentouati B, Chaib L and Chettih S (2016). Optimal power flow using the moth flam optimizer: A case study of the Algerian power system, *Indonesian Journal of Electrical Engineering and Computer Science*, 1(3), 431-445.

[107] Hassanien AE *et al.* (2017). An improved moth flame optimization algorithm based on rough sets for tomato diseases detection, *Computers and Electronics in Agriculture*, 136, 86-96.

[108] Ceylan O (2016). Harmonic elimination of multilevel inverters by moth-flame optimization algorithm, *International Symposium on Industrial Electronics* [Banja Luka, Bosnia: 2016], pp. 1-5.

[109] Parmar SA *et al.* (2016). Optimal active and reactive power dispatch problem solution using moth-flame optimizer algorithm, *International Conference on Energy Efficient Technologies for Sustainability,* 2016, pp. 491-496.

[110] Garg P and Gupta A (2017). Optimized Open Shortest Path First Algorithm Based on Moth Flame Optimization, *Indian Journal of Science and Technology*, 9(48).

[111] Fleming, J. A. (1916). The Principles of Electronic Wave Telegraphy and Telephony.

[112] S. G. Brown, Brit. Patent No. 14,449, 1899.

[113] L. Forest, U.S. Patent No. 749,131.

[114] Marconi G (1906). On methods, whereby the radiation of electric waves may be mainly confined to certain directions, and whereby the receptivity of a receiver may be restricted to electric waves emanating from certain directions, *Proceedings of the Royal Society of London, Series A, Containing Papers of a Mathematical and Physical Character*, 77(518), 413-421.

[115] Haupt RL. *Antenna arrays: a computational approach*, John Wiley & Sons, 2007.

[116] Spradley J (1966). A volumetric electrically scanned two-dimensional microwave antenna array, IEEE *International Convention Record*, pp. 204-212.

[117] 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.

[118] Yan KK and Lu Y (1997). Sidelobe reduction in array-pattern synthesis using genetic algorithm, *IEEE Transactions on Antennas and Propagation*, 45(7), 1117-1122.

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

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

[121] 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.

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

[123] Dib NI, Goudos SK and Muhsen H (2010). Application of Taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays, *Progress in Electromagnetics Research*, 102, 159-180.

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

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

[126] 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.

[127] Babayigit B, Akdagli A and Guney K (2006). A clonal selection algorithm for null synthesizing of linear antenna arrays by amplitude control, *Journal of electromagnetic Waves and Applications*, 20(8), 1007-1020.

[128] Guney K and Babayigit B (2008). Amplitude-only pattern nulling of linear antenna arrays with the use of an immune algorithm, *International Journal of RF and Microwave Computer-Aided Engineering*, 18(5), 397-409.

[129] Guney K and Onay M (2007). Amplitude-only pattern nulling of linear antenna arrays with the use of bees algorithm, *Progress in Electromagnetics Research*, 70, 21-36.

[130] Guney K and Basbug S (2008). Interference suppression of linear antenna arrays by amplitude-only control using a bacterial foraging algorithm, *Progress in Electromagnetics Research*, 79, 475-497.

[131] Guney K, Durmus A and Basbug S (2009). A plant growth simulation algorithm for pattern nulling of linear antenna arrays by amplitude control, *Progress in Electromagnetics Research B*, 17, 69-84.

[132] Karaboga N, Güney K and Akdagli A (2002). Null steering of linear antenna arrays with use of modified touring ant colony optimization algorithm, *International Journal of RF and Microwave Computer‐Aided Engineering*, 12(4), 375-383.

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

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

[135] 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.

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

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

[138] U. Singh and R. Salgotra (2016). Optimal synthesis of linear antenna arrays using modified spider monkey optimization, *Arabian Journal for Science and Engineering*, 1-17.

[139] Guney K and Basbug S (2011). Seeker optimization algorithm for interference suppression of linear antenna arrays by controlling position-only, phase-only, and amplitude-only, *International Journal of RF and Microwave Computer‐Aided Engineering*, 21(5), 505-518.

[140] Jamil M and Yang XS (2013). A literature survey of benchmark functions for global optimization problems, *International Journal of Mathematical Modelling and Numerical Optimization*, 4(2), 150-194.

[141] Derrac J (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.

[142] Khodier MM and Al-Aqeel M (2009). Linear and circular array optimization: A study using particle swarm intelligence, *Progress in Electromagnetics Research B*, 15, 347-373.

[143] Khodier M (2013). Optimisation of antenna arrays using the cuckoo search algorithm, *IET Microwaves, Antennas & Propagation*, 7(6), 458-464.

LIST OF PUBLICATIONS

[1] Kaur KP, Salgotra R and Singh U (2017). An Improved Firefly algorithm for Numerical Optimization, *International Conference on Innovations in Information, Embedded and Communication Systems* [4^{th}: Coimbatore, India: 2017]. (Status: Accepted)

[2] Kaur KP, Singh U and Salgotra R (2017). An Enhanced Moth flame optimization. *In Neural Computing and Application*. (Status: Communicating)

Komal_14.7.17

ORIGINALITY REPORT

%8	%4	%5	%2
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	mcf.gsfc.nasa.gov Internet Source	%1
2	www.slideshare.net Internet Source	%1
3	Singh, Urvinder, and Rohit Salgotra. "Synthesis of linear antenna array using flower pollination algorithm", Neural Computing and Applications, 2016. Publication	%1
4	uvb.nrel.colostate.edu Internet Source	<%1
5	Submitted to Arab Open University Student Paper	<%1
6	www.alimirjalili.com Internet Source	<%1
7	Submitted to CONACYT Student Paper	<%1
8	Submitted to Heriot-Watt University Student Paper	<%1

9	Mirjalili, Seyedali. "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm", Knowledge-Based Systems, 2015. Publication	<%1
10	www.scribd.com Internet Source	<%1
11	jnusrv01.kek.jp Internet Source	<%1
12	goes.ngdc.noaa.gov Internet Source	<%1
13	"Bio-Inspired Computing -- Theories and Applications", Springer Nature, 2015 Publication	<%1
14	Ouyang, Hai-bin, Li-qun Gao, Xiang-yong Kong, De-xuan Zou, and Steven Li. "Teaching-learning based optimization with global crossover for global optimization problems", Applied Mathematics and Computation, 2015. Publication	<%1
15	sd-www.jhuapl.edu Internet Source	<%1
16	Lecture Notes in Computer Science, 2015. Publication	<%1
17	Studies in Computational Intelligence, 2016. Publication	<%1

18	Nipotepat Muangkote, Khamron Sunat, Sirapat Chiewchanwattana. "Multilevel thresholding for satellite image segmentation with moth-flame based optimization", 2016 13th International Joint Conference on Computer Science and Software Engineering (JCSSE), 2016 Publication	<%1
19	Caughlan, G.R.. "Tables of thermonuclear reaction rates for low-mass nuclei (1 =< Z =< 14)", Atomic Data and Nuclear Data Tables, 198503 Publication	<%1
20	Submitted to Stefan cel Mare University of Suceava Student Paper	<%1
21	Li, Cunbin, Shuke Li, and Yunqi Liu. "A least squares support vector machine model optimized by moth-flame optimization algorithm for annual power load forecasting", Applied Intelligence, 2016. Publication	<%1
22	www.ijcaonline.org Int ernet Source	<%1
23	www.coursehero.com Int ernet Source	<%1
24	Submitted to Indian School of Mines Student Paper	<%1

25	Singh, Shishir Kumar, Kuldeep Yadav, and G.K. Mahanti. "Generation of Staircase Type Side Lobe Level from Linear Array Antenna Using Fast Fourier Transform", 2014 Fourth International Conference on Advances in Computing and Communications, 2014. Publication	<%1
26	Submitted to Dublin City University Student Paper	<%1
27	Cholmin Rim, Songhao Piao, Guo Li, Unsun Pak. "A niching chaos optimization algorithm for multimodal optimization", Soft Computing, 2016 Publication	<%1
28	Smida, A, L Gargouri, R Ghayoula, H Trabelsi, and A Gharsallah. "Amplitude-only adaptive nulling based on Taguchi's method", Eighth International Multi-Conference on Systems Signals & Devices, 2011. Publication	<%1
29	Zhao, Huiru, Haoran Zhao, and Sen Guo. "Using GM (1,1) Optimized by MFO with Rolling Mechanism to Forecast the Electricity Consumption of Inner Mongolia", Applied Sciences, 2016. Publication	<%1

30	Khalilpourazari, Soheyl, and Seyed Hamid Reza Pasandideh. "Multi-item EOQ model with nonlinear unit holding cost and partial backordering: moth-flame optimization algorithm", Journal of Industrial and Production Engineering, 2016. Publication	<%1
31	semspub.epa.gov Internet Source	<%1
32	tes.la.asu.edu Internet Source	<%1
33	emergency.cdc.gov Internet Source	<%1
34	Indrajit N. Trivedi, Pradeep Jangir, Siddharth A. Parmar, Narottam Jangir. "Optimal power flow with voltage stability improvement and loss reduction in power system using Moth-Flame Optimizer", Neural Computing and Applications, 2016 Publication	<%1
35	code.google.com Internet Source	<%1
36	Chakraborty, S., N. Dey, S. Samanta, A.S. Ashour, and V.E. Balas. "Firefly algorithm for optimized nonrigid demons registration", Bio-Inspired Computation and Applications in	<%1

Image Processing, 2016.
Publication

37	Submitted to VIT University Student Paper	<%1
38	Submitted to Birla Institute of Technology and Science Pilani Student Paper	<%1
39	Sharaqa, Ashraf, and Nihad Dib. "Design of Linear and Elliptical Antenna Arrays Using Biogeography Based Optimization", Arabian Journal for Science and Engineering, 2014. Publication	<%1
40	Lecture Notes in Electrical Engineering, 2014. Publication	<%1
41	Advances in Intelligent Systems and Computing, 2015. Publication	<%1
42	dl1.ponato.com Int ernet Source	<%1
43	archive.epa.gov Int ernet Source	<%1
44	Submitted to Columbia University Student Paper	<%1
45	Singh, Urvinder, and Rohit Salgotra. "Optimal Synthesis of Linear Antenna Arrays Using	<%1

Modified Spider Monkey Optimization", Arabian Journal for Science and Engineering, 2016.
Publication

---

46 Yu, Kunjie, Xin Wang, and Zhenlei Wang. "Multiple learning particle swarm optimization with space transformation perturbation and its application in ethylene cracking furnace optimization", Knowledge-Based Systems, 2016.
Publication

<%1

---

47 El-Sayed, M.E.M.. "Nonlinear structural optimization using goal programming", Computers and Structures, 1989
Publication

<%1

---

48 www.jpier.org
Internet Source

<%1

---

49 Kitayama, S.. "Differential evolution as the global optimization technique and its application to structural optimization", Applied Soft Computing Journal, 201106
Publication

<%1

---

50 Li, Wei Wang, Lei Yao, Quanzhu Jiang, Qi. "Cloud particles differential evolution algorithm: a novel optimization method for global numerical o", Mathematical Problems in Engineering, Annual 2015 Issue
Publication

<%1

51	www.softcomputing.net Internet Source	<1%
52	Submitted to National Institute of Technology, Kurukshetra Student Paper	<1%
53	Submitted to Indian Institute of Technology, Madras Student Paper	<1%
54	Submitted to University of Stellenbosch, South Africa Student Paper	<1%
55	M. Vasudevan. "Genetic-Algorithm-Based Computational Models for Optimizing the Process Parameters of A-TIG Welding to Achieve Target Bead Geometry in Type 304 L(N) and 316 L(N) Stainless Steels", Materials and Manufacturing Processes, 6/2007 Publication	<1%
56	cmbdata.gsfc.nasa.gov Internet Source	<1%
57	Submitted to Higher Education Commission Pakistan Student Paper	<1%
58	Studies in Computational Intelligence, 2008. Publication	<1%

59	Jangir, Narottam, Mahesh H. Pandya, Indrajit N. Trivedi, R.H. Bhesdadiya, Pradeep Jangir, and Arvind Kumar. "Moth-Flame optimization Algorithm for solving real challenging constrained engineering optimization problems", 2016 IEEE Students Conference on Electrical Electronics and Computer Science (SCEECS), 2016. Publication	<1%
60	Luo, Jianping, Qiqi Liu, Yun Yang, Xia Li, Min-rong Chen, and Wenming Cao. "An artificial bee colony algorithm for multi-objective optimisation", Applied Soft Computing, 2017. Publication	<1%
61	Poikolainen, Ilpo, Giovanni Iacca, Fabio Caraffini, and Ferrante Neri. "Focusing the search: a progressively shrinking memetic computing framework", International Journal of Innovative Computing and Applications, 2013. Publication	<1%

EXCLUDE QUOTES	ON	EXCLUDE MATCHES	< 10 WORDS
EXCLUDE BIBLIOGRAPHY	ON		