# DESIGN OF CIRCULAR AND CONCENTRIC CIRCULAR ANTENNA ARRAY USING AN IMPROVED FLOWER POLLINATION ALGORITHM

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

MASTER OF ENGINEERING

in

Electronics and Communication Engineering

Submitted By


**DEEPIKA SINGH**

Roll No. 801561007


Under Supervision of


**Dr. URVINDER SINGH**
Assistant Professor
Thapar University, Patiala



ELECTRONICS AND COMMUNICATION ENGINEERING DEPARTMENT

THAPAR UNIVERSITY, PATIALA, PUNJAB


JULY, 2017

## DECLARATION

I, Deepika Singh hereby declare that the work presented in this thesis entitled "DESIGN OF CIRCULAR AND CONCENTRIC CIRCULAR ANTENNA ARRAY USING AN IMPROVED FLOWER POLLINATION ALGORITHM" in partial fulfillment of the requirement for the award of degree of Master of Engineering submitted at Electronics & Communication department, Thapar University, Patiala is an authentic record of work carried out under supervision of Dr. Urvinder Singh (Assistant Professor, ECED, Thapar University, Patiala) 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.

Deepika Singh

Date:......17.07.17......

Deepika Singh
801561007

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

Date:......17/7/17......

Dr. Urvinder Singh
Assistant Professor
ECED

## ACKNOWLEDGEMENT

It is my proud privilege to acknowledge and extend my gratitude to several persons who helped me directly or indirectly in completion of this report. I express my heart full indebtedness and owe a deep sense of gratitude to my teacher and my faculty guide **Dr. Urvinder Singh** for his sincere guidance and support with encouragement to go ahead.

I would like to give my sincere gratitude to **Dr. Alpana Agarwal**, Head of ECED, for providing me with the adequate infrastructure for carrying out the work. I am also thankful to **Dr. HemDutt Joshi**, P.G. Coordinator and **Dr. Amit Mishra,** Program Coordinator 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. Komalpreet Kaur** who have more or less contributed to the preparation of this thesis. 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.

Deepika Singh

## ABSTRACT

Flower pollination algorithm (FPA) is a novel meta-heuristic algorithm inspired from pollination of flowers. It has been applied to various fields of research and proven its worth. But the algorithm suffers from certain limitations which confines its scope to different field of applications. In present work, a modified version of FPA namely an improved flower pollination algorithm (IFPA) has been proposed to improve exploration, enhance local search and to maintain good balance between intensification and diversification. Cauchy operator has been added instead of Levy flight for improving the explorative capability. Secondly, a modified probability switching has been used to balance exploration and exploitation. The last change considers the effect of current best for enhancing the local search phase. The results of IFPA are tested on twenty benchmark functions for different population sizes. The proposed approach is also tested on different dimension sizes of benchmark problems. The results of IFPA are then compared with differential evolution (DE), bat algorithm (BA), bat flower pollination algorithm (BFP) and flower pollination algorithm (FPA). Experimental results show that IFPA algorithm is better when compared to other algorithms. Further, statistical testing has been done to prove the significance of IFPA.

IFPA is then tested on an important real-world application in the field of antenna design. Two different antenna synthesis design problem of circular antenna array (CAA) and concentric circular antenna array (CCAA) has been done using IFPA. Three cases of both arrays are analysed and then compared with different standard algorithms which shows the superiority of proposed approach in reducing the side lobe levels and first null beam width.

## TABLE OF CONTENTS


| Sr. No | Name of the Chapters | Page No |
| --- | --- | --- |
| | *Declaration* | *ii* |
| | *Acknowledgement* | *iii* |
| | *Abstract* | *iv* |
| | *Table of Contents* | *v-vi* |
| | *List of Tables* | *vii* |
| | *List of Figures* | *viii* |
| | *List of Abbreviations* | *ix* |
| *Chapter 1* | Introduction | 1-13 |
| *1.1* | History of Optimization | 1 |
| *1.2* | Optimization Design | 1 |
| *1.3* | Classification based upon optimization problems | 2 |
| *1.4* | Basics of Optimization | 3 |
| *1.5* | Nature Inspired Algorithms | 5 |
| | *1.5.1* Need of Nature Inspired Algorithms | 5 |
| | *1.5.2* Components of Nature Inspired Algorithms | 5 |
| | *1.5.3* Types of Algorithms | 6 |
| *1.6* | Application of Optimization Algorithms | 10 |
| *1.7* | Antenna Array | 10 |
| *1.8* | Objectives | 12 |
| *1.9* | Thesis Outline | 12 |
| *Chapter 2* | Literature Survey | 14-22 |
| *2.1* | Literature Survey of Nature Inspired Algorithms | 14 |
| *2.2* | Literature Survey of Antenna Array | 20 |
| | *2.2.1* Circular Antenna Array. | 20 |
| | *2.2.2* Concentric Circular Antenna Array | 21 |
| *Chapter 3* | Proposed Approach | 23-28 |
| *3.1* | Flower Pollination Algorithm | 23 |
| *3.2* | Improved Flower Pollination Algorithm | 25 |
| *Chapter 4* | Problem Formulation | 29-31 |
| *4.1* | Design Equations | 29 |
| | *4.1.1* Circular Antenna Array | 29 |
| | *4.1.2* Concentric Circular Antenna Array | 30 |

| | | |
|---|---|---|
| *Chapter 5* | Results and Discussion | 32-60 |
| *5.1* | Benchmarking Results | 32 |
| *5.1.1* | Test Function | 32 |
| *5.1.2* | Parameter Settings | 33 |
| *5.1.3* | Influence of Population Size | 34 |
| *5.1.4* | Effect of Dimension | 44 |
| *5.1.5* | Statistical Testing | 47 |
| *5.1.6* | Convergence Curve | 48 |
| *5.2* | Simulation Results for the synthesis of Circular Antenna Array | 52 |
| *5.3* | Simulation Results for the synthesis of Concentric Circular Antenna Array | 56 |
| *Chapter 6* | Conclusions and Future Scope | 61-62 |
| *6.1* | Conclusion | 61 |
| *6.2* | Future Scope | 61 |
| | *References* | 63 |
| | *Publications* | 70 |

## LISTS OF TABLES


| **Sr. No** | **Table Details** | **Page No** |
|---|---|---|
| *Table 5.1* | *Test Function Table* | 32 |
| *Table 5.2* | *Parameter settings* | 34 |
| *Table 5.3* | *Results of IFPA compared with different other algorithms for population size 20* | 35 |
| *Table 5.4* | *Results of IFPA compared with different other algorithms for population size 40* | 37 |
| *Table 5.5* | *Results of IFPA compared with different other algorithms for population size 60* | 40 |
| *Table 5.6* | *Results of IFPA compared with different other algorithms for population size 80* | 42 |
| *Table 5.7* | *Results of IFPA compared with other algorithms for dimension size of 50* | 45 |
| *Table 5.8* | *Results of IFPA compared with other algorithms for dimension size of 100* | 46 |
| *Table 5.9* | *p test values of different algorithms* | 48 |
| *Table 5.10* | *Comparison of results obtained by IFPA with various algorithms for element N=8* | 53 |
| *Table 5.11* | *Comparison of results obtained by IFPA with various algorithms for element N=10* | 55 |
| *Table 5.12* | *Comparison of results obtained by IFPA with various algorithms for element N=12* | 56 |
| *Table 5.13* | *Comparison of results obtained by IFPA with various algorithms for ring (N1=4, N2=6, N3=8)* | 57 |
| *Table 5.14* | *Comparison of results obtained by IFPA with various algorithms for ring (N1=6, N2=8, N3=10)* | 58 |
| *Table 5.15* | *Comparison of results obtained by IFPA with various algorithms for ring (N1=8, N2=10, N3=12)* | 60 |

## LISTS OF FIGURES


| Sr. No | Figure Details | Page No |
|---|---|---|
| *Figure 1.1* | *Model for design of optimization* | 2 |
| *Figure 1.2* | *Cities connected for TSP problem* | 4 |
| *Figure 1.3* | *Concept of local and global optima* | 6 |
| *Figure 1.4* | *Nature inspired algorithms* | 6 |
| *Figure 1.5* | *Examples of SI* | 8 |
| *Figure 1.6* | *Outline of work* | 13 |
| *Figure 3.1* | *Pseudo code of FPA* | 24 |
| *Figure 3.2* | *Effect of p with iterations* | 25 |
| *Figure 3.3* | *Flowchart of IFPA* | 28 |
| *Figure 3.4* | *Pseudo code of IFPA* | 27 |
| *Figure 4.1* | *N-element non-uniform circular antenna array* | 30 |
| *Figure 4.2* | *Geometrical configuration of CCAA* | 31 |
| *Figure 5.1* | *Convergence graphs* | 49 |
| *Figure 5.2* | *Radiation pattern obtained by non-uniform CAA for N=8 elements* | 53 |
| *Figure 5.3* | *Radiation pattern obtained by non-uniform CAA for N=10 elements* | 54 |
| *Figure 5.4* | *Radiation pattern obtained by non-uniform CAA for N=12 elements* | 55 |
| *Figure 5.5* | *Radiation pattern obtained for comparative study of various algorithms for N1=4, N2=6, N3=8 elements* | 58 |
| *Figure 5.6* | *Radiation pattern obtained for comparative study of various algorithms for N1=6, N2=8, N3=10 elements* | 59 |
| *Figure 5.7* | *Radiation pattern obtained for comparative study of various algorithms for N1=8, N2=10, N3=12 elements* | 59 |

# LIST OF ABBREVIATIONS

| | |
|---|---|
| ABC | Artificial Bee Colony |
| ACO | Ant Colony Optimization |
| BA | Bat Algorithm |
| BBBC | Big Bang Big Crunch |
| BFP | Bat Flower Pollination Algorithm |
| BH | Black Hole |
| CAA | Circular Antenna Array |
| CCAA | Circular Concentric Antenna Array |
| CS | Cuckoo Search |
| CSO | Cat Swarm Optimization |
| CSS | Charged System Search |
| 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 |
| HEP | Hybrid Evolutionary Programming |
| IFPA | Improved Flower Pollination Algorithm |
| LAA | Linear Antenna Array |
| NIA | Nature Inspired Algorithms |
| PSO | Particle Swarm Optimization |
| SA | Simulated Annealing |
| SI | Swarm Intelligence |
| SLL | Side Lobe Level |
| WOA | Whale Optimization Algorithm |

# CHAPTER 1
# INTRODUCTION

## 1.1 HISTORY OF OPTIMIZATION

The origin of optimization can be found out from 300 BC, when Euclid found the minimum distance between two joining points as the length of straight line. Euclid also proved that square possess the greatest area amongst the rectangles by calculating the length of all edges. As the era proceeded, Heron in 100 BC proved that the light travels the shortest distance between any two points when being reflected from the mirror. The optimization problems such as calculating the optimal dimension of wine barrel by J. Kepler (1615) and minimum time taken as light travels between any two points by P. De Fermat (1675) was considered. Then two scientist namely, I. Newton in 1660s and G W von Leibniz in 1670s discovered the mathematical analysis for having the basis for calculus of variations. Afterwards, J. L. Lagrange invented the constraint optimization problem that involves addition of the unfamiliar multipliers. Cauchy in 1847 was the first scientist who made his attempt for solving gradient based method application for unconstraint optimization. Then simplex method was proposed in 1947 by G. Dantzig. In 1948, N. Karmarkar discovered polynomial time algorithm that is one of the boom in interior point methods of optimization. There after, these advancement led to the discovery of new optimization techniques.

## 1.2 OPTIMIZATION DESIGN

The simple optimal design needs to be attained by taking the different sets of random solutions then best solutions among them are find out after comparing them. A single problem formulation cannot be applied to all designs since the parameters of problem design vary from product to product so different techniques are used for different problems. A mathematical model is designed for elaborating the process of optimization as given in Figure 1.1. The procedure begins with choosing the proper design variables which are varied during the whole process of optimization. The problem may have many variables which are much sensitive and important for the functioning of any design. The design variables cannot be chosen randomly, these must have to satisfy specific functional requirements. So proper choice of design variables is important. Next step begins with setting the constraints which represents some kind of functional relationships with design parameters. This is basically the condition which have to be satisfied for the feasible design. Constraints are optional, they are not necessary. After that objective functions are defined which is basically either minimization of cost or maximization of lifetime of product. It can be said that maximizing desired value and minimizing the value of undesired ones. The optimization problem may include single as well as multivariable objective function. Then maximum and minimum boundary condition values are defined. After that, the best optimization algorithm fit for the design is chosen since design problems vary from one problem to another. At last, solution/solutions value are obtained.

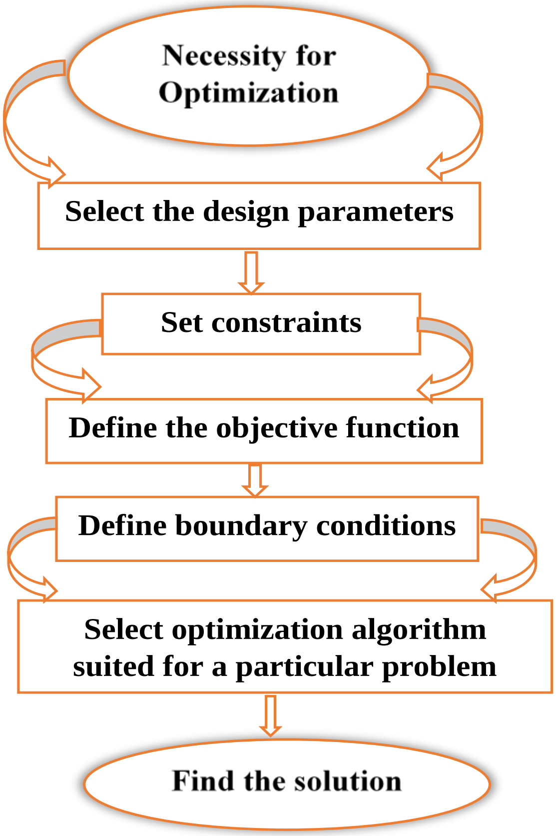



Figure 1.1 Model for design of optimization

## 1.3 CLASSIFICATION BASED UPON OPTIMIZATION PROBLEMS

The optimization problems can be categorized based upon the constraint type, nature of equations used, design variables and number of objective function values. This classification is elaborated below in detail:

– Based upon Constraints- The optimization problem is said to be constraint problem if the problem is subjected to one or more than one constraint otherwise it is defined as unconstraint problem.
– Based on nature of equations used- The optimization problem can be classified into linear, quadratic or non-linear programming problem. In linear programming problems, a problem is said to be linear if the constraints and objective function of the problems are linear function with respect to design variables. Here, the objective function, equality constraint and inequality constraint are linear. In quadratic programming, objective

function is quadratic while constraints function are considered as linear function of the optimization variables. In non-linear programming, if any of function either constraint or objective function is non-linear then optimization problem can be considered as non-linear function. This optimization problem is most commonly used and all problems are generally the case of non-linear function only.

- Based on design variable- Optimization problems can be grouped into integer programming and real valued programming problems based on value of decision variables. In integer programming model, the design variables could take only integer (discrete) values. While for real valued programming, problem is allowed to maximize or minimize a function by taking real values within certain range value.
- Based on objective function- Here, objective function is either single or multi-objective problems. In single objective problem, there is only single value of objective function. Multi-objective problem can be denoted as:
  Find the value of x that minimizes

$$f_1(x), f_2(x), f_3(x), \dots \dots \dots, f_n(x;) \qquad \text{subject to} \qquad g_i(x) \leq 0, i = 0,1,2 \dots \dots \dots, k$$

**1.4 BASICS OF OPTIMIZATION**

Optimization is the process of maximizing or minimizing the objective function. It is the method of maximizing the required factors and minimizing undesired factors in order to maximize profit. It is way of finding best solution under given bounded constraints. The main goal of optimization is to increase the efficiency and reduce the cost of production. Optimization methods are defined as the way of obtaining optimal solution which responds to the certain objective function. The problem solving of real word applications motivated the researcher to discover new optimized algorithms. Since the traditional optimization algorithms like mathematical approaches, find it more difficult to solve practical problems which involves various design variables. So, it has been an active research field and many researches are carried out day by day for the advancement of it. Mathematically, optimization can be represented as:

$$\begin{array}{lll} Minimize & f(x), & x = [x_1, x_2, \dots \dots \dots, x_n] \\ subject\ to & g_i(x) \leq 0, & i = 1,2, \dots \dots, m \\ & h_i(x) = 0, & i = 1,2, \dots \dots, p \end{array}$$

where, f(x) is the objective function to be minimized, $g_i(x)$ is inequality constraint and $h_i(x)$ is equality constraint.

One of the most popular example of optimization is travelling salesman problem (TSP) in which salesman has to travel N cities by shortest route and within shortest time. The order or the way in which salesman travel do not matter. Each city is connected to other close city either by railways, roads or airplanes. Each link between the cities contain weights or cost that is attached to it. As it can be seen from Figure 1.2 that there are four cities namely A, B, C and D and all are connected to one another. So, the main aim in this problem is to apply optimization method by which salesman could travel as to cover shortest path in shortest time.

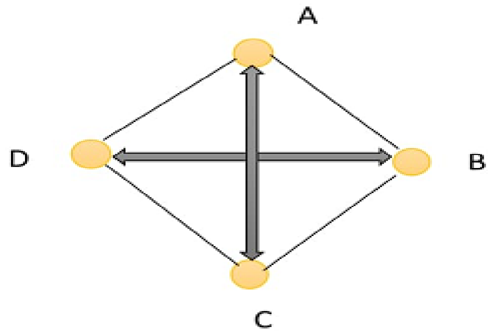



Figure 1.2 Figure showing cities connected for TSP problem

There are many methods of optimization being presented in past years. These methods are categorized into two groups: deterministic algorithms and the stochastic algorithms. Deterministic algorithm deals with certain rules for taking the solution from one point to another. So these algorithms are suitable to only some kind of situations. Hence, these algorithms are only good for unimodal functions i.e., the functions which are having only one global optima. So to overcome the drawbacks of deterministic algorithms, stochastic algorithms were proposed. Stochastic algorithms follow probabilistic rules so these are much popular as compared to deterministic algorithms. These algorithms are good in finding the solution for multimodal functions i.e., functions having many local optima solutions. Some of the examples of stochastic algorithms are genetic algorithm (GA) [1], spider monkey optimization (SMO) [2], biogeography based optimization (BBO) [3], harmony search (HS) [4] and so on.

Stochastic algorithms can be divided into two categories: heuristic and meta-heuristic algorithms. Heuristic algorithms find the high quality solution by using hit and trial method within computational time. Meta-heuristic algorithms are advance version of heuristic algorithms. Meta-heuristic algorithms are becoming popular due to their capability in solving both continuous as well as discrete problems. These algorithms have the feature of global search for finding optimum solution and even these algorithms are not dependent on gradient search method. Meta-heuristic algorithms used both local search as well as randomization. Due to this, these algorithms are much better for global optimization problems. The detailed study of these algorithms are discussed below.

## 1.5 NATURE INSPIRED ALGORITHMS (NIA)

It has been a well-known fact that nature is a main source of inspiration in all fields of technology. In the field of optimization, nature has played a vital role for the proposal of many algorithms. Nature inspired algorithms are inspired from the physical or biological phenomenon. These algorithms are either inspired from collective behavior of ants, birds, bats or are based on the concept of Darwinian Theory of evolution. These algorithms are becoming advanced research field for researchers.

### 1.5.1 Need of Nature Inspired Algorithms

The main choice of appropriate optimization algorithm method depends upon the optimization problems. There were many classical methods in the past like direct method, gradient method, linear programming method, interior point method and so on that were used for solving the optimization problem. But for the real world applications, problems may include complexities like non-linearity, continuous or mixed variables, discrete variable, discontinuity and so on. The search area/space is so large that it is not possible to find the optimum solution within reasonable time. Hence, in these situations, classical optimization methods are not suitable. Therefore, various advanced optimization algorithms like nature based algorithms are used.

### 1.5.2 Components of Nature Inspired Algorithms

There are basically two major components which affects the searching ability of any meta-heuristic algorithm: Exploration and exploitation. Exploration (diversification) is searching around the unknown area and exploitation (intensification) is exploiting the solution obtained by diversification. Exploration leads to find better or even novel solution structures. While on the other hand exploitation leads to only small changes to already searched individuals so it may sometimes get similar solutions or may even get better solutions. Exploration ensures that solution does not get struck in local optima while exploitation mainly concentrates around best solution for the convergence speed. The concept of local and global optima is shown in Figure 1.3. Many previous studies suggest that exploration should be carried out first in order to examine the whole search area then exploitation at later stage for improving the solutions of exploration [5]. A good optimization technique is the one which accurately balance between these two components [6]. Less exploration and more exploitation will trap the solution in local optima. While on the other hand more exploration and less exploitation will decrease the converging speed of the solution and degrade the performance of the solution.

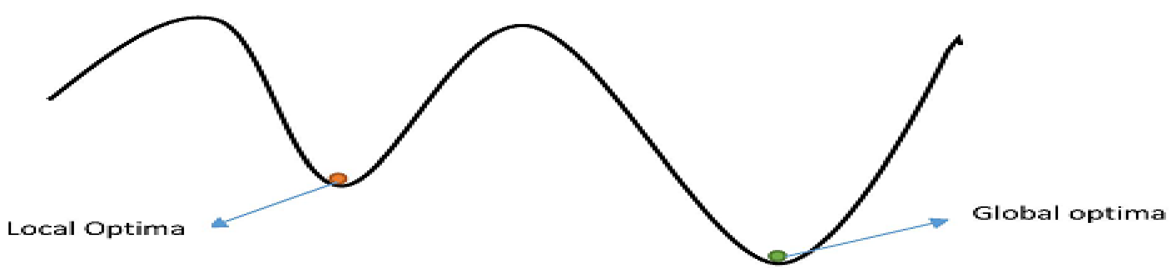



Figure 1.3 Concept of local and global optima

### 1.5.3 Types of Algorithms

NIA can be categorized into three ways: Evolutionary based algorithms, Physics based algorithms and Swarm intelligence based algorithms as shown in Figure 1.4. Evolutionary algorithms are basically inspired by the law of natural evolution phenomenon. The main advantage of these algorithms is that best individuals from the population are combined and passed on to the next generation. This helps algorithms to optimize the solution with course of generations. These algorithms are considered as general purpose algorithms that are fruitful for finding the near optimal solution to real valued and numerical test problems. These algorithms are applied for the solutions of problems for which analytical and exact methods could not find optimum solution within the accurate computational time.

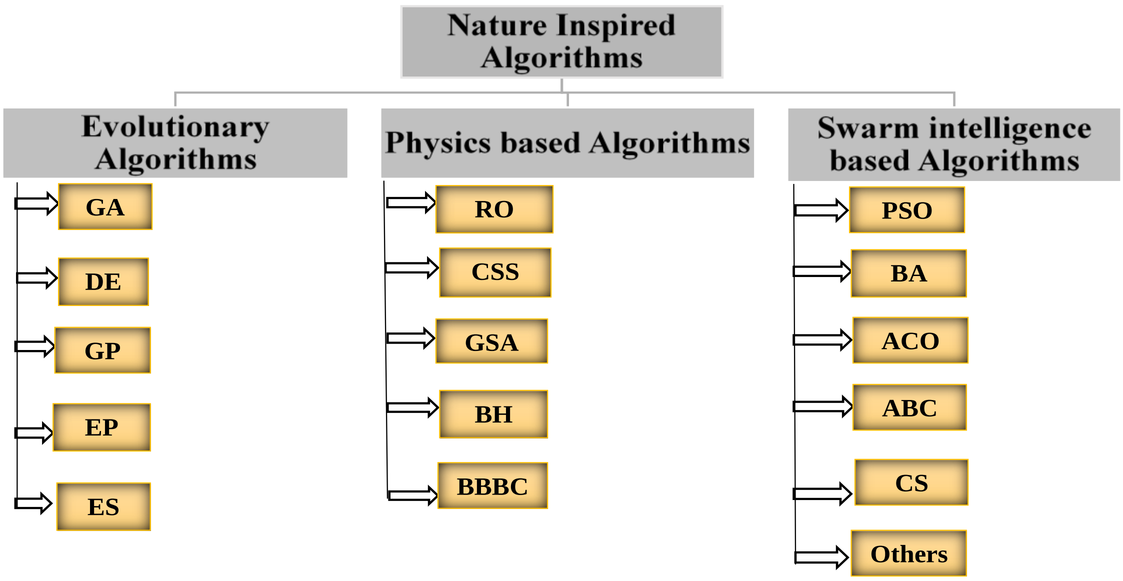



Figure 1.4 Figure showing nature inspired algorithms

There are many reasons due to which these algorithms are well suited for many application like:

- EA methods gives extensive varieties of constraints which are generally not provided by any other traditional algorithms.
- These methods can be generally extended to multi-objective optimization techniques.
- These methods are much suited for the optimization of parameters since these are quantitative based methods.
- EA can easily hybridized with other optimization algorithms.

The most famous algorithm in this group is GA which was introduced by Holland [7] in 1995 and based on Darwinian theory of "survival of the fittest". Here, the new population is formed by the process of crossover and the mutation which guarantees that new solution will be better than the previous generation solution. Other famous algorithm in this category is flower pollination algorithm (FPA) [8]. This algorithm is based upon the pollination of flowers. Pollination is attained by either by cross-pollination or by self-pollination. In cross-pollination, pollination is performed by pollen from the flowers which are from the different plant, on the other hand, self-pollination is pollen either from same flower or it can be different flower of same plant. Other algorithms in this category include differential evolution (DE) [9], genetic programming (GP) [10], evolutionary programming (EP) [11], evolutionary strategy (ES) [12] and so on. Thus in this thesis, work has been done on one of the most popular evolutionary algorithm (FPA). The detailed study of this algorithm is carried out in Chapter 2 while Chapter 3 deals with an improved version of FPA.

Physics based algorithms are inspired from the phenomena of physics which are based on gravitation, electromagnetic theory, Newton theory, quantum theory, electrostatics and so on. The famous algorithms in this group include gravitational search algorithm (GSA) which was proposed by Rashedi *et al*. [13]. This algorithm is based upon the law of gravitation and interaction of masses. Each agent is considered as object and masses are the criterion for measuring their performance. Each agent attracts towards the other by the gravitational force and it leads to the movement of solution towards the heavier mass which is a good solution. Hence, in this way heavier massed are attracted towards the lighter ones which helps the algorithm in exploiting the solution in better way. Other algorithms in this category include ray optimization (RO) [14] algorithm, charged system search (CSS) [15], black hole (BH) [16] optimization algorithm and big bang big crunch (BBBC) [17].

Swarm intelligence based algorithms are based upon social behavior of group of animals. Swarm intelligence can be defined as attempt for designing the algorithms and problem solving devices that is inspired by social collective behavior of insect colonies and different other societies [18]. Swarm is generally referred to a group of any interacting individuals or agents [19]. Swarm intelligence is based upon the population of self-organized agents or individuals which are locally interacting to one another and even to their environment.

There are two basic concept that are sufficient and necessary for showing SI behavior:

1) Self-Organization-It may be considered as the set of dynamic mechanisms that results due to structures in global level through the interaction between low-level components. All these mechanism are based upon certain rules which ensures that interaction between components occur at local level information without considering the global level. There are four properties of self-organization according to Bonabeau *et al.*[18] that are:

Figure 1.5 Figures showing examples of SI [20]

   a) Positive Feedback-It is basically the rule of thumb that encourages the creation of structures. Dances of bees in ABC can be considered as one of the example of positive feedback.
   b) Negative Feedback-It is used for balancing the negative feedback and for the stabilization of collective patterns. Negative feedback is needed for avoiding the saturation which occurs due to exhaustion of food sources, competition for the food sources and so on.
   c) Fluctuations-It can be defined as randomness or random walk between the swarms which is main criterion needed for increasing the diversity and finding new solutions.
   d) Multiple Interactions-It is the main component which is needed for mutually tolerant agents that makes the use of results obtain from their own as well as other individual's activities.

2) Division of labour- There are various tasks in swarms that are achieved simultaneously by the groups of specialized individuals. This concept is known as division of labour. The tasks which are achieved by specialized individuals are better and much efficient than the sequential task performed by group of unspecialized individuals [21].

The advantages of SI based algorithms are given below:

- These algorithms have less parameters to adjust.
- These algorithms preserve the information obtained from the search space with respect of iterations.
- These algorithms are much easier for implementing.
- SI based algorithms have the space for memory for saving the best solutions.

One of the example of swarm is the bees swarming around their hives as shown in Figure 1.5. But these can be extended to other systems having similar behavior like fish, ants, birds and so on. The main algorithm in this category include particle swarm optimization (PSO) developed by Kennedy & Eberhart in 1995 [22]. This algorithm mimics the behavior of birds flocking. The particles move around the search space for finding the optimal solution. In this algorithm, the current solution updated its position with respect to the position of global best and personal best solution. Other famous algorithm is ant colony algorithm (ACO) developed by Dorigo *et al.* [23] which is based on the ants intelligence in finding the shortest route towards their food. Other algorithms include bat algorithm (BA) [24], artificially bee colony (ABC) [25], cuckoo search (CS) [26] and others [27, 28].

SI is one of the fast-growing research area but still it suffers from certain drawbacks:

- Irregularity of particles-These algorithms find difficulty in predicting the future behavior of individual agents. It can also be said that even small change in a rule may lead to huge change in group behavior of agents. The behavior of individual agent cannot predict the behavior of whole group.
- Time Limited Applications-Since the path of SI agents are not pre-defined but it is becoming prominent. Hence, these algorithms are not generally useful for time critical applications like decisions which involves time critical situations, on-line controlling of the system and acceptable performance within required time such as maintaining the nuclear reactor temperature.
- Stagnation-The SI algorithms generally lacks the central co-ordination so these algorithms suffers from premature convergence and stagnation problem. It could be well explained by the concept of ACO when all ants follow the same path and visit the same path so it leads to stagnation problem.
- Parameter tuning-Tuning the parameters of these algorithms is one of the most critical drawback like most stochastic algorithms. It is because the parameters of these algorithms are problem dependent.

## 1.6 APPLICATION OF OPTIMIZATION ALGORITHMS

Optimization Algorithms are applied in different real applications and different fields of engineering.

- Electrical Engineering-In the field of electrical, these algorithms are applied for economic load dispatch problems, power point tracking for the photovoltaic systems, sizing of var compensator in case of power systems, solar radiation forecasting and so on.
- Mechanical Engineering-In the field of mechanical engineering, these are applied for the designing of truss and steel structures, biodiesel engine modelling, cell formation problems, etc.
- Image processing- Applying these algorithms for image processing, these algorithms are used for feature selection, enhancement of satellite imaging, thresholding for grey level segmentation and so on.
- Communication Engineering- These algorithms are applied for wireless sensor networks, frequency fraction reuse used for Orthogonal FDMA (frequency division multiplexing access) and so on.

There are also many other engineering field like control systems, antenna design, computer science and many more in which these algorithms are generally applied.

## 1.7 ANTENNA ARRAY

Antenna is meant to couple the transmitter to particular medium, e.g., free space. In other words, antenna can be defined as transitional structure between the guiding medium and the free space. It is basically act as radiator to radiate electromagnetic waves from transmitting antenna side to receiver antenna side then converts these electromagnetic waves to electrical signals which are then applied to input or receiver input side. Antenna can also be defined as a transducer which is used for converting radio-frequency wave into the alternating currents and vice-versa. Hence, antenna can be said to be a sensor, transducer, electromagnetic radiator and impedance matching device. It is used in various applications like radar, mobile, wireless and satellite communication.

There are certain parameters which characterize the properties of an antenna like radiation pattern, radiation pattern lobes, beam width and so on. Radiation pattern can be defined as the graphical representation of properties of radiation as function of directional/space coordinates. It basically shows the lobes pattern at various angles. Hence, main lobe is the lobe in the direction of maximum radiation. Sides lobes level (SLL) are unwanted radiation that are in undesired direction. Beam width is an angular separation between the two same points in opposite side of the main beam. One of the popular beam width is first null beam width (FNBW) that can be said to be separation between first nulls of pattern. The main goal should be to minimize the effect of SLL and FNBW. Generally, a single element has wide radiation pattern and also the gain of single antenna is not sufficient for any application. The gain of an antenna can be increased by increasing the dimension of antenna which makes the antenna bulky and this method also affect the operating frequency which is not desired in most of the applications. Hence, the most reliable and widely used method is an antenna arrays.

Antenna Arrays are the set of similar antenna collectively operating as the single element of radiation either in geometrical or electrical configurations. These arrays are mostly fed coherently. These arrays have capability for providing change in direction of radiation pattern with the help of excitations provided by antenna elements. Unlike single element of an antenna which have fixed pattern for radiation, antenna arrays' pattern can be altered by the excitation of antenna elements by providing current excitation. This characteristics help the antenna array to choose desired pattern instead of changing their dimensions. Arrays also have the characteristic of delivering diversity gain in multipath reception of signals. Radiation pattern of antenna array can be modified with the help of various parameters. These several parameters are basically inter spacing of element, geometrical configurations (circular, linear, spherical and so on), excitations (amplitude & phase) and so on. By controlling the received signals through different single antenna elements by different criterion, antenna arrays can be applied to many signal functions like enhancement of gain, spatial filtering, suppression of interference and so on. Antenna array is suggested for improvement of various communication applications by enlarging coverage area, increment of channel capacity, tailoring the beam shape and so on [29]. Random array design can cause many problems like electromagnetic radiation pollution and wastage of power which cause disastrous for limited power communication devices. So, array must be accurately designed so to avoid these problems.

Antenna array are widely used in communication either radar, satellite, mobile or wireless communication for improving signal quality which leads to increment in the capacity, coverage area and link quality. These system performance depends mainly on the design of arrays [30]. Antenna arrays are of many types based upon their geometry such as linear, circular, concentric, planar array and so on. Linear antenna array (LAA) is one of the simplest array whose elements of antenna are arranged in straight line. In circular array, elements are arranged circularly around the perimeter of the circle. These arrays have the advantage of providing 2 D scan as well as have less effect of mutual coupling, that's why these arrays are widely used.

Concentric circular arrays are advance version of circular array where there are multiple rings of circular array. In planar arrays, elements of antenna are arranged around some planar surface like rectangular surface. The main aim of these antenna array design is to reduce SLL as much as it can be, while preserving the main beam gain to be same. The goal of array synthesis is to design the physical layout of array pattern which gives the closest radiation pattern as the desired one.

The classical based optimization algorithms are not appropriate for the antenna arrays design due to many reasons like i) existence of non-differentiable objective function, ii) convergence problem and complexity of algorithms, iii) these algorithms are extremely sensitive to starting points since number of variables are large hence solution space will also be large, iv) piece-wise linearity is not possible for cost approximation, v) frequently get convergent to local optimal solutions and repeatedly visiting the same suboptimal solutions, vi) multi-modal solutions are not easily tackled by classical methods. Therefore, nature inspired algorithms are needed for the synthesis for antenna arrays. Hence in present work, an improved version of most popular nature inspired algorithm (FPA) is used for the optimization of non-uniform CAA and CCAA. The detailed geometry and background of CAA and CCAA are discussed in chapter 4 and synthesis of these arrays using improved FPA is analyzed in chapter 5.

### 1.8 OBJECTIVES

There are certain objectives of this work which are discussed below:

a) Study of literature in detail.
b) Design of an improved flower pollination algorithm.
c) Implementation of Improved Flower Pollination algorithm over benchmarking functions
d) Implementation of improved algorithm for the synthesis of circular as well as concentric antenna array for suppression of side lobe.

### 1.9 THESIS OUTLINE

The general overview of the present work can be elaborated as follow: Chapter 2 presents the extensive literature survey of various optimization algorithms. This chapter also deals with nature inspired algorithms and their application for the synthesis of circular and concentric circular antenna arrays. Chapter 3 describes about problem formulation for CAA and CCAA. Chapter 4 deals with original FPA and enhanced version of FPA (IFPA). In chapter 5, experimental results and its study are given. This chapter discusses about test function evaluation, analysis of population as well as dimension sizes and comparison with other existing algorithms. The results for practical application are also analyzed in this chapter. Last chapter 6, deals with the conclusion obtained from the results and its future scope.

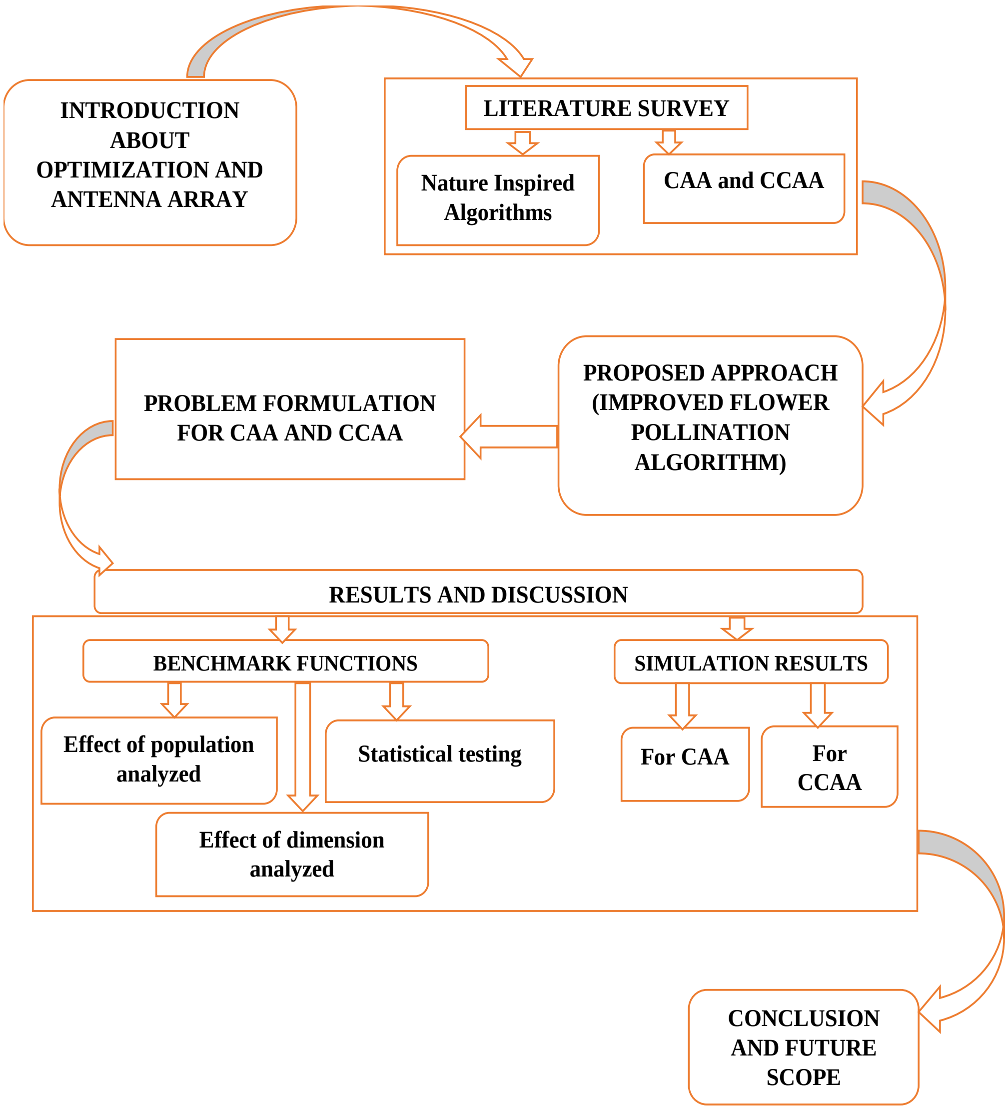



Figure 1.6 Outline of work

# CHAPTER 2
# LITERATURE SURVEY

## 2.1 LITERATURE SURVEY OF NATURE INSPIRED ALGORITHMS

Swarm intelligence (SI) concept was firstly introduced by Beni & Wang [31] in 1993. It is basically the collective groups of interacting agents. SI has become the famous research interest area for many scientists in latest years. There are some examples of SI like ACO that can be thought as swarm in which agents are ants. Similarly, an immune system may be considered as swarm of the cells and crowd can be thought to be swarm of people. In the same way, PSO is considered as swarm of birds flocking. This section deals with some of the SI based algorithms and their related works.

Particle swarm optimization is one of the population based stochastic optimization algorithm which was developed in 1995 by Kennedy & Eberhart [22]. This algorithm is inspired from social behavior of flock of birds. The birds in flock are represented as the particles. Since a single particle has no power for solving any problem, so particles interact with each other. These particles move in multi-dimensional space with velocity by considering the distance between current best and previously best location (pbest). This algorithm even update the velocity by seeing the distance between the current best and global best location (gbest). In this way, positions of the particle are updated. There are mainly three parameters, exploration, exploitation and convergence speed which are taken into account for the improvement of any algorithm. Many researches are carried out by the researchers for improving the performance of basic PSO. Liu *et al.* had replaced the concept of position and velocity by probability density function i.e. wave function ($\Psi(x, t)$) [32]. The particle moves by considering the effect of mean of all best position of particles. Cauchy distribution is used as a mutation operator because of more chances of escaping from local optima solution. In this way, the proposed approach was much better in improving global search capability. In [33] used the concept of chaos searching strategy for avoiding the swarm particle to entrap into local minima. Logistic mapping chaotic function is used in this paper to avoid premature convergence and algorithm stagnation. This function is also used at the initialization of algorithm for distribution of the particle in more symmetrical way. Levy flight concept is introduced by Jensi *et al.* [34] in velocity equation. By the incorporation Levy flight in velocity updating, the swarm particle takes long jump due to randomness. In this way, swarm diversity, exploration capability and convergence efficiency of the algorithm are increased. In this work, levy flight parameter, $\beta$ is taken as 1.5. In [35], PSO has been modified by overcoming its drawback of stagnation and slow convergence. Firstly, PSO has been incorporated with ABC inspired fitness function. Secondly, self-balancing of position was applied in both fitness based as well as PSO search phase. For PSO based search process, new position is updated by using previously best and globally best solution. In globally best solution, weight is updated by using exponentially decreasing function. This helped in improvement of exploration (diversification) of the solution. While for fitness based search method, solutions are updated by using their fitness values. Here, best fit members are given more chances for updating their position as compared to less fit members. For checking the reliability and efficiency of SBPSO (self-balanced PSO), results are tested on twenty standard benchmark functions and compared with different other algorithms. Due to attraction phase in original PSO, all particles move towards the pbest and gbest which lead to similarity between the particles. In this way diversity between the particles is decreased. In [36], first modification is done to increase the diversity by generating the particle either by velocity updating equation or position updating equation. Second modification is done to avoid premature convergence by introducing the concept of neighborhood searching strategy by local as well as global neighborhood search strategy. In this way, balance between exploitation and exploration is maintained. The original PSO had been modified with exploration technique [37]. In this modification, the current position of particle was updated with extrapolating the optimum global best position of the particle by redefining the search towards the global best value. This is done with the help of adding two explorative coefficient values in position equation. The proposed approach was able to achieve faster convergence and deterministic for getting global best solution.

FA basically mimics social behavior of the fireflies. FA concept was introduced by Yang [27] in 2009. The fireflies find their prey or communicate with each other with their flashing pattern. Less bright firefly is attracted towards the brighter one and attractiveness of firefly decreases with increase in distance. The chaotic maps were used for tuning the parameters of FA [38] which increases the performance of algorithm. In this proposed algorithm, light absorption coefficient ($\gamma$) is tuned with different chaotic maps and sinusoidal map is found to be better than all. Attractiveness coefficient ($\beta$) is also tuned with different chaotic maps but here gauss map shows better performance when being compared with other maps. Since, attractiveness coefficient is much efficient parameter as compared to $\gamma$ so gauss map is considered best map. Original FA was enhanced by using the concept of opposition based learning [39]. In this concept, a candidate solution and its equivalent opposition candidate solution is generated which helps the algorithm to increase its diversity. So in this work, worst firefly is removed by using opposition learning and it is replaced by new generated firefly. This concept helps the algorithm in escaping from local optimal stagnation and increase the diversity of population.

For checking the accuracy of modified FA, it is tested with sixteen standard benchmark functions. Wang *et al.* [40] had proposed three different modifications for the enhancement of FA. Firstly, dynamic parameters, α and β are adjusted by modifying their equations. Secondly, a random model of attraction is introduced in which a firefly is randomly attracted towards another at most once. This will reduce the computational complexity time and convergence speed of an algorithm. Lastly, three neighborhood search is carried out for improving the exploitation by local neighborhood search and two global search for the enhancement of exploration. A Cauchy mutation operation is introduced in global search for helping the algorithm to jump out from local minima. These neighborhood search will help to find out most fit candidate solution for the algorithm. The proposed approach was benchmarked with twelve standard benchmarking functions and compared with different algorithms. There are different application fields in which FA has been applied. The FA is modified by using exponential decreasing randomization factor in order to increase the convergence speed of the algorithm. FA equation is also updated by the introduction of scaling factor. In this way, the proposed approach was used for tuning the parameters of controller that is smith predictor [41] for integrating and unstable the delay process. The authors [42] have modified FA by updating the brightness of firefly by modifying the brightness equation. The proposed approach was then used for solving non-minimal phase systems of PID controller. Here the objective function is to reduce peak overshoot, setting time and ITAE. The modified approach is then compared with original FA which showed the effectiveness of proposed approach.

Bat algorithm (BA) is a metaheuristic algorithm that is based upon the echolocation nature of micro bats developed by Yang in 2010 [24]. 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. It can be elaborated that if a bat is nearer to the prey then its pulse rate increases and sound pulse decreases and if bat goes away, pulse decreases and sound increases.

There are many modifications done for improving the performance of Bat Algorithm. Generation of solution, exploration and exploitation are some of the factors that are taken into account for improvement of algorithm. The exploitation capability of bat algorithm has been improved by removing the premature convergence and adding a controlling factor for new step size [43]. Zhu *et al.* [44] has also tried to remove premature convergence and diversity of the solution by introducing the concept of quantum based bat algorithm and using the mean as the guiding parameter. This method is also benchmarked on twenty four functions and compared with other standard algorithms. The concept of directional bat [45] was introduced by emitting the pulse in two different directions to increase exploration capability. The author has also tried to balance exploitation and exploration by loudness and pulse rate modification and benchmarked with twenty functions and even compared with different algorithms. For improving the explorative capability of BA, pulse rate (r) and loudness (A) were equalized for each dimension [46]. So, r and A are assigned separately for each dimension. Hence, each dimension could perform exploitative and explorative capability simultaneously. BA can also be modified by using mutation operation in global search and Levy flight for updating the position [47]. Levy flight is good for diversity so it will help the algorithm to jump out local minima. Also, the pulse rate is modified so that r will only be updated when the best solution from current iteration will be better than best solution of final iteration. Hence, r will help increasing the diversity since new and new positions are explored. Many other modifications [48-50] are done for overcoming the drawbacks of basic BA.

BA has been applied in different real time problems for showing the efficiency of an algorithm. BA was used for optimum design of the reinforced concrete plane frames structures [51] and applied for three different frame structures.  Kaveh & Zakian [52] have improved BA by updating the scaling factor for step size in local search equation for reducing the computational time. Then this improved BA was applied for optimizing the size of skeletal structures and different examples are chosen for demonstrating the effectiveness of this algorithm. The original BA was enhanced by the concept of bacterial foraging strategy [53] where movement of bat is updated by chemotactic movement of the bacterium. This enhancement of BA increased the convergence time and applied in wireless sensor network for localization of nodes. It is compared with original BA which shows the effectiveness of enhanced algorithm and its success rate. Other modifications are also done and applied in different fields [54-56]. This algorithm has also been hybridized with other algorithms for improving the efficiency of original BA. In [57], BA has been hybridized with harmonic search algorithm which helps the algorithm to avoid in getting stuck in local optima and increase convergence speed. For checking its effectiveness, the hybrid algorithm is benchmarked with different functions and compared with many other algorithms. Differential Evolution is hybridized with BA [58] and tested on different benchmark functions and compared with original BA. This hybrid algorithm helps in the improvement of original BA. There are many other hybridization [59-61] done for improving the original BA.

Whale optimization algorithm (WOA) is latest swarm intelligence based algorithm developed by Mirjalli and Lewis in 2016 [62]. This algorithm is inspired by social behavior of humpback whales. This algorithm basically uses three operators to find prey, encircle it and forms bubble net foraging pattern. Kaveh and Ghazaan [63] had proposed enhanced WOA for improving the exploitation and exploration capability of the algorithm. Also Levy flight was introduced in WOA [64]. This modification was done in order to improve exploration ability and to avoid premature convergence. WOA has been applied successfully in many problems like optimizing vehicle route consumption [65], optimal mobile robot path planning [66], economic emission dispatch problem [67] and so on.

Physics based algorithm inspired from the phenomenon of physics and mimics the rules of physics. Big bang big crunch (BBBC) [17] is one of the physics based algorithm based upon the big bang and big crunch phase. The random points are generated by big bang phase and these points contract to a single point with the help of either minimal cost or center of mass approach by big crunch phase. The main drawback of this algorithm is premature convergence. This disadvantage was removed by Atalas [68] by uniform population method in which initial populations were evenly spread in search area. Chaotic maps were also used to overcome this drawback by increasing the diversity of population. There are many application areas in which BBBC has been applied like optimal design of steel frames [69], optimum design of reinforce concrete frames [70] and so on. Gravitational search algorithm (GSA) [13] is another physics based upon law of gravitation i.e., all objects are attracted towards one other by law of gravity. Sarafrazi *et al.* [71] have improved the GSA by using 'Disruption' operator in order to explore and exploit search area effectively. The applications in which GSA was applied include filter modeling [72], optimal power flow [73] and so on.

Evolutionary algorithms (EA) are population based stochastic algorithms. These algorithms need very less knowledge about the specific problem and even can be applied to vast areas of applications. These algorithm can be applied for the multi-objective, multi-modal, discontinuous and even noisy problems. According to the rule of these EA, only those algorithms will survive in future which will meet the certain specific criterion and remaining population will die. Based upon the fitness criterion, EA will select according to certain selection probability and will move to next generation for the survival of fittest solution. One of the most famous and widely used algorithm in this category is genetic algorithm (GA) [7]. This algorithm was proposed by Holland and is based upon the Darwinian theory of evolution. This algorithm is based upon three rules i.e., selection, crossover and mutation. Selection rule basically chooses the individual that will contribute to future generation. Crossover rule generally combine two parents for forming the children for future generation. Lastly, mutation randomly changes any individual parent. GA has proven its worth by applying in various problems like flowshop sequencing [74], generalized assignment problem [75], antenna design [76], multiprocessor scheduling [77] and so on.

Other popular algorithm under this class include flower pollination algorithm (FPA). FPA is one of the recently developed meta-heuristic algorithm [8]. This algorithm is inspired from pollination process of the flowers and mimics the reproduction of flowering plants with the help of pollination. Pollination of flowers can take place in two forms i.e., biotic and abiotic. Majority of the pollination takes place with the help of biotic pollination which require pollinators i.e., bees, insects and so on for transferring the pollens. While abiotic pollinator perform minimum pollination in which there is no need of pollinator. Here diffusion and wind helps the flower in pollination. There are many modifications done for increasing the performance of original FPA. In paper [78], FPA has been modified by using three strategies, firstly the improvement in local neighborhood search strategy to improve the exploitation capability. Secondly, dimension by dimension improvement is done in local search method in order to increase convergence and solution quality. The last strategy is switching probability in order to proper balance between exploitation and exploration. Nabil [79] has tried to increase the exploitation capability through cloning method and introduced a new scaling factor ($\gamma_2$) for step size in local pollination. Through parametric analysis, the authors claimed that for $\gamma_2$=3 works well for most of the functions. The author also tried to increase exploration to some level by replacing lowest fitness solutions with random one and have taken switching probability as 0.8. In this article, there was no proper discussion for using p=0.8. In [80], the concept of local search phase through scaling factor is used so that FPA may not start moving away from the best solution and get stuck in some local optima. So the main aim of the author was to improve the exploitative ability of FPA. The proposed version was then applied for solving the economic load dispatch problem.

There are many real world applications in which FPA has been applied. The authors have applied FPA for minimizing the weight of the truss structures [81]. The proposed approach has been tested with three different weight minimization problems of 2D and 3D structures. The authors have used modified version of FPA for solving the problem of OPF (Optimal power flow) [82] by finding best operating point in the field of power system. FPA was modified using opposition based points for initial solution and removing the concept of switching probability by combining global and local search equations. In this way, modified FPA is able to converge faster in less time. Other works on FPA include linear antenna arrays [83], combined emission and economic dispatch problem [84], wireless sensor networks [85], optimizing the relay coordination in electrical networks [86] and many more [87, 88]. There are many hybridization used for improvement of FPA. In [89], authors hybridized FPA with GA for improving search accuracy and tested this method on seven benchmarking functions for showing the competitiveness and superiority of the solution. PSO has been hybridized with FPA [90] and tested on several benchmark functions which shows its reliability, convergence and effectiveness for solving the problems. Simulated Annealing (SA) has been hybridized with FPA [91] and tested on different benchmark functions and compared with CS, PSO and FA which shows the better convergence rate of hybrid algorithm. Hence, there are many more other hybridization [92, 93] done with FPA for its improvement.

## 2.2 LITERATURE SURVEY OF ANTENNA ARRAY

Antenna array is basically the assembly of radiation element in geometrical and electrical configuration. For providing the highly directive pattern, fields of the element array should constructively interfere in required direction and destructively interfere in undesired direction. There are various method that can be used for shaping the required pattern of antenna such as excitation phase and amplitude of elements, displacement between elements and geometrical configuration of array (circular, linear, spherical, rectangular, etc.). The two most famous and widely used geometrical configurations i.e., circular and concentric array are discussed below:

### 2.2.1 Circular Array Antenna (CAA)

In circular array, elements are placed circularly around the ring. The circular array has advantages over other configurations like providing 2-D scan both in horizontal as well as vertical angular scan. These array can scan 360º horizontally without any distortion nearer to end-fire direction. Due to the absence of edge element because its geometry, the effect of mutual coupling is also very less. While synthesizing circular antenna array, the main task is to reduce SLL and minimize circumference of array element. The first meta-heuristic approach towards its design, was introduced by Panduro [94] with the help of real coded GA. This work is based on design of non-uniform circular antenna array by suppressing SLL having the fixed beam width constraints. GA helped in finding the optimal weights and distance between consecutive array elements. The proposed approach was tested on different array elements such as 8, 10 and 12. It is observed from the results that in comparison to uniform circular array, proposed approach reduces SLL up-to 57.49% and 41.40% for elements 10 and 12 respectively. This research is further proceeded by Najjar and Khodier [95] who implemented advanced PSO for maximum SLL reduction and smaller circumference. PSO was used for optimization of non-uniform CCA synthesis by adjusting the excitation and positions of elements. The results obtained are far better than GA. In [96], SA has been used for the reduction of SLL and beam width. Here, three cases (N=8, 10, 12) of CAA are considered and results demonstrated the effectiveness of SA when being compared with PSO. Further, work was proceeded by Singh and Kamal [97]. In this work, BBO was used for suppression of SLL with constraint of fixed main lobe beam width. The maximum reduction of SLL occurs for the case of N=10 when compared with GA. In [98], FA was used for the reduction of SLL with constraint of fixed major lobe. FA was used for determining optimum weights and positions for the case of CAA and optimum set of weights for the case of CCAA. The results showed the superiority of suppressing SLL when compared with GA, PSO and EP. Bacterial foraging optimization (BFO) algorithm is used for designing of circular array [99]. It is used to find the optimal values for the current excitations and distance in between array elements. The aim of using BFO is to maximally reduce the interference of side lobes with main lobes and also to improve the first null beam width (FNBW). The simulation is done on 8, 10 and 12 element circular arrays. The results of proposed approach are compared with other meta-heuristic algorithms like GA, PSO and SA. The radiation plots show that BFO is better performance in comparison to others. In [100], CSO was used for optimal designing of non-uniform single ring CCA and three ring CCAA. Three cases for CAA having elements 8, 10 and 12 and two three ring CCAA with set of elements 4, 6, 8 and 8, 10, 12 were considered. Results demonstrated that CSO provides best SLL reduction as compared with other standard algorithms. Das *et al.* [101] have used RGA (real coded genetic algorithm) to get the optimum location for the elements so as to get maximum suppression of SLL with least increment in FNBW. In this paper, three circular array with 8, 10 & 12 elements are used. Different cases like asymmetrical circular array with uniform angular spacing and non-uniform angular location of circular array are considered. Results demonstrate that SLL considerably reduced to much lower limit while vanishing of nulls.

### 2.2.2 Concentric Circular Antenna Array (CCAA)

A CCAA is an advance version of circular antenna array. A CCAA consists of concentric circular rings with different radii and each ring is having different number of elements. These arrays have an advantage that they can be used in narrowband as well as broadband beam forming application. In [102], CCAA using PSO with constriction factor for reducing the SLL without and with the central feeding element were discussed. Ten three rings (M=3) of CCAA are considered with non-uniform excitation and fixed element spacing of 0.55λ for first ring, 0.61λ for second ring and 0.75λ for third ring. Results showed that elements having central feeding element is much better in reducing SLL. Even, antenna array having elements 4, 6 & 8 with central feeding element gives much reduction in SLL. The results of PSO are also compared with binary GA that showed the better performance of PSO. In [103], CCAA design with or without central feed elements were examined for maximum reduction of SLL. In this proposed work, three algorithms EP, PSO and PSO with constriction factor and inertial weight approach were analyzed. EP has been proved to be best algorithm with maximum reduction of SLL for the case of 4, 6, 8 elements having central feed element.

Mandal *et al.* [104] applied hybrid EP for the synthesis for non-uniform CCAA design. In this work, different cases elements with central feeding and without central feeding were taken. Results revealed that non-uniform central feeding element provided maximum reduction in SLL and FNBW. CCAA with elements 4, 6, 8 gave grand reduction in SLL (-40.22 dB). CCAA design having non-uniform excitation and uniform separation between elements have been optimized with the help of BGA (binary coded GA) and BFO (bacterial foraging optimization) [105]. Here ten arrays having three rings in each array with inter-element spacing of 0.55λ for first, 0.61λ for second and 0.75λ for third ring were taken. The cases for uniform current excitation as well as non-uniform current excitations were also considered. The results showed that non-uniform current excitation gives more reduction in SLL as being compared to uniform excitation. The antenna array having elements 4, 6 and 8 in the rings give more reduction in SLL. BFO technique proved to be much significant in terms of SLL reduction as compared to BGA. Pal *et al.* [106] dealt with reduction of SLL and primary lobe beam width, BWFN by using hybrid DE and intensive weed optimization (DIWO). The distance between the elements are kept half of wavelength. The results of proposed approach was compared with other algorithms like PSO, IWO and DE. But DIWO outperforms other algorithms in terms of SLL and BWFN reduction.

The SI based algorithms are one of the most popular and widely used population based algorithms. PSO, FA and BA are most famous SI based algorithms. But due to certain drawbacks like irregularity of particles and limitation for limit critical situation based applications, these algorithms cannot be applied widely for large number of applications. Evolutionary based algorithms came into picture due to its applicability for noisy as well as discontinuous applications. One of the most popular EA is FPA that is discussed in section above. Since, very little work has been done on this algorithm. This algorithm even lacks in making proper balance between intensification and diversification. Hence, work has to be done in order to overcome its drawbacks. That’s why its enhanced version is proposed and used for the synthesis of antenna arrays.

# CHAPTER 3

# PROPOSED APPROACH

This chapter deals with details of proposed approach. Firstly, basic flower pollination algorithm is discussed and then its improved version is analyzed in detailed.

## 3.1 FLOWER POLLINATION ALGORITHM

Almost 80% of all plants species are flowering and are found to dominate earth's landscape from millions of years . These flowering species evolve through the process of pollination and this basic function helps flowers to reproduce. Pollination is fulfilled by pollinating species such as insects, birds and the transfer of pollen takes place through them. This process in flowers follow Darwin's theory of natural selection and provide optimal best flower. Some flower species attract only specified birds or insects for pollination leading to a specialized relationship between flower and pollinator, often called flower constancy [107]. Biotic and abiotic are the major forms of pollination. The pollination in which birds, insects, honey bees, bats and other animals are pollinators, that is called biotic. It consists of about 90% of flowering plants. Abiotic pollination on the other hand does not require any pollinators. Pollination mainly occur either through wind or diffusion. At least 200,000 varieties of pollinators are present in nature and are found to have flower constancy. This flower constancy helps flower pollinators to bypass some species and visit only specific species. It helps to maximize the transfer of pollens to flowers of same species and also increases reproductive capability of the same flower. The pollinators are also benefited such as in case of species like honey bee, they go only to specific species in order to collect nectar, hence maintain a minimum cost and intake more nectar.

Pollination can be classified among same or different species by either self or cross-pollination. In Self-pollination, fertilization takes place between different flowers of conspecific plants. This type of pollination keeps a particular species of flowers in the bio-system. It is also called as local pollination. Cross-pollination also called as allogamy, occurs in pollens of different flower species. Cross-pollination is generally biotic and mainly occurs over large distances and hence can be called as global pollination. These pollinators obeying Levy distribution and so follow a Levy flight behavior [108]. Bees, bats and flies follow biotic cross-pollinators.

All the above characteristics of plant can be idealized by the following rules [8]:

i. Biotic and the cross-pollination can be taken as global pollination which follows levy flights for carrying pollens from one place to other.
ii. The abiotic and self-pollination refers to local pollination.

iii. Flower constancy property is regarded as reproduction ratio which is being proportional to similarity between two flowers that are mainly involved in it.

iv. Switching probability, p$\epsilon$ [0, 1] helps to choose between local and global pollination.

Based on these four rules, FPA was proposed. For global pollination, fittest member can be represented as $g^*$ and fittest reproduction is the longest distance travelled by the insects. Mathematically, it can be represented as

$$x_i^{t+1} = x_i^t + L(g^* - x_i^t) \quad (3.1)$$

where, $x_i^t$ represents pollen *i* which is considered as the solution at iteration *t*, $g^*$ represents the best particle at each iteration. L is the levy distribution that represents the step size. Levy flight is used for mimicking the characteristic of steps taken by the insects for walking long distances. Hence, L can be updated as:

$$L \sim \left\{ \frac{\lambda \Gamma\ (\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda'}}\ (s \gg s_0 > 0) \right.$$

where, $\Gamma\ (\lambda)$ is gamma function and is valid for large value of s > 0.

The local pollination can be represented as

$$x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t) \quad (3.2)$$

where, $x_j^t$ and $x_k^t$ denoted the pollens between different flowers of the same plant, $\epsilon$ is basically normal distribution chosen between 0 and 1.

---

*Max or min objective function f(x), x=(x₁ , x₂ , … …..x_d )*
*Initialize the random population of n flowers*
*Find $g^*$ (best solution) from the population*
*Identify the switching probability p$\epsilon$[0,1]*
**while** *(iter < Maxgeneration)*
    **for** *i=1:n*
        **if** *rand<p*
        *find vector L(step size) from levy distribution*
        *Use global pollination by the equation* $x_i^{t+1} = x_i^t + L(g^* - x_i^t)$
    **else**
        *find the value of $\epsilon$ from uniform distribution*
        *Choose randomly the value of j and k from the solution*
        *Use local pollination by the equation* $x_i^{t+1} = x_i^t + \epsilon(x_j^t - x_k^t)$
    **End if**
    *Evaluate the value of new solutions*
    *If new solutions are better than old one, update them.*
  **End for**
  *Find the value of best Solution*
**End while**

---

Figure 3.1 Pseudo code of FPA

## 3.2 IMPROVED FLOWER POLLINATION ALGORITHM (IFPA)

It can be inferred from the literature that FPA suffers from many limitations. Firstly, switching probability is incapable of making a proper balance between exploration and exploitation. Secondly, there is a need of increasing the exploration capability of basic FPA. Thirdly, intensive exploitation must be taken into account for increasing the convergence speed of the algorithm. So in this present work, to overcome the drawbacks of existing literature, following ideas are proposed:

– A concept of improved dynamic switching probability is implemented.
– Levy flight behavior is replaced by Cauchy based operator in order to enhance global search.
– Local search is also improved by even considering the effect of current best solution.

Modification 1-*Switching Probability Concept*

The switching probability, $p$ plays an important role as it is a controlling parameter for exploration and exploitation. In general, the aim should be initially exploring the whole space then slowly moving towards the exploitation so as to search around the best particles. So in proposed approach, work has been done to balance the global and local search effectively. Hence, dynamic probability is updated by using exponentially decreasing function [109] in order to effectively balance between global and local search. The probability is updated by the following equation:

$$\mathrm{p} = p_{final} + (p_{initial} - p_{final}).\, e^{-\left(\frac{iter}{Maxgeneration/10}\right)} \tag{3.3}$$

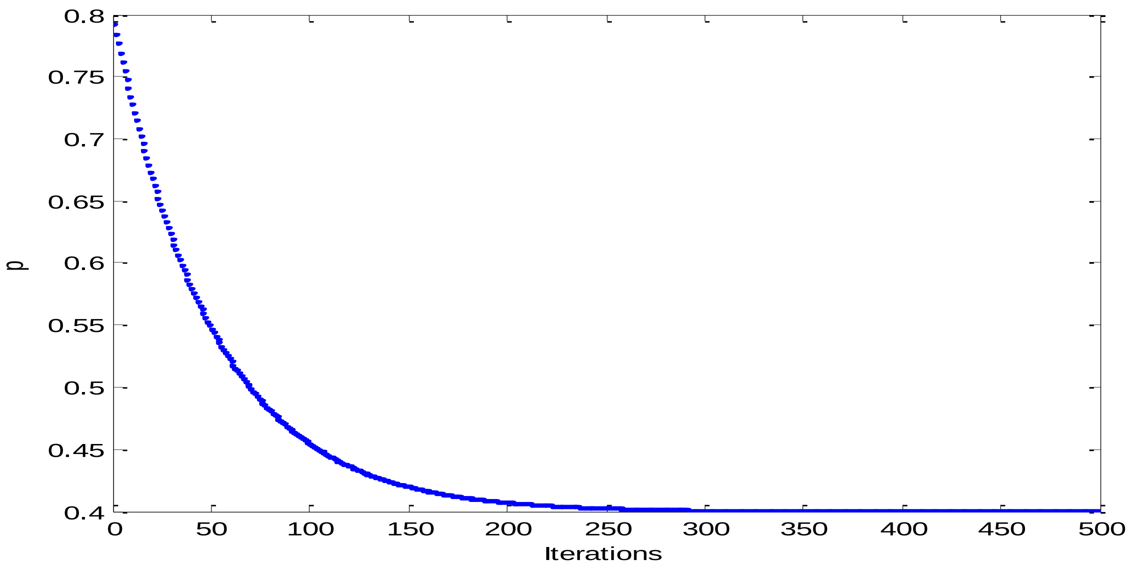



Figure 3.2 Effect of p with iterations

where, $p_{final}$=0.4 and $p_{initial}$= 0.8. It can be seen that at the initial iterations, algorithm will move towards the global search. In present work, the switching probability is exponentially decreased from range 0.8 to 0.4. Hence, it can be said that initially there will be 80 % chances for the pollinators to explore the area and 20 % chances for exploiting the area. As the algorithm proceeds, the exploring chances start decreasing and chances for exploiting start increasing. In this way, there will be proper balancing between exploitation and exploration. Figure 3.2 is showing the effect of p with respect to iterations.

Modification 2-*Global Search based upon Cauchy Operator*

Cauchy operator has large step size due to which it has very strong global searching ability. This characteristic of this operation helps algorithm for avoiding the premature convergence [11] and hence helps in avoiding local minima. This operator is being replaced with levy flight operator used for step size in global pollination part. This operation is generated with the help of Cauchy distribution which is based on random number distribution given as follows:

$$y = \frac{1}{2} + \frac{1}{\pi}\arctan\left(\frac{\delta}{g}\right) \tag{3.4}$$

The Cauchy density function can be given as:

$$f_{Cauchy(0,g)}(\delta) = \frac{1}{\pi}\frac{g}{g^2 + \delta^2} \tag{3.5}$$

where, y is basically a random distribution between 0 and 1, g is scale parameter which is taken as1. $\delta$ can be solved as follows:

$$\delta = \tan\left(\pi(y - \frac{1}{2})\right)$$

The global pollination equation can be modified as:

$$x_i^{t+1} = x_i^t + \ C(\delta)(g^* - x_i^t) \tag{3.6}$$

Since the usage of Cauchy operator allows for larger mutation, so the algorithm will get more chances for exploring the search space more effectively.

Modification 3-*Improved Local Search*

In this modification, pollinators not only update its position according to local best experience but also update its position by considering the effect of current best position. So, here if the new position fitness is better than older one then new position is updated by taken in account the effect of previous solution. This will help the pollinators to enhance local search capability by keeping the effect of diversification in neighborhood. In this way, pollinators will get more accurate solution since previous effect is also considered. The equation is updated as follows:

$$x_i^{t+1} = x_i^t + r_1\left(x_j^t - x_k^t\right) + r_1(g^* - x_i^t) \tag{3.7}$$

where, $x_j^t$ and $x_k^t$ are two solutions of jth and kth pollinators of the flower in the neighborhood ($j \neq$k). $r_1$ and $r_2$ are two uniform distributed random numbers in the range [0,1]. These random parameters will help the pollinators to randomly search the local neighborhood effectively. In this way, local pollination search is improved.

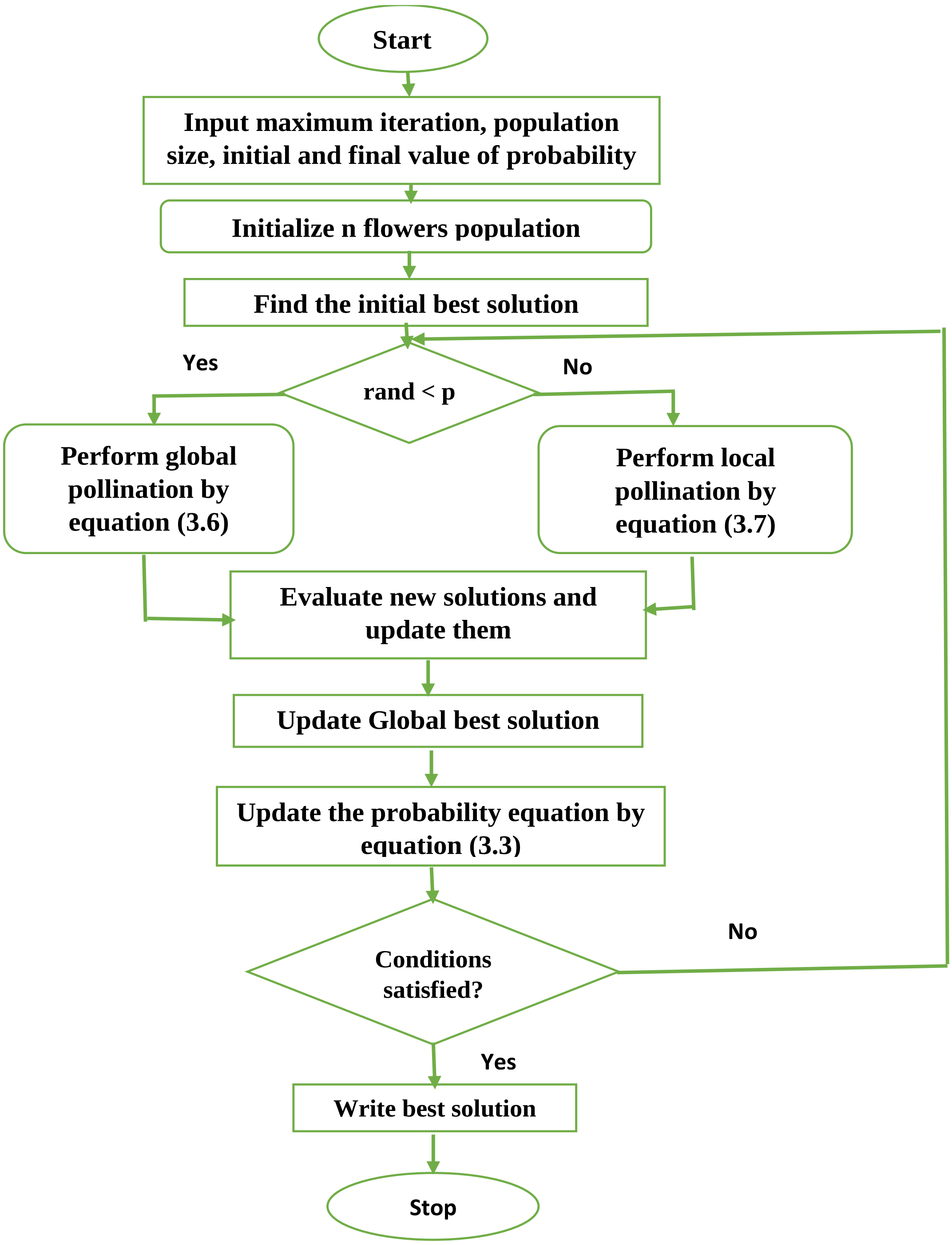



Figure 3.3 Flowchart of IFPA

*Max or min objective function f(x),* $x=(x_1, x_2, \ldots\ldots x_d)$
*Initialize the random population of n flowers*
*Find* $g^*$ *(best solution) from the population*
*Identify the switching probability* $p\epsilon[0,1]$
**while** *(iter < Maxgeneration)*
**for** *i=1:n*
**if** *rand<p*
*find vector C(step size) from Cauchy distribution*
*Use global pollination by the equation* $x_i^{t+1} = x_i^t + C(\delta)(g^* - x_i^t)$
**else**
*find the value of* $\epsilon$ *from uniform distribution*
*Choose randomly the value of j and k from the solution*
*Use local pollination by the equation by equation (3.7)*
**End if**
*Evaluate the value of new solutions*
*If new solutions are better than old one, update them.*
*Modify the probability by equation (3.3)*
**End for**
*Find the value of best Solution*
**End while**

Figure 3.4 Pseudo code of IFPA

This chapter deals with the proposed approach (IFPA) used for the implementation of antenna array. Next chapter will discussed about the benchmarking results of this proposed approach and it will be applied for the synthesis of CAA and CCAA. Detailed discussion about this will be given in next chapter.

# CHAPTER 4

# PROBLEM FORMULATION

The present work deals with the synthesis of circular and concentric antenna array using IFPA. CAA is chosen due to its advantage of providing 2D scan both horizontally as well as vertically. These array can also scan 360° horizontally having no distortion nearby end fire directions. Due to the absence of edge elements, these arrays can also reduce the effect of mutual coupling hence it is much easier to handle mutual coupling effect. CCAA is advance version of CAA, these arrays provide the advantage of its applicability in broadband as well as narrowband beam forming applications. In this work, IFPA is used for determining an optimum set of positions and amplitude that will give radiation pattern with maximum reduction in SLL having constraint of fixed major lobe beam width in case of CAA. While in the case of CCAA, an optimal set weights for CCAA are optimized for SLL reduction with the help of IFPA.

## 4.1 DESIGN EQUATIONS

### 4.1.1 Circular Antenna Array (CAA)

The circular antenna array is shown in Figure 4.1. It consists of N-elements placed non-uniformly around the circle in x-y plane having radius a. The elements of circular array can be supposed to possess isotropic properties. The array factor (AF) can be given as:

$$AF(\theta) = \sum_{n=1}^{N} I_n e^{j(kacos(\theta-\phi_n)+\alpha_n)} \tag{4.1}$$

where $$\mathrm{ka} = \left(\frac{2\pi \mathrm{a}}{\lambda}\right) = \sum_{i=1}^{N} d_i \tag{4.2}$$

where, $\alpha_n$ and $I_n$ denotes phase and amplitude excitation of antenna array of N-element and k is wave number given by $\mathrm{k} = 2\pi/\lambda$. λ is wavelength and θ is basically angle of incidence of wave. $\phi$ is defined as angular position which is given by-

$$\phi_n = (2\pi/ka)\sum_{i=1}^{n} d_i \tag{4.3}$$

If the main beam is to be directed towards the $\theta_0$ direction, excitation angle $\alpha_n$ of the nth element array can be given as

$$\alpha_n = -kacos(\theta_0 - \phi_0) \tag{4.4}$$

For the present work, $\theta_0$ is considered as 0.

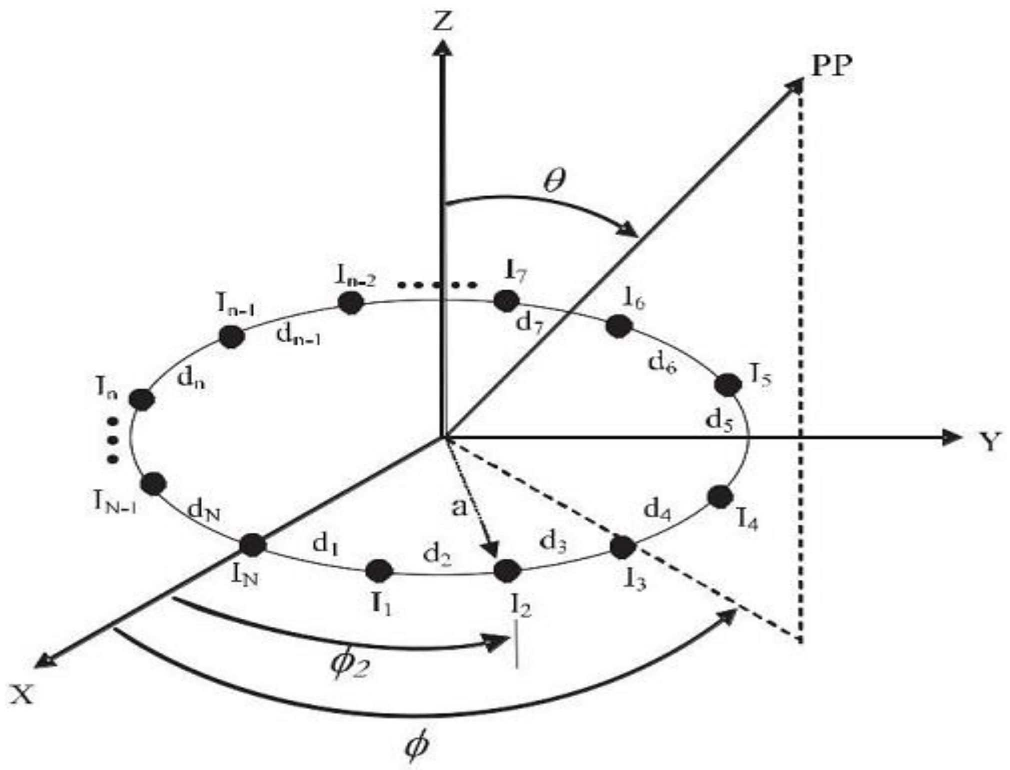



Figure 4.1 N-element non-uniform circular antenna array scanning at point PP [100]

### 4.1.2 Concentric Circular Antenna Array (CCAA)

In CCAA, antenna elements are arranged in multiple concentric rings that differ in number of elements and radii of rings. Figure 4.2 shows the configuration of P concentric rings of antenna with radius $r_p$ and having $N_p$ number of elements in ring P. The elements are assumed to be isotropic and radiation pattern of CCAA can be written in terms of array factor. So, its array factor can be written as:

$$AF(\theta,\phi,I) = 1 + \sum_{p=1}^{P}\sum_{i=1}^{N_p} I_{pi}\exp[j\left(kr_p sin\theta \cos\left(\phi - \phi_{pi}\right) + \alpha_{pi}\right)] \qquad (4.5)$$

where, k=2π/λ, λ is wavelength of the signal, $I_p$ is excitation amplitude of pth ring. $\phi$ is azimuth angle measured from positive x direction, θ is zenith angle from positive z-axis.

$\phi_{pi}$ is element to element separation measured from positive direction of x-axis. $\phi_{pi}$ can be assumed to be distributed uniformly as:

$$\phi_{pi} = 2\pi\left(\frac{i}{N_p}\right); p = 1,2, \dots, P\ ; i = 1,2, \dots \dots . N_p \qquad (4.6)$$

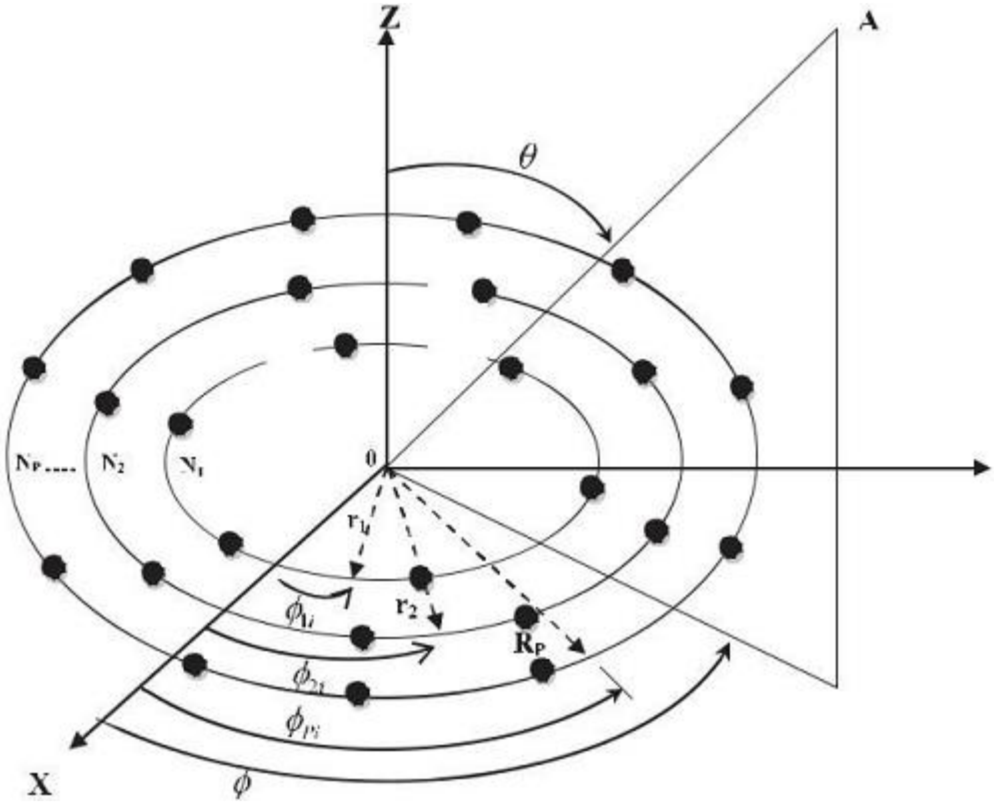



Figure 4.2 Geometrical configuration of CCAA [100]

$\alpha_{pi}$ is phase difference between the elements in array that is basically the function of radii of ring $r_p$ and angular separation between the elements $\phi_{pi}$.

$$\alpha_{pi} = -Kr_p sin\theta_0 \cos(\phi_0 - \phi_{pi})\,;\ \mathrm{p} = 1,2, \dots \dots, \mathrm{P};\ \mathrm{i} = 1,2, \dots \dots, N_p \qquad (4.7)$$

where, $\phi_0$ and $\theta_0$ can be seen as the values of $\phi$ and θ where main lobe peak is found. In this present work $\phi_0$=90 and $\theta_0$=90 is taken.

The array factor for CCAA can be rewritten as follows:

$$AF(\phi, I) = 1 + \sum_{p=1}^{P} \sum_{i=1}^{N_p} I_{pi} \exp[j(kr_p \cos(\phi - \phi_{pi}) - \cos(\phi_0 - \phi_{pi}))] \qquad (4.8)$$

# CHAPTER 5
# RESULTS AND DISCUSSION

This chapter presents the results of benchmark functions, synthesis of circular antenna array and concentric circular antenna array using IFPA. First part deals with test functions applied to test the performance of IFPA and compared with different other algorithms. The effect of population and dimension sizes are also analyzed on algorithms. Second part deals with the synthesis of circular array for reduction of SLL and FNBW. Here, three different cases, 8-element, 10-element and 12-element arrays are considered. Last section deals with the synthesis of concentric circular antenna array. Here also, three cases having number of elements (4, 6, 8), (6, 8, 10) and (8, 10, 12) in antenna array rings are discussed. The simulations are performed on Window 8, RAM 2 GB with core 2 duo processor and MATLAB version 2013a.

## 5.1 BENMARKING RESULTS

### 5.1.1 Test Function

The IFPA algorithm is tested on twenty benchmark functions for evaluating its performance. These functions are either unimodal or multimodal and have different characteristics like dimensions, search range and optimum value. Unimodal functions are those which are having only single global minima while multimodal functions are having a number of local optima's. The test functions with their characteristics are given below in Table 5.1.

| Test Functions | Objective Functions | Search range | Dimension | $f_{min}$ |
|---|---|---|---|---|
| Ackley | $f_1(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$ | [−100, 100] | 30 | 0 |
| Scaffer | $f_2(x) = \left[\frac{1}{n-1}\sqrt{s_i}.\left(\sin\left(50.0 s_i^{\frac{1}{5}}\right)+1\right)\right]^2 s_i = \sqrt{x_i^2 + x_{i+1}^2}$ | [−100, 100] | 30 | 0 |
| Levy | $f_3(x) = sin^2(\pi w_1) + \sum_{i=1}^{d-1}(w_i-1)^2[1+10sin^2(\pi w_i+1)] + (w_d-1)^2[1+10sin^2(2\pi w_d)]$ | [-10 10] | 30 | 0 |
| Griewank | $f_4 = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos(\frac{x_i}{\sqrt{i}}) + 1$ | [−600, 600] | 30 | 0 |
| Rastrigin | $f_5(x) = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$ | [−5.12, 5.12] | 30 | 0 |

| | | | | |
|---|---|---|---|---|
| Rotated Hyper-Ellipsoid | $f_6(x) = \sum_{i=1}^{d} \sum_{j=1}^{i} x_j^2$ | [-65.536, 65.536] | 30 | 0 |
| Weierstrass | $f_7(x) = \sum_{i=1}^{D} \sum_{k=0}^{kmax} [a^k \cos\left(2\pi b^k (x_i + 0.5)\right)] - D \sum_{k=0}^{kmax} [a^k \cos(2\pi b^k . 0.5)]$; *where a=0.5, b=3, kmax=20* | [−0.5, 0.5] | 50 | 0 |
| Sphere | $f_8(x) = \sum_{i=1}^{D} x_i^2$ | [-10 10] | 30 | 0 |
| Powell | $f_9(x) = \sum_{i=1}^{d} [(x_{4i-3} - 10x_{4i-2})^2 + 5(x_{4i-1} - x_{4i})^2 + (x_{4i-2} - 2x_{4i-1})^4 + 10(x_{4i-3} - x_{4i})^2]$ | [-4 5] | 30 | 0 |
| Sum Of Different Powers | $f_{10}(x) = \sum_{i=1}^{D} x_i^{i+1}$ | [-1 1] | 30 | 0 |
| Sum Squares | $f_{11}(x) = \sum_{i=1}^{D} i x_i^2$ | [-10 10] | 30 | 0 |
| Schaffer N.2 | $f_{12} = 0.5 + \frac{sin^2(x_1^2 - x_2^2)^2 - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}$ | [−100,100] | 2 | 0 |
| Schaffer N.4 | $f_{13} = 0.5 + \frac{cos^2(sin(x_1^2 - x_2^2)^2 - 0.5}{1 + 0.001(x_1^2 + x_2^2)^2}$ | [−100,100] | 2 | 0 |
| Beale | $f_{14}(x) = [1.5 - x_1(1 - x_2)]^2 + [2.25 - x_1(1 - x_2^2)]^2 + [2.625 - x_1(1 - x_2^3)]^2$ | [−4.5, 4.5] | 2 | 0 |
| De Jong F5 | $f_{15}(x) = [0.002 + \sum_{j=1}^{25} [j + \left(x_1 - a_{1j}\right)^6 + \left(x_2 - a_{2j}\right)^6]^{-1}]^{-1}$ $(a_{ij})= \begin{pmatrix} -32 & -16 & 0 & 16 & 32 & -32 & \ldots & 0 & 16 & 32 \\ -32 & -32 & -32 & -32 & -32 & -16 & \ldots & 32 & 32 & 32 \end{pmatrix}$ | [−65.536, 65.536] | 2 | 0 |
| Sixhumpcamel | $f_{16}(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right) x_1^2 + x_1 x_2 + (-4 + 4x_2^2)x_2^2$ | [−5, 5] | 2 | −1.0316 |
| Branin | $f_{17}(x) = (x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right) cosx_1 + 10$ | [−5, 10] | 2 | 0.397887 |
| Easom | $f_{18}(x) = -cosx_1 cosx_2 e^{(-(x_1-\pi)^2 - (x_2-\pi)^2)}$ | [−10, 10] | 2 | -1 |
| Colville | $f_{19}(x) = 100(x_1 - x_2^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x_3)^2 + 10.1((x_2 - 1)^2 + (x_4 - 1)^2) + 19.8(x_2 - 1)(x_4 - 1)$ | [-10 10] | 4 | 0 |
| Drop-Wave | $f_{20}(x) = -\frac{1 + \cos(12\sqrt{x_1^2 - x_2^2})}{0.5 + (x_1^2 - x_2^2) + 2}$ | [-5.12,5.12] | 2 | 0 |

Table 5.1 Test Function Table

### 5.1.2 Parameter Settings

For testing the results using benchmark functions, different algorithms namely FPA [8], BA [24], DE [9], BFP (bat flower pollination algorithm) [110] are applied for twenty benchmarking functions. These algorithms run for 50 runs and 500 maximum number of iterations. The parameter settings for different algorithms are given below:

| Algorithms | Parameter | Values |
|---|---|---|
| DE | F(Control Parameter) | 1.5 |
| | CR(Crossover Rate) | 0.5 |
| | Maximum Iterations | 500 |
| BA | Loudness(A) | 0.5 |
| | Pulse Rate(r) | 0.5 |
| | Maximum Iterations | 500 |
| BFP | Loudness(A) | 0.5 |
| | Pulse Rate(r) | 0.5 |
| | Switching probability, p | 0.8 |
| | Maximum Iterations | 500 |
| FPA | Switching probability, p | 0.8 |
| | Maximum Iterations | 500 |
| IFPA | Initial p | 0.8 |
| | Final p | 0.4 |
| | Maximum Iterations | 500 |

Table 5.2 Parameter settings

### 5.1.3 Influence of Population Size

In present work, four different population sizes 20, 40, 60 and 80 are taken for testing the performance of IFPA. The obtained results are compared with different state-of-the-art algorithms. The population sizes are varied while keeping dimension fixed. Table 5.3 shows the comparison results for population size 20, Table 5.4 for population 40, Table 5.5 for population 60 and Table 5.6 for population size 80. The results are discussed as below:

Case I Population size 20

The results for population size 20 are given in Table 5.3. For function $f_1$, IFPA provides better results while other algorithm results are also comparable. For function $f_2$, IFPA is able to achieve global solution and no other algorithm is even closer to it. For functions $f_3$ and $f_4$, IFPA is getting better results while FPA and BA results are also comparable. BA is found to be best for function $f_5$. For function $f_6$, FPA and IFPA are competitive while rest algorithms are comparable. For functions $f_7$ to $f_9$ , best algorithm in these cases is IFPA while rest algorithms are also quite comparable. Best results are achieved by IFPA for function $f_{10}$ although FPA and BA results are also comparable to IFPA. For function $f_{11}$, IFPA is found to be better than other algorithms. For function $f_{12}$, optimal solution is achieved by IFPA while other algorithms get stuck in some sub-optimal solution. All algorithms are consistent in giving better solution for function $f_{13}$ while results of BFP and IFPA are comparable. For function $f_{14}$, IFPA is able to reach global optima. For $f_{15}$ FPA can be said to be better in terms of standard deviation. For $f_{16}$ and $f_{17}$, best, worst, mean are same for all algorithms, hence IFPA is better in terms of standard deviation. Optimal solution is achieved by IFPA for function $f_{18}$. For $f_{19}$, best results are achieved by IFPA. FPA is best among all algorithms in case of function $f_{20}$. It can be concluded that IFPA is better for sixteen functions, FPA for two functions, BA for two functions, DE for one function and BFP is not even good for any function so it is worst among all algorithms.

| Function | Algorithm | Best | Worst | Mean | Standard deviation |
|---|---|---|---|---|---|
| $\boldsymbol{f_1}$ | FPA | 20.9110 | 21.1748 | 21.0909 | 0.0543 |
| | BA | 19.9945 | 20.0002 | 19.9997 | 9.07E-04 |
| | DE | 20.2512 | 20.9120 | 20.4696 | 0.1288 |
| | BFP | 21.2831 | 21.5169 | 21.4117 | 0.0621 |
| | IFPA | **15.2047** | 20.0007 | **19.9041** | 0.6782 |
| $\boldsymbol{f_2}$ | FPA | 2.21E-11 | 3.67E-06 | 2.14E-07 | 5.59E-07 |
| | BA | 0.0031 | 0.4420 | 0.1762 | 0.1216 |
| | DE | 0.0018 | 0.3804 | 0.0887 | 0.0766 |
| | BFP | 0.0993 | 0.4935 | 0.4164 | 0.0957 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $\boldsymbol{f_3}$ | FPA | 7.4369 | 27.5825 | 16.6038 | 5.5072 |
| | BA | 21.5575 | 99.7949 | 50.5521 | 21.2471 |
| | DE | 43.2259 | 2.91E+02 | 1.49E+02 | 64.8792 |
| | BFP | 1.43E+02 | 3.14E+02 | 2.38E+02 | 39.1929 |
| | IFPA | **2.1363** | **24.2861** | **9.3177** | **3.7443** |
| $\boldsymbol{f_4}$ | FPA | 9.2655 | 61.6138 | 32.0290 | 10.9757 |
| | BA | 85.9132 | 4.58E+02 | 2.28E+02 | 79.6625 |
| | DE | 2.18E+02 | 7.70E+02 | 5.94E+02 | 1.20E+02 |
| | BFP | 3.62E+02 | 8.65E+02 | 6.38E+02 | 1.21E+02 |
| | IFPA | **3.8447** | **37.8817** | **15.8363** | **8.5317** |
| $\boldsymbol{f_5}$ | FPA | 1.40E+02 | 2.09E+02 | 1.83E+02 | 13.2170 |
| | BA | **31.8448** | 1.80E+02 | 99.2256 | 33.2521 |
| | DE | 1.80E+02 | 5.00E+02 | 4.24E+02 | 63.9031 |
| | BFP | 3.48E+02 | 5.11E+02 | 4.44E+02 | 40.5718 |
| | IFPA | 49.2976 | 1.12E+02 | 79.9486 | 15.7964 |
| $\boldsymbol{f_6}$ | FPA | 7.74E+03 | 3.90E+04 | 1.87E+04 | 6.81E+03 |
| | BA | 6.54E+04 | 3.45E+05 | 1.61E+05 | 6.20E+04 |
| | DE | 1.02E+05 | 5.36E+05 | 3.85E+05 | 1.17E+05 |
| | BFP | 2.27E+05 | 5.65E+05 | 4.15E+05 | 9.67E+04 |
| | IFPA | **3.07E+03** | **2.51E+04** | **8.89E+03** | **4.72E+03** |
| $\boldsymbol{f_7}$ | FPA | 50.1115 | 70.3378 | 63.4783 | 5.7816 |
| | BA | 50.6694 | 77.0234 | 61.5011 | 6.0299 |
| | DE | 41.7407 | 73.7880 | 53.5144 | 6.6480 |
| | BFP | 66.0523 | 90.7637 | 83.4279 | 5.3372 |
| | IFPA | **31.1453** | **47.0764** | **39.8466** | **3.7866** |
| $\boldsymbol{f_8}$ | FPA | 5.1767 | 21.1847 | 8.6794 | 2.9056 |
| | BA | 15.9738 | 86.0038 | 37.5689 | 16.0374 |
| | DE | 58.0683 | 2.29E+02 | 1.75E+02 | 35.4580 |
| | BFP | 90.8473 | 2.49E+02 | 1.87E+02 | 36.6693 |
| | IFPA | **0.6959** | **14.5569** | **4.4123** | **2.5008** |
| $\boldsymbol{f_9}$ | FPA | 47.6987 | 6.42E+02 | 1.50E+02 | 95.4533 |
| | BA | 1.48E+02 | 4.98E+03 | 1.21E+03 | 8.00E+02 |
| | DE | 8.61E+02 | 2.14E+04 | 6.05E+03 | 4.26E+03 |
| | BFP | 4.32E+03 | 4.16E+04 | 1.82E+04 | 7.76E+03 |
| | IFPA | **8.6605** | **2.61E+02** | **84.1477** | **58.4166** |

| | | | | | |
|---|---|---|---|---|---|
| $f_{10}$ | FPA | 7.31E-08 | 9.47E-05 | 1.18E-05 | 1.66E-05 |
| | BA | 1.39E-07 | 5.98E-06 | 1.00E-06 | 9.83E-07 |
| | DE | 0.2039 | 1.8038 | 0.9458 | 0.3235 |
| | BFP | 0.1119 | 1.5979 | 0.8107 | 0.3989 |
| | IFPA | **9.90E-10** | 8.91E-05 | 9.65E-06 | 1.88E-05 |
| $f_{11}$ | FPA | 1.69E+02 | 9.93E+02 | 4.29E+02 | 1.51E+02 |
| | BA | 1.50E+03 | 4.88E+03 | 2.98E+03 | 9.22E+02 |
| | DE | 2.17E+03 | 1.30E+04 | 9.03E+03 | 2.74E+03 |
| | BFP | 5.49E+03 | 1.48E+04 | 9.78E+03 | 1.98E+03 |
| | IFPA | **50.7534** | **4.97E+02** | **2.31E+02** | **1.13E+02** |
| $f_{12}$ | FPA | 8.43E-12 | 2.76E-06 | 2.97E-07 | 5.09E-07 |
| | BA | 0.0054 | 0.4880 | 0.1791 | 0.1497 |
| | DE | 0.00E-00 | 0.1689 | 0.0106 | 0.0270 |
| | BFP | 0.0626 | 0.4892 | 0.3705 | 0.1025 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{13}$ | FPA | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | BA | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | DE | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | BFP | 0.5000 | 0.5000 | 0.5000 | 6.59E-06 |
| | IFPA | 0.5000 | 0.5000 | 0.5000 | 2.84E-08 |
| $f_{14}$ | FPA | 1.91E-12 | 4.63E-06 | 3.80E-07 | 9.23E-07 |
| | BA | 8.58E-12 | 0.8995 | 0.2426 | 0.3253 |
| | DE | 0.00E-00 | 0.4656 | 0.0186 | 0.0908 |
| | BFP | 0.0837 | 13.9340 | 4.1291 | 4.0622 |
| | IFPA | **0.00E-00** | 0.7621 | 0.0152 | 0.1078 |
| $f_{15}$ | FPA | 0.9980 | 1.0343 | 0.9995 | **0.0060** |
| | BA | 0.9980 | 52.4729 | 12.9569 | 9.3272 |
| | DE | 0.9980 | 12.6705 | 2.4657 | 2.7042 |
| | BFP | 11.0882 | 4.96E+02 | 3.06E+02 | 1.88E+02 |
| | IFPA | 0.9980 | 4.9504 | 1.2951 | 0.8541 |
| $f_{16}$ | FPA | -1.0316 | -1.0316 | -1.0316 | 1.61E-07 |
| | BA | -1.0316 | -0.2154 | -0.9663 | 0.2236 |
| | DE | -1.0316 | -1.0200 | -1.0313 | 0.0016 |
| | BFP | -1.0230 | 40.2811 | 5.2714 | 8.8662 |
| | IFPA | -1.0316 | -1.0316 | -1.0316 | **6.72E-16** |
| $f_{17}$ | FPA | 0.3979 | 0.3979 | 0.3979 | 1.10E-05 |
| | BA | 0.3979 | 2.6877 | 0.4436 | 0.3238 |
| | DE | 0.3979 | 0.4022 | 0.3980 | 6.44E-04 |
| | BFP | 0.4090 | 16.3983 | 4.9712 | 3.7151 |
| | IFPA | 0.3979 | 0.3979 | 0.3979 | **3.36E-16** |
| $f_{18}$ | FPA | -1.0000 | -1.0000 | -1.0000 | 1.10E-09 |
| | BA | -1.0000 | -3.22E-06 | -0.8821 | 0.3111 |
| | DE | -1.0000 | -0.8730 | -0.9958 | 0.0203 |
| | BFP | -0.5498 | -4.97E-60 | -0.0461 | 0.1201 |
| | IFPA | -1.0000 | -1.0000 | -1.0000 | **0.00E-00** |
| $f_{19}$ | FPA | 0.0201 | 6.7230 | 1.2510 | 1.2592 |
| | BA | 1.0952 | 4.94E+03 | 4.99E+02 | 1.08E+03 |
| | DE | 2.24E-05 | 1.67E+03 | 50.3138 | 2.42E+02 |
| | BFP | 3.15E+02 | 8.01E+04 | 1.14E+04 | 1.48E+04 |
| | IFPA | **0.00E-00** | **6.1503** | **0.8481** | 1.7807 |
| $f_{20}$ | FPA | -1.0000 | -0.9958 | -0.9995 | **7.57E-04** |
| | BA | -1.0000 | -0.2297 | -0.8012 | 0.1909 |
| | DE | -1.0000 | -0.9361 | -0.9616 | 0.0295 |
| | BFP | -0.9362 | -0.1858 | -0.5707 | 0.2125 |
| | IFPA | -1.0000 | -0.9362 | -0.9987 | 0.0090 |

Table 5.3 Results of IFPA compared with different other algorithms for population size 20

Case II Population size 40

The results for population size 40 are analyzed in Table 5.4. For function $f_1$, best results are achieved by IFPA while results of other algorithms are also comparable. In case of function $f_2$, IFPA has achieved exact global optimal solution although DE is providing competitive result in terms of best value. For functions $f_3$ to $f_7$, all algorithms give competitive results while IFPA is best among all. IFPA is also best among all algorithms for function $f_8$ and $f_9$. For function $f_{10}$, IFPA is able to achieve superior performance and no algorithm is even closer to it. IFPA is achieving better results for $f_{11}$ even though results of other algorithms are also quite comparable to IFPA. In case of functions $f_{12}$ and $f_{14}$, IFPA attains global best solution while other algorithms get stuck in some sub-optimal solution. For $f_{13}$, FPA, BA, DE and BFP provide exact zero standard deviation so there is no clue given for measuring the performance of this function. For functions $f_{15}$, $f_{16}$ and $f_{17}$, IFPA is almost closer to best solution while results of other algorithms are also quite competitive. For $f_{18}$, $f_{19}$ and $f_{20}$, IFPA achieves global optimum solution and no algorithm is even closer to it except DE which is closer to IFPA. At last, it can be concluded that IFPA is better for eighteen functions, DE for five functions, FPA and BA for two functions each, BFP for one function. So IFPA is best among all for population size 40.

| Function | Algorithm | Best | Worst | Mean | Standard deviation |
|---|---|---|---|---|---|
| $\boldsymbol{f_1}$ | FPA | 20.8682 | 21.1524 | 21.0568 | 0.06338 |
| | BA | 19.9895 | 20.0000 | 19.9996 | 0.0014 |
| | DE | 20.2657 | 20.8332 | 20.4677 | 0.1377 |
| | BFP | 21.1164 | 21.4276 | 21.3111 | 0.0754 |
| | IFPA | **5.1864** | 20.0018 | **19.7038** | 2.0950 |
| $\boldsymbol{f_2}$ | FPA | 9.64E-10 | 1.91E-06 | 1.71E-07 | 3.17E-07 |
| | BA | 5.90E-14 | 0.3500 | 0.0741 | 0.0791 |
| | DE | 0.00E-00 | 0.1293 | 0.0094 | 0.0197 |
| | BFP | 0.0397 | 0.4847 | 0.3391 | 0.1139 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $\boldsymbol{f_3}$ | FPA | 12.1573 | 40.5982 | 26.9623 | 7.0665 |
| | BA | 6.2540 | 39.1277 | 18.6075 | 7.1550 |
| | DE | 76.0753 | 2.57E+02 | 1.73E+02 | 46.1058 |
| | BFP | 1.18E+02 | 2.88E+02 | 2.11E+02 | 36.0677 |
| | IFPA | **0.7583** | **12.4744** | **4.7648** | **2.5800** |
| $\boldsymbol{f_4}$ | FPA | 11.2945 | 45.8340 | 28.3257 | 8.0484 |
| | BA | 87.5959 | 3.40E+02 | 1.64E+02 | 54.6780 |
| | DE | 1.96E+02 | 7.27E+02 | 5.83E+02 | 88.9315 |
| | BFP | 3.70E+02 | 7.21E+02 | 5.86E+02 | 95.5944 |
| | IFPA | **0.9793** | **5.5156** | **1.6812** | **0.9161** |
| $\boldsymbol{f_5}$ | FPA | 1.71E+02 | 2.29E+02 | 1.98E+02 | 13.7772 |
| | BA | 32.8369 | 1.70E+02 | 85.7090 | 30.4648 |
| | DE | 2.49E+02 | 4.77E+02 | 4.13E+02 | 50.0437 |
| | BFP | 3.24E+02 | 4.85E+02 | 4.12E+02 | 34.7657 |
| | IFPA | **20.8032** | **77.0127** | **45.7600** | **11.6564** |
| $\boldsymbol{f_6}$ | FPA | 1.02E+04 | 2.76E+04 | 1.77E+04 | 4.50E+03 |

| | | | | | |
|---|---|---|---|---|---|
| | BA | 2.37E+04 | 1.99E+05 | 1.00E+05 | 3.61E+04 |
| | DE | 1.60E+05 | 5.29E+05 | 3.89E+05 | 9.32E+04 |
| | BFP | 2.56E+05 | 5.65E+05 | 4.16E+05 | 7.15E+04 |
| | IFPA | **18.8188** | **1.80E+03** | **4.64E+02** | **4.28E+02** |
| $f_7$ | FPA | 61.8880 | 70.3235 | 65.8094 | **1.7508** |
| | BA | 42.9900 | 70.1629 | 54.1291 | 5.2363 |
| | DE | 46.5217 | 71.1113 | 58.1938 | 5.9909 |
| | BFP | 69.3334 | 88.5839 | 81.1974 | 5.3107 |
| | IFPA | **27.0552** | **37.2629** | **32.4573** | 2.7605 |
| $f_8$ | FPA | 4.6038 | 17.1392 | 8.2441 | 2.4460 |
| | BA | **2.86E-05** | 19.1597 | 6.1087 | 4.9500 |
| | DE | 93.8742 | 2.11E+02 | 1.70E+02 | 23.3228 |
| | BFP | 1.06E+02 | 2.10E+02 | 1.71E+02 | 23.0650 |
| | IFPA | 0.0114 | 0.6177 | 0.1348 | 0.1235 |
| $f_9$ | FPA | 71.4903 | 3.86E+02 | 1.97E+02 | 72.8067 |
| | BA | 4.6520 | 7.64E+02 | 1.39E+02 | 1.48E+02 |
| | DE | 5.96E+02 | 1.72E+04 | 6.35E+03 | 3.65E+03 |
| | BFP | 6.67E+03 | 3.29E+04 | 1.53E+04 | 5.45E+03 |
| | IFPA | **0.1212** | **9.5906** | **2.3579** | **2.0407** |
| $f_{10}$ | FPA | 4.82E-06 | 7.94E-04 | 1.37E-04 | 1.75E-04 |
| | BA | 1.91E-08 | 1.65E-07 | 8.25E-08 | 3.79E-08 |
| | DE | 0.2113 | 1.2080 | 0.7390 | 0.2339 |
| | BFP | 0.1064 | 1.1859 | 0.5558 | 0.2220 |
| | IFPA | **1.84E-15** | **7.46E-09** | **7.62E-10** | **1.75E-09** |
| $f_{11}$ | FPA | 1.81E+02 | 6.95E+02 | 4.12E+02 | 1.06E+02 |
| | BA | 69.2847 | 2.42E+03 | 1.03E+03 | 4.76E+02 |
| | DE | 4.21E+03 | 1.14E+04 | 8.91E+03 | 1.79E+03 |
| | BFP | 6.16E+03 | 1.22E+04 | 9.60E+03 | 1.56E+03 |
| | IFPA | **0.2960** | **35.5454** | **7.1378** | **8.0702** |
| $f_{12}$ | FPA | 8.37E-10 | 6.88E-07 | 1.13E-07 | 1.52E-07 |
| | BA | 6.61E-14 | 0.2747 | 0.0651 | 0.0767 |
| | DE | 0.00E-00 | 0.0158 | 5.33E-04 | 0.0023 |
| | BFP | 0.0020 | 0.4813 | 0.3339 | 0.1357 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{13}$ | FPA | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | BA | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | DE | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | BFP | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | IFPA | 0.5000 | 0.5000 | 0.5000 | 2.69E-09 |
| $f_{14}$ | FPA | 1.43E-12 | 9.63E-06 | 5.04E-07 | 1.63E-06 |
| | BA | 9.28E-11 | 0.8023 | 0.1295 | 0.2623 |
| | DE | 0.00E-00 | 1.47E-05 | 4.57E-07 | 2.21E-06 |
| | BFP | 0.0041 | 6.6516 | 1.4852 | 1.6144 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{15}$ | FPA | 0.9980 | 0.9985 | 0.9980 | 8.52E-05 |
| | BA | 0.9980 | 23.8094 | 10.3573 | 7.0133 |
| | DE | 0.9980 | 5.9288 | 1.4917 | 1.3699 |
| | BFP | 4.0136 | 4.89E+02 | 1.59E+02 | 1.64E+02 |
| | IFPA | 0.9980 | 0.9980 | 0.9980 | **1.87E-16** |
| $f_{16}$ | FPA | -1.0316 | -1.0316 | -1.0316 | 7.50E-08 |
| | BA | -1.0316 | -0.2154 | -0.9989 | 0.1615 |
| | DE | -1.0316 | -1.0316 | -1.0316 | 6.72E-16 |
| | BFP | -1.0268 | 7.0678 | 0.0994 | 1.2140 |
| | IFPA | -1.0316 | -1.0315 | -1.0316 | **6.72E-16** |
| $f_{17}$ | FPA | 0.3979 | 0.3979 | 0.3979 | 3.27E-06 |
| | BA | 0.3979 | 0.3979 | 0.3979 | 4.06E-10 |
| | DE | 0.3979 | 0.3979 | 0.3979 | 4.79E-05 |

| | | | | | |
|---|---|---|---|---|---|
| | BFP | 0.4151 | 10.3487 | 2.6077 | 2.1852 |
| | IFPA | 0.3979 | 0.3979 | 0.3979 | **3.36E-16** |
| $f_{18}$ | FPA | -1.0000 | -1.0000 | -1.0000 | 5.20E-10 |
| | BA | -1.0000 | -1.0000 | -1.0000 | 2.97E-10 |
| | DE | -1.0000 | -1.0000 | -1.0000 | 0.00E-00 |
| | BFP | -0.6989 | -1.72E-16 | -0.0759 | 0.1468 |
| | IFPA | -1.0000 | -1.0000 | -1.0000 | **0.00E-00** |
| $f_{19}$ | FPA | 0.1266 | 3.6857 | 1.1492 | 0.8794 |
| | BA | 1.09E-04 | 6.13E+02 | 28.8585 | 92.3963 |
| | DE | 2.32E-06 | 6.6686 | 0.4347 | 1.3215 |
| | BFP | 85.8750 | 1.15E+04 | 3.35E+03 | 3.20E+03 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{20}$ | FPA | -1.0000 | -0.9978 | -0.9997 | 3.84E-04 |
| | BA | -1.0000 | -0.6194 | -0.9036 | 0.0763 |
| | DE | -1.0000 | -0.9361 | -0.9730 | 0.0316 |
| | BFP | -0.9362 | -0.1858 | -0.5929 | 0.1900 |
| | IFPA | -1.0000 | -1.0000 | -1.0000 | **0.00E-00** |

Table 5.4 Results of IFPA compared with different other algorithms for population size 40

Case III Population size 60

The results for population size 60 are shown in Table 5.5. For function $f_1$, best results are obtained by IFPA while BA is competitive in terms of standard deviation. For function $f_2$, IFPA is able to reach global optimum solution although DE is comparable in terms of best value. IFPA is giving better solution for functions $f_3$ to $f_9$ and no algorithm is even comparable to it. For $f_{10}$, IFPA is giving much superior performance and closer to optimal solution even though FPA and BA are also providing competitive results . IFPA is better results when being compared to other algorithms in terms of best, worst, mean and standard deviation for function $f_{11}$. For $f_{12}$ and $f_{13}$, IFPA is able to reach exact optimum solution. For function $f_{13}$, best , worst and mean are same for all functions, only standard deviation is the criterion for measuring its performance which is zero for all algorithms except IFPA and BFP, hence no algorithm can be said to be best. In case of functions $f_{15}$ to $f_{17}$, IFPA is closer to optimal global solution. IFPA has reached to exact global best solution in case for functions $f_{18}$ to $f_{20}$. Hence, IFPA is better for nineteen functions, DE for six functions, BA for two functions and FPA and BFP for one function each. In this case, FPA and BFP are worst and IFPA is best among all algorithms in case of population size 60.

| Function | Algorithm | Best | Worst | Mean | Standard deviation |
|---|---|---|---|---|---|
| $f_1$ | FPA | 20.9681 | 21.1273 | 21.0514 | 0.0432 |
| | BA | 19.9988 | 20.0000 | 19.9998 | 2.18E-04 |
| | DE | 20.3261 | 20.9175 | 20.5837 | 0.1389 |
| | BFP | 20.9399 | 21.3228 | 21.2373 | 0.0668 |
| | IFPA | **4.8198** | 20.0000 | **19.6964** | 2.1468 |
| $f_2$ | FPA | 7.75E-10 | 2.25E-06 | 1.59E-07 | 3.48E-07 |
| | BA | 1.29E-13 | 0.1849 | 0.0479 | 0.0497 |
| | DE | **0.00E-00** | 0.0606 | 0.0064 | 0.0115 |
| | BFP | 0.0116 | 0.4876 | 0.3178 | 0.1344 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_3$ | FPA | 21.1950 | 58.517 | 36.5075 | 8.2231 |
| | BA | 3.7348 | 38.0682 | 11.0739 | 6.1225 |
| | DE | 76.2844 | 2.49E+02 | 1.71E+02 | 42.3565 |
| | BFP | 1.05E+02 | 2.56E+02 | 1.87E+02 | 32.2163 |
| | IFPA | **0.2694** | **10.0915** | **3.2628** | **2.3190** |
| $f_4$ | FPA | 21.1554 | 79.3592 | 21.1554 | 12.1881 |
| | BA | 64.2175 | 2.47E+02 | 64.2175 | 41.1938 |
| | DE | 3.48E+02 | 7.13E+02 | 3.48E+02 | 71.5080 |
| | BFP | 3.32E+02 | 7.62E+02 | 3.32E+02 | 87.2505 |
| | IFPA | **0.0907** | **0.9931** | **0.4663** | **0.2457** |
| $f_5$ | FPA | 1.65E+02 | 2.40E+02 | 2.03E+02 | 16.3656 |
| | BA | 32.8373 | 1.80E+02 | 74.4456 | 30.9377 |
| | DE | 2.81E+02 | 4.77E+02 | 4.21E+02 | 39.1614 |
| | BFP | 3.18E+02 | 4.40E+02 | 3.92E+02 | 30.1428 |
| | IFPA | **19.0294** | **69.6669** | **40.7002** | **12.2359** |
| $f_6$ | FPA | 1.52E+04 | 4.26E+04 | 2.71E+04 | 7.00E+03 |
| | BA | 2.46E+04 | 1.60E+05 | 7.04E+04 | 2.72E+04 |
| | DE | 1.87E+05 | 4.81E+05 | 3.65E+05 | 7.52E+04 |
| | BFP | 2.28E+05 | 5.36E+05 | 3.86E+05 | 7.142E+04 |
| | IFPA | **0.1258** | **3.31E+02** | **13.7418** | **48.6146** |
| $f_7$ | FPA | 60.9148 | 68.7374 | 30.7530 | 1.6779 |
| | BA | 35.9460 | 62.5141 | 50.7526 | 5.9103 |
| | DE | 44.7698 | 73.3859 | 58.8839 | 5.9894 |
| | BFP | 64.9063 | 87.4538 | 79.7241 | 4.8982 |
| | IFPA | **22.6270** | **36.2572** | **28.8781** | 3.0547 |
| $f_8$ | FPA | 6.1037 | 17.7312 | 12.2761 | 2.6530 |
| | BA | 1.30E-05 | 3.4364 | 0.24091 | 0.7070 |
| | DE | 1.03E+02 | 2.00E+02 | 1.64E+02 | 18.8026 |
| | BFP | 70.4948 | 2.02E+02 | 1.60E+02 | 29.0757 |
| | IFPA | 5.26E-05 | **0.0450** | **0.0036** | **0.0080** |
| $f_9$ | FPA | 1.00E+02 | 4.77E+02 | 3.09E+02 | 99.9300 |
| | BA | 0.0643 | 53.8723 | 7.3841 | 11.9448 |
| | DE | 2.00E+03 | 1.35E+04 | 7.42E+03 | 2.75E+03 |
| | BFP | 4.00E+03 | 2.16E+04 | 1.20E+04 | 3.89E+03 |
| | IFPA | **0.0138** | **3.1886** | **0.2447** | **0.4831** |
| $f_{10}$ | FPA | 5.63E-05 | 0.0024 | 5.83E-04 | 5.08E-04 |
| | BA | 1.05E-08 | 8.20E-08 | 4.17E-08 | 1.87E-08 |
| | DE | 0.2200 | 1.0998 | 0.6351 | 0.2014 |
| | BFP | 0.0762 | 0.8353 | 0.3465 | 0.1866 |
| | IFPA | **2.93E-19** | **7.59E-12** | **1.85E-13** | **1.07E-12** |
| $f_{11}$ | FPA | 3.40E+02 | 1.00E+03 | 6.34E+02 | 1.57E+02 |
| | BA | 1.5726 | 1.19E+03 | 3.83E+02 | 2.39E+02 |
| | DE | 5.49E+03 | 1.15E+04 | 8.85E+03 | 1.31E+03 |
| | BFP | 5.06E+03 | 1.18E+04 | 9.38E+03 | 1.36E+03 |
| | IFPA | **8.40E-04** | **4.5187** | **0.2055** | **0.6387** |

| | | | | | |
|---|---|---|---|---|---|
| $f_{12}$ | FPA | 6.11E-11 | 2.48E-06 | 2.23E-07 | 3.89E-07 |
| | BA | 3.30E-13 | 0.2499 | 0.0450 | 0.0544 |
| | DE | 0.00E-00 | 0.0012 | 3.83E-05 | 1.85E-04 |
| | BFP | 0.0359 | 0.4873 | 0.2938 | 0.1303 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{13}$ | FPA | 0.5000 | 0.5000 | 0.5000 | 0.00-00 |
| | BA | 0.5000 | 0.5000 | 0.5000 | 0.00-00 |
| | DE | 0.5000 | 0.5000 | 0.5000 | 0.00-00 |
| | BFP | 0.5000 | 0.5000 | 0.5000 | 0.00-00 |
| | IFPA | 0.5000 | 0.5000 | 0.5000 | 8.53E-11 |
| $f_{14}$ | FPA | 6.29E-11 | 1.64E-06 | 1.70E-07 | 2.67E-07 |
| | BA | 2.85E-12 | 0.7028 | 0.1662 | 0.2722 |
| | DE | 0.00E-00 | 2.84E-13 | 5.69E-15 | 4.02E-14 |
| | BFP | 1.01E-06 | 4.7335 | 0.6574 | 0.9683 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{15}$ | FPA | 0.9980 | 0.9980 | 0.9980 | 8.90E-06 |
| | BA | 0.9980 | 23.8094 | 10.6052 | 7.0639 |
| | DE | 0.9980 | 5.9288 | 1.2548 | 0.9927 |
| | BFP | 1.9971 | 3.97E+02 | 65.0562 | 86.8961 |
| | IFPA | 0.9980 | 0.9980 | 0.9980 | **1.12E-16** |
| $f_{16}$ | FPA | -1.0316 | -1.0316 | -1.0316 | 7.46E-08 |
| | BA | -1.0316 | -0.2154 | -0.9663 | 0.2236 |
| | DE | -1.0316 | -1.0316 | -1.0316 | 6.72E-16 |
| | BFP | -1.0316 | 2.0904 | -0.4357 | 0.6914 |
| | IFPA | -1.0316 | -1.0315 | -1.0316 | **6.72E-16** |
| $f_{17}$ | FPA | 0.3979 | 0.3979 | 0.3979 | 4.40E-06 |
| | BA | 0.3979 | 0.3979 | 0.3979 | 3.28E-10 |
| | DE | 0.3979 | 0.3979 | 0.3979 | 3.36E-16 |
| | BFP | 0.3979 | 7.7479 | 1.5591 | 1.3551 |
| | IFPA | 0.3979 | 0.3979 | 0.3979 | **3.36E-16** |
| $f_{18}$ | FPA | -1.0000 | -1.0000 | -1.0000 | 4.41E-09 |
| | BA | -1.0000 | -1.0000 | -1.0000 | 2.95E-10 |
| | DE | -1.0000 | -1.0000 | -1.0000 | 0.00E-00 |
| | BFP | -1.0000 | -4.17E-12 | -0.2467 | 0.3236 |
| | IFPA | -1.0000 | -1.0000 | -1.0000 | **0.00E-00** |
| $f_{19}$ | FPA | 0.1905 | 2.2203 | 0.8892 | 0.5018 |
| | BA | 8.03E-06 | 1.16E+02 | 4.1397 | 16.5430 |
| | DE | 7.86E-06 | 4.2382 | 0.1152 | 0.6160 |
| | BFP | 35.6926 | 1.37E+04 | 2.29E+03 | 2.38E+03 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{20}$ | FPA | -1.0000 | -0.9989 | -0.9998 | 1.89E-04 |
| | BA | -1.0000 | -0.7857 | -0.9159 | 0.0593 |
| | DE | -1.0000 | -0.9362 | -0.9897 | 0.0236 |
| | BFP | -1.0000 | -0.3691 | -0.7117 | 0.1650 |
| | IFPA | -1.0000 | -1.0000 | -1.0000 | **0.00E-00** |

Table 5.5 Results of IFPA compared with different other algorithms for population size 60

Case IV Population size 80

The effect of population size 80 on algorithms are analyzed in Table 5.6. For function $f_1$, best value is achieved by IFPA although BA is giving competitive result in terms of standard deviation. IFPA is able to achieve exact global optimum solution for function $f_2$. In case of functions $f_3$ to $f_7$, IFPA is getting better results while other algorithms are also comparable to IFPA. For functions $f_8$ and $f_9$, IFPA is closer to global solution but BA is giving competitive results to it. IFPA is much closer to optimum solution for $f_{10}$ and $f_{11}$ while no other algorithm is even closer to it. For functions $f_{12}$ and $f_{14}$, IFPA is able to reach global optimum solution and DE is closer to optimal solution. Best, worst and mean are same for all algorithms and standard deviation is also same for FPA, BA, DE and BFP except IFPA, hence it can not be said which algorithm will be best for $f_{13}$. For functions $f_{15}$ to $f_{17}$, IFPA is providing better optimal solution in terms of standard deviation although FPA, BA and DE are competitive to it. For $f_{18}$ to $f_{20}$, IFPA achieves exact global solution and no other algorithm is even comparable to it. In this case, IFPA is better for nineteen functions, DE for six functions, FPA, BA and BFP for one function each. Overall IFPA is best among all algorithms for population size 80.

| Function | Algorithm | Best | Worst | Mean | Standard deviation |
|---|---|---|---|---|---|
| $\boldsymbol{f_1}$ | FPA | 20.8726 | 21.1174 | 21.0266 | 0.0536 |
| | BA | 19.9860 | 20.0000 | 19.9989 | 0.0029 |
| | DE | 20.2985 | 20.8757 | 20.6359 | 0.1492 |
| | BFP | 20.9966 | 21.3104 | 21.1850 | 0.0644 |
| | IFPA | **7.3315** | 20.0000 | **19.7152** | 1.8008 |
| $\boldsymbol{f_2}$ | FPA | 1.05E-09 | 1.25E-06 | 1.07E-07 | 1.86E-07 |
| | BA | 4.10E-14 | 0.1175 | 0.0230 | 0.0303 |
| | DE | **0.00E-00** | 0.0203 | 0.0012 | 0.0040 |
| | BFP | 0.0211 | 0.4550 | 0.3006 | 0.1135 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $\boldsymbol{f_3}$ | FPA | 21.7667 | 59.3657 | 40.8387 | 7.6556 |
| | BA | 3.1669 | 23.5423 | 9.9604 | 4.17035 |
| | DE | 81.5756 | 2.42E+02 | 1.71E+02 | 36.6105 |
| | BFP | 1.12E+02 | 2.28E+02 | 1.78E+02 | 28.3373 |
| | IFPA | **0.1791** | **10.0330** | **3.0672** | **2.2177** |
| $\boldsymbol{f_4}$ | FPA | 27.9035 | 82.0064 | 57.4010 | 12.1957 |
| | BA | 52.6887 | 1.53E+02 | 1.01E+02 | 24.7632 |
| | DE | 3.86E+02 | 6.83E+02 | 5.54E+02 | 71.0289 |
| | BFP | 3.36E+02 | 6.78E+02 | 5.27E+02 | 94.8944 |
| | IFPA | **1.00E-03** | **1.0382** | **0.0809** | **0.1523** |
| $\boldsymbol{f_5}$ | FPA | 1.74E+02 | 2.28E+02 | 2.07E+02 | 12.5403 |
| | BA | 31.8419 | 1.75E+02 | 78.2262 | 32.0307 |
| | DE | 3.52E+02 | 4.79E+02 | 4.19E+02 | 29.8850 |
| | BFP | 3.09E+02 | 4.33E+02 | 3.78E+02 | 28.6785 |
| | IFPA | **16.9278** | **90.5413** | **42.6483** | 15.5038 |
| $\boldsymbol{f_6}$ | FPA | 1.63E+04 | 5.14E+04 | 3.34E+04 | 7.62E+03 |
| | BA | 2.19E+04 | 9.78E+04 | 5.49E+04 | 1.75E+04 |
| | DE | 2.03E+05 | 4.58E+05 | 3.63E+05 | 6.29E+04 |
| | BFP | 1.67E+05 | 5.14E+05 | 3.85E+05 | 6.94E+04 |
| | IFPA | **2.28E-03** | **0.9622** | **0.0788** | **0.1749** |
| $\boldsymbol{f_7}$ | FPA | 59.5566 | 67.9041 | 64.4727 | 1.8338 |
| | BA | 37.8480 | 57.4201 | 49.3426 | 4.8467 |
| | DE | 54.3677 | 71.1992 | 64.1690 | 4.0934 |
| | BFP | 68.6913 | 86.0600 | 80.0936 | 3.6844 |
| | IFPA | **20.1480** | **34.0870** | **26.3555** | 2.8949 |
| $\boldsymbol{f_8}$ | FPA | 8.2065 | 24.6333 | 15.3131 | 3.3769 |
| | BA | 1.29E-05 | 3.65E-05 | 1.82E-05 | 4.49E-06 |
| | DE | 1.23E+02 | 1.97E+02 | 1.64E+02 | 15.7469 |

| | | | | | |
|---|---|---|---|---|---|
| | BFP | 74.8528 | 1.89E+02 | 1.46E+02 | 25.1378 |
| | IFPA | **6.47E-07** | 4.13E-04 | 4.82E-05 | 8.82E-05 |
| $f_9$ | FPA | 1.99E+02 | 7.90E+02 | 4.42E+02 | 1.54E+02 |
| | BA | 0.0093 | 27.4566 | 1.0603 | 4.4123 |
| | DE | 2.70E+03 | 1.76E+04 | 8.35E+03 | 3.14E+03 |
| | BFP | 3.58E+03 | 1.82E+04 | 1.15E+04 | 3.57E+03 |
| | IFPA | **7.20E-04** | **0.1434** | **0.0161** | **0.0230** |
| $f_{10}$ | FPA | 6.57E-05 | 0.0030 | 9.12E-04 | 7.15E-04 |
| | BA | 3.91E-09 | 5.96E-08 | 2.40E-08 | 1.25E-08 |
| | DE | 0.2674 | 1.1127 | 0.6465 | 0.2176 |
| | BFP | 0.0314 | 0.9455 | 0.3003 | 0.1863 |
| | IFPA | **8.81E-26** | **2.78E-15** | **6.00E-17** | **3.93E-16** |
| $f_{11}$ | FPA | 3.93E+02 | 1.35E+03 | 7.87E+02 | 1.93E+02 |
| | BA | 1.6111 | 3.09E+02 | 1.01E+02 | 80.3935 |
| | DE | 4.36E+03 | 1.12E+04 | 8.71E+03 | 1.28E+03 |
| | BFP | 4.73E+03 | 1.23E+04 | 8.52E+03 | 1.60E+03 |
| | IFPA | **4.86E-05** | **7.63E-03** | **1.25E-03** | **1.64E-03** |
| $f_{12}$ | FPA | 5.62E-10 | 4.19E-07 | 6.73E-08 | 8.31E-08 |
| | BA | 6.21E-15 | 0.2004 | 0.0265 | 0.0393 |
| | DE | **0.00E-00** | 2.42E-04 | 4.89E-06 | 3.43E-05 |
| | BFP | 9.30E-03 | 0.4776 | 0.2827 | 0.1333 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{13}$ | FPA | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | BA | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | DE | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | BFP | 0.5000 | 0.5000 | 0.5000 | 0.00E-00 |
| | IFPA | 0.5000 | 0.5000 | 0.5000 | 1.81E-10 |
| $f_{14}$ | FPA | 7.66E-11 | 1.70E-06 | 1.73E-07 | 2.81E-07 |
| | BA | 4.92E-12 | 0.5966 | 0.0520 | 0.1585 |
| | DE | 0.00E-00 | 1.72E-30 | 4.29E-32 | 2.44E-31 |
| | BFP | 4.48E-06 | 2.5877 | 0.5233 | 0.6181 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_{15}$ | FPA | 0.9980 | 0.9980 | 0.9980 | 7.64E-06 |
| | BA | 0.9980 | 23.8094 | 8.6015 | 7.1839 |
| | DE | 0.9980 | 1.9920 | 1.0377 | 0.1967 |
| | BFP | 1.6264 | 3.86E+02 | 47.9478 | 77.7661 |
| | IFPA | 0.9980 | 0.9980 | 0.9980 | **1.12E-16** |
| $f_{16}$ | FPA | -1.0316 | -1.0316 | -1.0316 | 6.15E-08 |
| | BA | -1.0316 | -1.0316 | -1.0316 | 5.28E-10 |
| | DE | -1.0316 | -1.0316 | -1.0316 | 6.72E-16 |
| | BFP | -1.0316 | 0.1483 | -0.7413 | 0.34678 |
| | IFPA | -1.0316 | -1.0315 | -1.0316 | **6.72E-16** |
| $f_{17}$ | FPA | 0.3979 | 0.3979 | 0.3979 | 2.13E-06 |
| | BA | 0.3979 | 0.3979 | 0.3979 | 2.02E-10 |
| | DE | 0.3979 | 0.3979 | 0.3979 | 3.36E-16 |
| | BFP | 0.3979 | 2.7586 | 0.9279 | 0.6449 |
| | IFPA | 0.3979 | 0.3979 | 0.3979 | **3.36E-16** |
| $f_{18}$ | FPA | -1.000 | -1.000 | -1.000 | 4.54E-10 |
| | BA | -1.000 | -1.000 | -1.000 | 1.70E-10 |
| | DE | -1.000 | -1.000 | -1.000 | 0.00E-00 |
| | BFP | -1.000 | -8.7431e-12 | -0.3868 | 0.3925 |
| | IFPA | -1.000 | -1.000 | -1.000 | **0.00E-00** |
| $f_{19}$ | FPA | 0.03867 | 2.2997 | 0.7694 | 0.5073 |
| | BA | 1.07E-06 | 7.8389 | 1.2438 | 1.9710 |
| | DE | 1.47E-06 | 0.0760 | 0.0030 | 0.0111 |
| | BFP | 40.6429 | 1.42E+04 | 2.42E+03 | 2.54E+03 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |

| | | | | | |
|---|---|---|---|---|---|
| $f_{20}$ | FPA | -1.0000 | -0.9981 | -0.9998 | 3.59e-04 |
| | BA | -1.0000 | -0.7857 | -0.9207 | 0.0519 |
| | DE | -1.0000 | -0.9362 | -0.9936 | 0.0193 |
| | BFP | -1.0000 | -0.2888 | -0.7150 | 0.1637 |
| | IFPA | -1.0000 | -1.0000 | -1.0000 | **0.00E-00** |

Table 5.6 Results of IFPA compared with different other algorithms for population size 80

The performance of IFPA is evaluated for different population sizes (20, 40, 60 and 80) on twenty benchmarking functions for each population size. The results obtained by IFPA is better for all population sizes but results are closer to global minima in case of population size 80. It is due to increase in diversity for higher population size which leads to improvement in exploring ability hence helps in avoiding local optima. That's why we are using 80 population size as standard for evaluation the performance of proposed approach and other standard algorithms. Hence, it can be concluded that IFPA is best and BFP is worst among all algorithms. The convergence graphs are also drawn for population size 80 only.

### 5.1.4 Effect of dimension

In the following subsections, effect of dimensions 30, 50 and 100 are analyzed for testing the performance of IFPA. Here, the population size 80 is considered for the evaluation their performance since this population size is giving better results as compared with other population sizes. IFPA is compared with other standard algorithms like BA, DE, BFP and original FPA. Out of twenty benchmarking functions, eleven functions are used for evaluating the performance because other benchmark functions are fixed dimension based. Table 5.6 shows the effect of dimension 30, Table 5.7 provides effect for dimension 50 and Table 5.8 for dimension 100.

Case I Effect of dimension 30

The results for the dimension 30 are analyzed in Table 5.6. It has been already discussed before in population size 80. In this particular case, out of eleven benchmark functions, IFPA is better for almost all functions.

Case II Effect of dimension 50

The results for dimension size 50 are given in Table 5.7. For function $f_1$, IFPA can be considered to be best algorithm. IFPA has reached to exact optimum solution for $f_2$ while for $f_3$ and $f_4$, IFPA is much closer to global optimum. For function $f_5$, IFPA achieved better solution even and for $f_6$, IFPA is comparable to other algorithm results. IFPA is better in terms of best, worst and mean for $f_7$ while FPA is comparable in terms of standard deviation. IFPA is giving competitive results as compared with other algorithm results for $f_8$, $f_9$ and $f_{11}$. For function $f_{10}$, IFPA is almost closer to global optima and shows much superior performance while BA is competitive. Hence, IFPA is better for almost all functions.

| Function | Algorithm | Best | Worst | Mean | Standard deviation |
|---|---|---|---|---|---|
| $f_1$ | FPA | 21.1094 | 21.2645 | 21.2008 | 0.0356 |
| | BA | 19.9808 | 20.0000 | 19.9994 | 0.0027 |
| | DE | 20.5248 | 21.0884 | 20.8762 | 0.1341 |
| | BFP | 21.1781 | 21.4231 | 21.3183 | 0.0542 |
| | IFPA | **19.0609** | 20.0003 | **19.9812** | 0.1328 |
| $f_2$ | FPA | 4.40E-10 | 5.44E-07 | 7.46E-08 | 9.46E-08 |
| | BA | 2.22E-14 | 0.1580 | 0.0354 | 0.0418 |
| | DE | 0.00E-00 | 0.0187 | 0.0023 | 0.0047 |
| | BFP | 0.0063 | 0.4661 | 0.2565 | 0.1245 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_3$ | FPA | 62.4518 | 1.18e+02 | 87.0456 | 13.1236 |
| | BA | 6.1333 | 53.1223 | 18.1310 | 8.4910 |
| | DE | 2.62E+02 | 4.58e+02 | 3.81e+02 | 45.8663 |
| | BFP | 2.17E+02 | 4.49e+02 | 3.50e+02 | 51.1348 |
| | IFPA | **0.4059** | **18.0673** | **5.4213** | **3.3951** |
| $f_4$ | FPA | 84.8071 | 2.16E+02 | 1.33E+02 | 28.9707 |
| | BA | 1.02E+02 | 3.14E+02 | 1.84E+02 | 52.9408 |
| | DE | 8.90E+02 | 1.22E+03 | 1.05E+03 | 78.0804 |
| | BFP | 4.41E+02 | 1.24E+03 | 9.90E+02 | 1.62E+02 |
| | IFPA | **0.5172** | **1.6376** | **1.0088** | **0.1962** |
| $f_5$ | FPA | 3.67E+02 | 4.48e+02 | 4.08e+02 | 19.7278 |
| | BA | 59.7052 | 2.43e+02 | 1.18e+02 | 41.5525 |
| | DE | 6.31E+02 | 8.28e+02 | 7.56e+02 | 39.2757 |
| | BFP | 5.32E+02 | 7.52e+02 | 6.77e+02 | 47.1737 |
| | IFPA | **33.4254** | 1.06e+02 | **69.0190** | **16.2925** |
| $f_6$ | FPA | 8.81E+04 | 2.15E+05 | 1.48E+05 | 2.99E+04 |
| | BA | 8.36E+04 | 3.46E+05 | 1.64E+05 | 6.02E+04 |
| | DE | 7.32E+05 | 1.48E+06 | 1.21E+06 | 1.26E+05 |
| | BFP | 6.96E+05 | 1.43E+06 | 1.15E+06 | 1.86E+05 |
| | IFPA | **7.3884** | 2.37E+03 | 1.38E+02 | 3.36E+02 |
| $f_7$ | FPA | 59.5566 | 67.9041 | 64.4727 | 1.8338 |
| | BA | 37.8480 | 57.4201 | 49.3426 | 4.8467 |
| | DE | 54.3677 | 71.1992 | 64.1690 | 4.0934 |
| | BFP | 68.6913 | 86.0600 | 80.0936 | 3.6844 |
| | IFPA | **20.1480** | **34.0870** | **26.3555** | 2.8949 |
| $f_8$ | FPA | 23.0414 | 54.2363 | 37.5318 | 6.6127 |
| | BA | 4.72E-05 | 3.5223 | 0.1310 | 0.5251 |
| | DE | 2.26E+02 | 3.48E+02 | 3.05E+02 | 23.3656 |
| | BFP | 1.28E+02 | 3.483E+02 | 2.81E+02 | 48.3507 |
| | IFPA | 0.0011 | **0.3290** | **0.0345** | **0.0635** |
| $f_9$ | FPA | 7.40E+02 | 2.62E+03 | 1.54E+03 | 4.46E+02 |
| | BA | 0.1102 | 16.7843 | 4.6541 | 4.9414 |
| | DE | 1.06E+04 | 4.19E+04 | 2.72E+04 | 6.69E+03 |
| | BFP | 1.06E+04 | 4.14E+04 | 2.62E+04 | 7.88E+03 |
| | IFPA | **0.0969** | **3.6138** | **1.1483** | **0.8493** |
| $f_{10}$ | FPA | 5.07E-05 | 0.0061 | 0.0011 | 0.0010 |
| | BA | 4.55E-09 | 5.08E-08 | 2.33E-08 | 1.18E-08 |
| | DE | 0.3267 | 1.5688 | 0.8224 | 0.2595 |
| | BFP | 0.0380 | 0.9186 | 0.3814 | 0.2088 |
| | IFPA | **1.18E-23** | **1.88E-13** | **7.32E-15** | **2.79E-14** |

| | | | | | |
|---|---|---|---|---|---|
| $f_{11}$ | FPA | 1.25E+03 | 4.70E+03 | 3.29E+03 | 7.93E+02 |
| | BA | 62.5021 | 1.69E+03 | 7.66E+02 | 4.04E+02 |
| | DE | 2.13E+04 | 3.22E+04 | 2.82E+04 | 2.27E+03 |
| | BFP | 1.32E+04 | 3.46E+04 | 2.66E+04 | 5.11E+03 |
| | IFPA | **0.0716** | **8.3485** | **1.6956** | **1.8852** |

Table 5.7 Results of IFPA compared with other algorithms for dimension size of 50

Case III Effect of dimension 100

The result analysis for the dimension 100 is shown in Table 5.8. BA is better for function $f_2$. IFPA has reached global optima for $f_2$ and DE is competitive in terms of best value. For functions $f_3$ and $f_4$, IFPA is nearest to optimal solution as compared to other algorithms. IFPA is comparable as being compared to other algorithms for $f_5$ and $f_6$. For $f_7$, best, worst and mean values are better for IFPA while BA is competitive in terms of standard deviation. For $f_8$ and $f_9$, IFPA is better in terms of worst, mean and standard deviation and BA results are quite comparable. For $f_{10}$, IFPA is much closer to optimum solution and FPA and BA are giving competitive results. Results of all algorithms are only comparable in case of $f_{11}$. It can be concluded that IFPA is better for nine functions only.

| Function | Algorithm | Best | Worst | Mean | Standard deviation |
|---|---|---|---|---|---|
| $f_1$ | FPA | 21.3358 | 21.4073 | 21.3767 | 0.0185 |
| | BA | 19.9908 | 20.0001 | 19.9995 | 0.0015 |
| | DE | 20.9477 | 21.3508 | 21.1627 | 0.0986 |
| | BFP | 21.3698 | 21.5142 | 21.4500 | 0.0273 |
| | IFPA | 20.0000 | 20.0060 | 20.0002 | **8.82e-04** |
| $f_2$ | FPA | 1.92e-10 | 8.57e-07 | 9.71e-08 | 1.38e-07 |
| | BA | 0.0031 | 0.1680 | 0.0398 | 0.0415 |
| | DE | 0.00E-00 | 0.0223 | 0.0015 | 0.0044 |
| | BFP | 0.0600 | 0.4536 | 0.2811 | 0.1205 |
| | IFPA | **0.00E-00** | **0.00E-00** | **0.00E-00** | **0.00E-00** |
| $f_3$ | FPA | 1.31e+02 | 2.81e+02 | 2.00e+02 | 36.1719 |
| | BA | 23.3232 | 86.9886 | 43.1687 | 12.3256 |
| | DE | 7.63e+02 | 1.01e+03 | 9.13e+02 | 58.6408 |
| | BFP | 5.15e+02 | 1.06e+03 | 8.55e+02 | 1.09e+02 |
| | IFPA | **4.4487** | **28.3031** | **13.0378** | **5.1641** |
| $f_4$ | FPA | 2.08e+02 | 4.62e+02 | 3.30e+02 | 58.8268 |
| | BA | 2.09e+02 | 5.94e+02 | 3.76e+02 | 83.3498 |
| | DE | 2.05e+03 | 2.54e+03 | 2.37e+03 | 1.20e+02 |
| | BFP | 1.27e+03 | 2.62e+03 | 2.27e+03 | 3.18e+02 |
| | IFPA | **2.6801** | **12.5971** | **6.5113** | **2.2222** |
| $f_5$ | FPA | 8.82e+02 | 1.03e+03 | 9.56e+02 | 36.8476 |
| | BA | **98.5425** | 3.62e+02 | 2.04e+02 | 59.2076 |
| | DE | 1.45e+03 | 1.69e+03 | 1.60e+03 | 45.7660 |
| | BFP | 1.21e+03 | 1.62e+03 | 1.48e+03 | 98.6605 |
| | IFPA | 1.09e+02 | 2.90e+02 | 1.95e+02 | 43.9923 |
| $f_6$ | FPA | 3.60e+05 | 1.06e+06 | 7.07e+05 | 1.66e+05 |
| | BA | 4.10e+05 | 1.87e+06 | 8.01e+05 | 2.34e+05 |
| | DE | 4.88e+06 | 6.09e+06 | 5.48e+06 | 3.23e+05 |
| | BFP | 2.76e+06 | 6.41e+06 | 5.18e+06 | 9.38e+05 |

| | | | | | |
|---|---|---|---|---|---|
| | IFPA | **3.16e+03** | 3.68e+04 | 1.07e+04 | 5.35e+03 |
| $f_7$ | FPA | 1.31e+02 | 1.53e+02 | 67.3714 | 4.2141 |
| | BA | 79.9709 | 1.10e+02 | 96.8159 | 8.0767 |
| | DE | 1.28e+02 | 1.61e+02 | 1.46e+02 | 7.1105 |
| | BFP | 1.29e+02 | 1.79e+02 | 1.67e+02 | 10.0361 |
| | IFPA | **50.1762** | **73.3272** | **61.8410** | 5.1555 |
| $f_8$ | FPA | 48.8038 | 1.34e+02 | 92.4915 | 19.6283 |
| | BA | **4.78e-04** | 26.7244 | 6.0133 | 5.1846 |
| | DE | 6.29e+02 | 7.45e+02 | 6.92e+02 | 26.4848 |
| | BFP | 3.39e+02 | 7.48e+02 | 6.11e+02 | 1.13e+02 |
| | IFPA | 0.5754 | 3.2718 | 1.5535 | 0.6481 |
| $f_9$ | FPA | 2.06E+03 | 7.65E+03 | 4.85E+03 | 1.18e+03 |
| | BA | 11.3558 | 4.10E+02 | 1.03E+02 | 96.3331 |
| | DE | 6.44E+04 | 1.19E+05 | 9.11E+04 | 1.29e+04 |
| | BFP | 3.43E+04 | 1.20E+05 | 8.17E+04 | 2.23e+04 |
| | IFPA | 12.4679 | **87.7884** | **32.6844** | **14.1584** |
| $f_{10}$ | FPA | 1.29E-04 | 0.0108 | 0.0028 | 0.0023 |
| | BA | 6.02E-09 | 7.28E-08 | 2.29E-08 | 1.24E-08 |
| | DE | 0.4974 | 2.0086 | 1.1470 | 0.3142 |
| | BFP | 0.0868 | 1.3223 | 0.6850 | 0.3203 |
| | IFPA | **3.62E-17** | **2.16E-10** | **5.85E-12** | **3.08E-11** |
| $f_{11}$ | FPA | 7.91E+03 | 2.57E+04 | 1.73E+04 | 4.00E+03 |
| | BA | 1.48E+03 | 1.25E+04 | 6.49E+03 | 2.56E+03 |
| | DE | 1.08E+05 | 1.39E+05 | 1.26E+05 | 7.42E+03 |
| | BFP | 7.36E+04 | 1.42E+05 | 1.19E+05 | 1.59E+04 |
| | IFPA | **1.21E+02** | 5.82E+02 | 2.72E+02 | **1.09E+02** |

Table 5.8 Results of IFPA compared with other algorithms for dimension size of 100

As it can be seen from above sub-sections that IFPA outperforms all algorithms. Out of eleven functions, for dimension size 30, IFPA is better for all functions, BA and FPA for 2 functions each and DE for one function. For dimension of size 50, IFPA is also better for almost all functions while BA, DE and FPA are better for one function each. For dimension size 100, IFPA is better for nine functions, BA for three functions and FPA for two functions and DE for one function. Out of total function evaluation of 33 benchmark functions, IFPA is best for thirty one functions, BA for six functions, FPA for five functions, DE for three functions and BFP is not good for even any function. Hence, IFPA shows superior performance as compared to other state-of-art algorithms. It can also be seen that as the dimension of the function increases, number of variables and complexity of algorithm also increases so performance of IFPA starts deteriorating. Still IFPA performance does not deteriorate much and results are better in comparison to other standard algorithms.

### 5.1.5 Statistical Testing

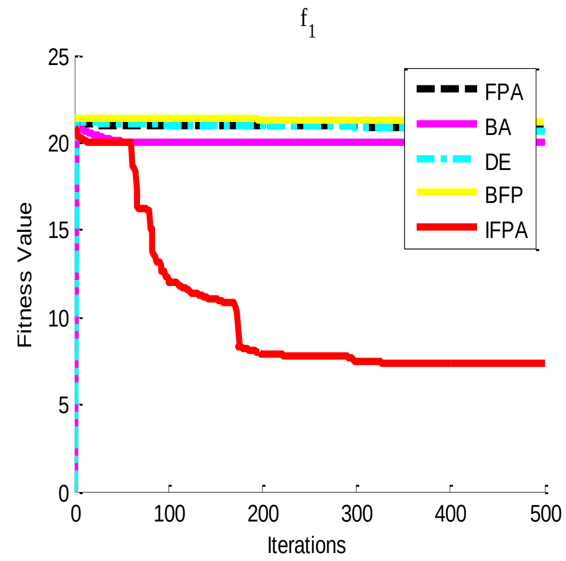
Here, Wilcoxon's rank test [111] is carried out for testing the performance of proposed approach with other algorithms. This test is performed for confirming whether the proposed approach will be better or significant for a given problem in comparison with other popular existing algorithms. Hence, this test gives the p values which gives the difference between best algorithm and other algorithm. The algorithm is considered to be statistical significant if the value of $p < 0.05$. So, here best algorithm is denoted by NA since best algorithm can not be compared to itself while other algorithms give the value in comparison with the best algorithm. The '~' sign signifies that the algorithm is comparable, its value cannot be compared with other algorithms. It can be seen form Table 5.9, that IFPA shows better statistical performance for nineteen functions, FPA, DE, BA and BFP for one each. Hence, it can be said that IFPA is much better in comparison to other algorithms for even statistical testing.

| Function | FPA | BA | DE | BFP | IFPA |
|---|---|---|---|---|---|
| $f_1$ | 3.50E-19 | 0.0169 | 3.50E-19 | 3.50E-19 | NA |
| $f_2$ | 3.31E-20 | 3.31E-20 | 1.83E-10 | 3.31E-20 | NA |
| $f_3$ | 7.06E-18 | 8.10E-15 | 7.06E-18 | 7.06E-18 | NA |
| $f_4$ | 7.06E-18 | 7.06E-18 | 7.06E-18 | 7.06E-18 | NA |
| $f_5$ | 7.06E-18 | 9.81E-11 | 7.06E-18 | 7.06E-18 | NA |
| $f_6$ | 7.06E-18 | 7.06E-18 | 7.06E-18 | 7.06E-18 | NA |
| $f_7$ | 7.06E-18 | 7.06E-18 | 7.06E-18 | 7.06E-18 | NA |
| $f_8$ | 7.06E-18 | 0.2010 | 7.06E-18 | 7.06E-18 | NA |
| $f_9$ | 7.06E-18 | 2.67E-12 | 7.06E-18 | 7.06E-18 | NA |
| $f_{10}$ | 7.06E-18 | 7.06E-18 | 7.06E-18 | 7.06E-18 | NA |
| $f_{11}$ | 7.06E-18 | 7.06E-18 | 7.06E-18 | 7.06E-18 | NA |
| $f_{12}$ | 3.31E-20 | 3.31E-20 | 0.0822 | 3.31E-20 | NA |
| $f_{13}$ | NA | ~ | ~ | ~ | 3.30E-20 |
| $f_{14}$ | 3.31E-20 | 3.31E-20 | 0.0123 | 3.31E-20 | NA |
| $f_{15}$ | 9.11E-20 | 9.07E-20 | 1.14E-08 | 9.11E-20 | NA |
| $f_{16}$ | 3.31E-20 | 3.31E-20 | ~ | 3.31E-20 | NA |
| $f_{17}$ | 3.31E-20 | 3.31E-20 | ~ | 3.31E-20 | NA |
| $f_{18}$ | 3.31E-20 | 3.31E-20 | ~ | 3.31E-20 | NA |
| $f_{19}$ | 3.31E-20 | 3.31E-20 | 3.31E-20 | 3.31E-20 | NA |
| $f_{20}$ | 3.31E-20 | 3.31E-20 | 0.0231 | 3.31E-20 | NA |

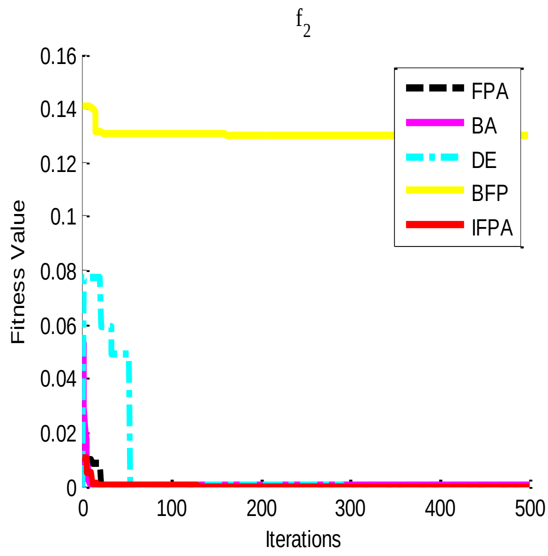
Table 5.9 p test values of different algorithms

### 5.1.6 Convergence Curves

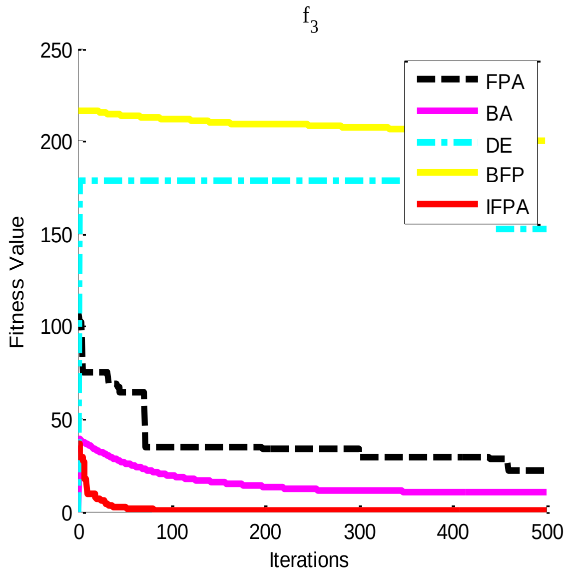
The convergence graphs for the proposed approach are shown in Figure 5.1. These curves are basically used for checking the time used by the algorithms for achieving global solution. These convergence curves of proposed approach are compared with other algorithms like FPA, BA, DE and BFP. These graphs are used for showing the superior performance of proposed approach with other algorithms.



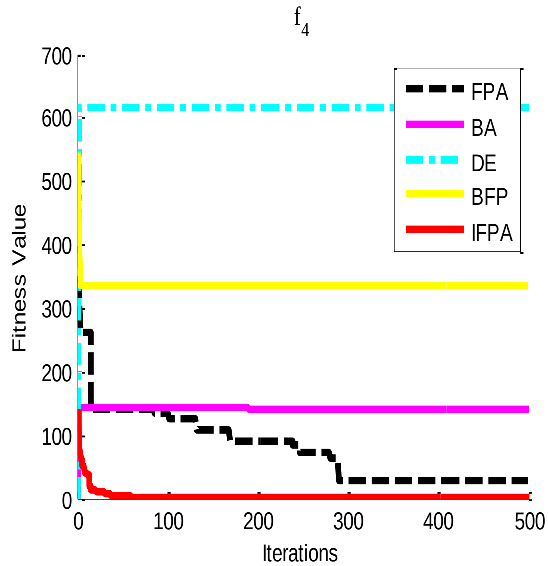
$f_1$
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

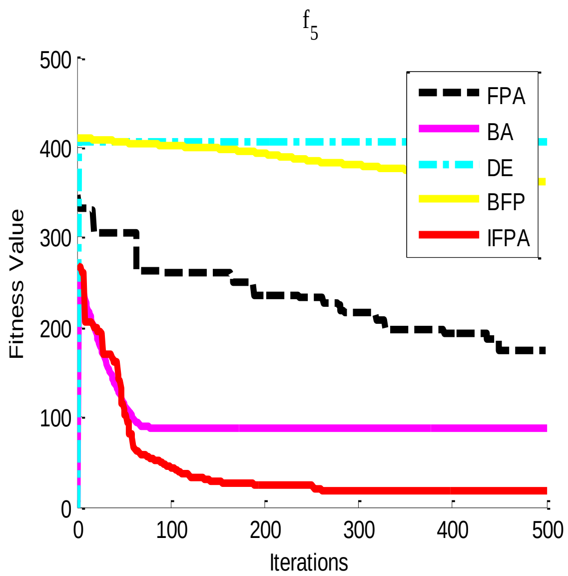
$f_2$
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

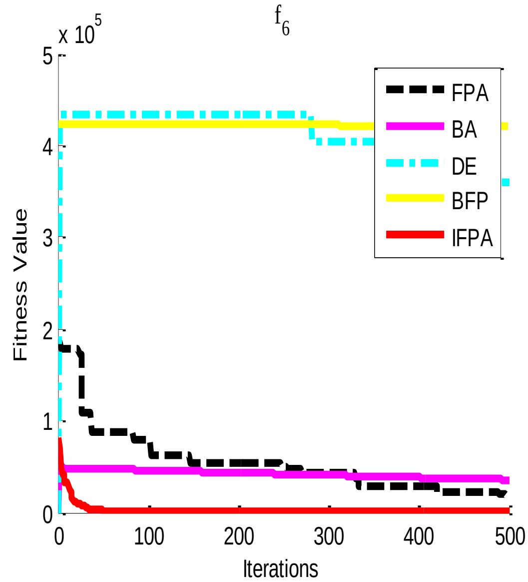
$f_3$
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

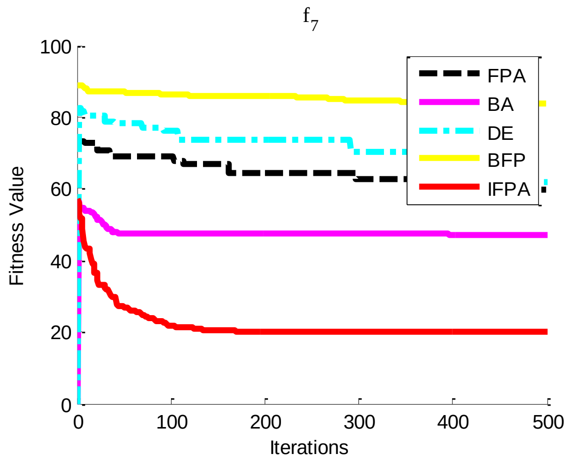
$f_4$
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

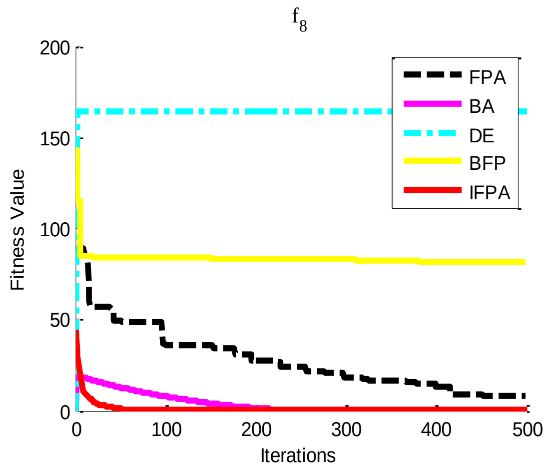



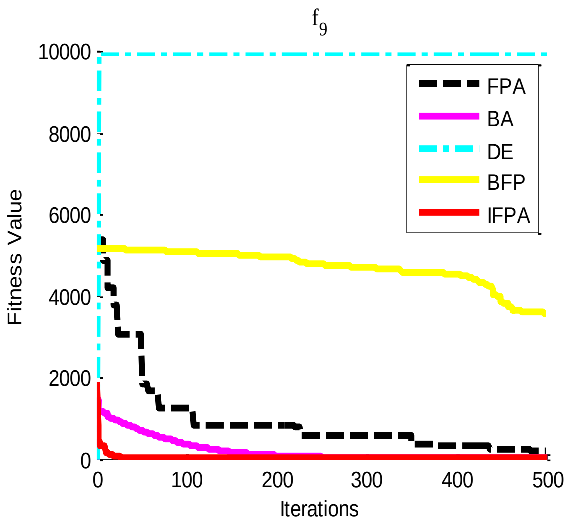
$f_5$
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

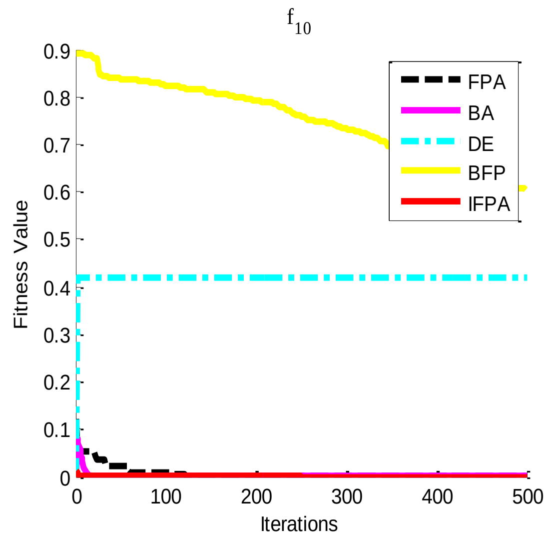



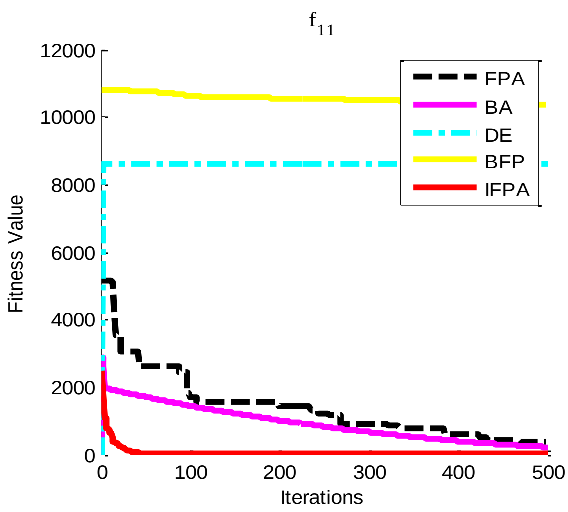
$f_6$
x $10^5$
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

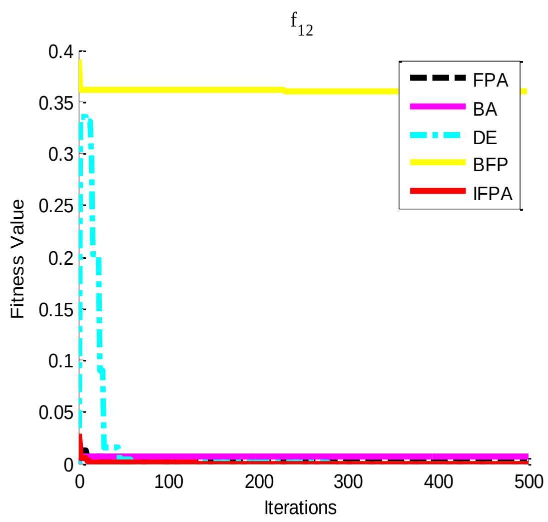

f7
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

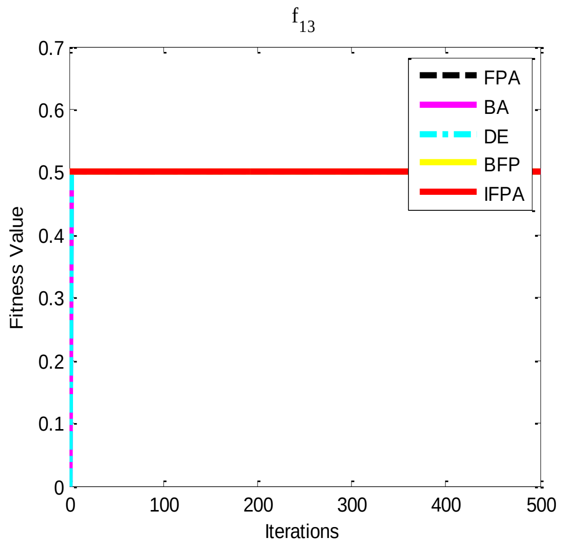

f8
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

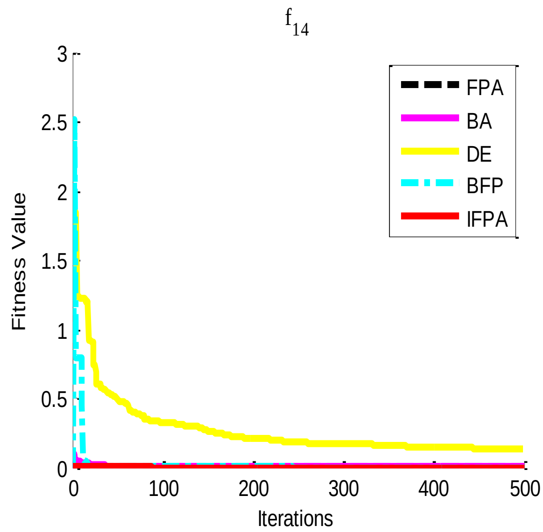

f9
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

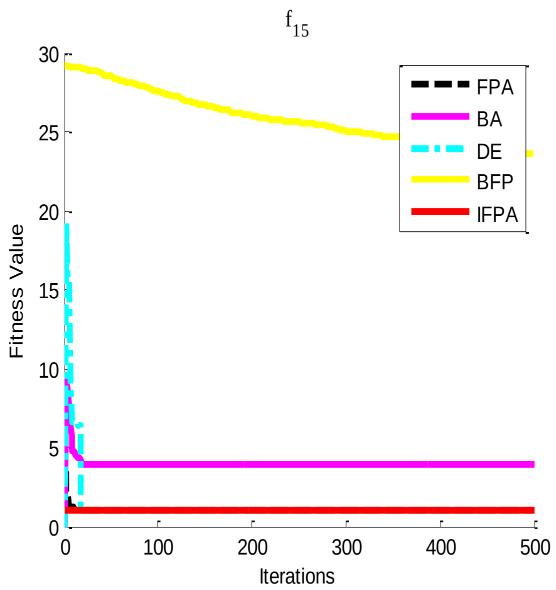

f10
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

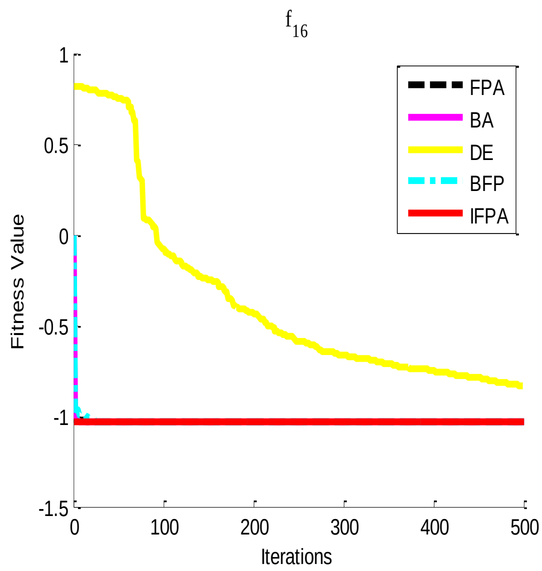

f11
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

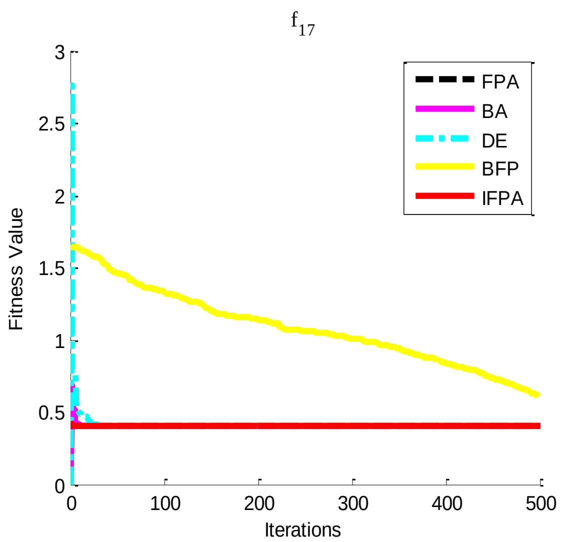

f12
Fitness Value
Iterations
FPA
BA
DE
BFP
IFPA

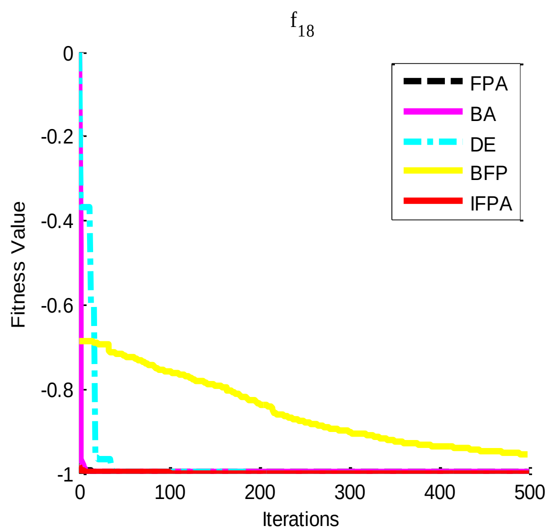

f13
Fitness Value
Iterations
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
100
200
300
400
500
FPA
BA
DE
BFP
IFPA


f14
Fitness Value
Iterations
0
0.5
1
1.5
2
2.5
3
100
200
300
400
500
FPA
BA
DE
BFP
IFPA


f15
Fitness Value
Iterations
0
5
10
15
20
25
30
100
200
300
400
500
FPA
BA
DE
BFP
IFPA


f16
Fitness Value
Iterations
-1.5
-1
-0.5
0
0.5
1
100
200
300
400
500
FPA
BA
DE
BFP
IFPA


f17
Fitness Value
Iterations
0
0.5
1
1.5
2
2.5
3
100
200
300
400
500
FPA
BA
DE
BFP
IFPA


f18
Fitness Value
Iterations
-1
-0.8
-0.6
-0.4
-0.2
0
100
200
300
400
500
FPA
BA
DE
BFP
IFPA

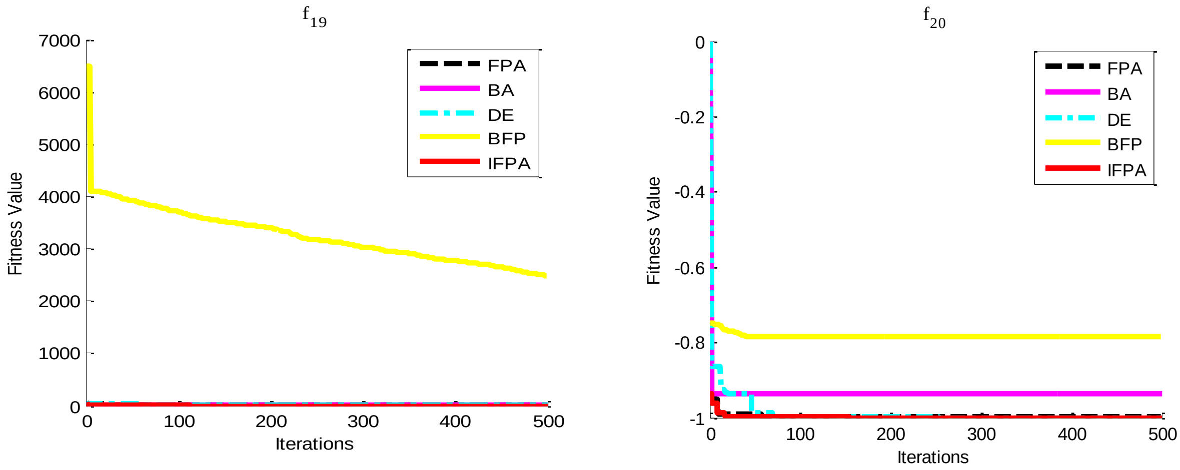



Figure 5.1 Convergence graphs of proposed approach compared with other different algorithms

## 5.2 SIMULATION RESULTS FOR SYNTHESIS OF CIRCULAR ANTENNA ARRAY

In this section, IFPA is used for synthesis of non-uniform circular array having different number of elements. The main aim of synthesis of antenna is to determine the geometrical and electrical properties for CAA for reducing the SLL and FNBW. This is basically done by controlling the excitation of current and position of antenna array. The objective function used for the achievement of desired characteristics are given by:

$$F = w_1 * SLL + w_2 * FNBW \tag{5.1}$$

where $w_1$ and $w_2$ are weighting coefficients, FNBW is first null beam width in degrees. The main aim of this optimization problem is to minimize $F$ for obtaining the set of values for amplitude [$I_1$, $I_2$, $I_3$.......... $I_N$] and distance between the elements [$d_1$, $d_2$, $d_3$.......... $d_N$]. The separation between the elements are in the range [0, λ] and the amplitude varied between the range [0, 1], where λ is wavelength of signal. The parameter settings are as follows:

Population size =50, Maximum generations=500, $w_1$=11, $w_2$=1.

In this section, simulation results for reducing the SLL for three non-uniform CAAs with 8, 10 and 12 elements are discussed. For N=8, there are 16 parameters to be optimized that is eight parameters are of current excitation and eight for inter-element separations. Table 5.10 shows the optimized currents and positions along with obtained SLL and FNBW for IFPA in comparison with PSO [95], FA [98], GA [94], BBO [97], SA [96] and CSO [100]. It can be seen that side lobes are suppressed to -13.90 dB as compared to -10.79 dB, -13 dB, -9.81 dB, -12.24 dB and -12 dB for PSO, FA, GA, BBO and SA respectively. Only CSO has SLL of -15.77 dB which is better than IFPA. It can be seen from Table 5.10 that FNBW is more for

IFPA as compared to other standard algorithms. It is well known that there is a trade-off between SLL and FNBW. It means reduction in SLL will increase FNBW and vice-versa. It can also be seen that in case of PSO, FA, GA, BBO, SA, CSO and IFPA, circumference of antenna is 4.49λ, 4.65λ, 4.40λ, 9.07λ, 5.87λ, 5.79 λ and 4.37 λ respectively. It can easily be concluded that antenna size will be smaller for IFPA as compared to other algorithms. Due to this, IFPA will have more FNBW as compared with other algorithms. The radiation pattern of 8-element circular array of IFPA in comparison with other algorithms are shown in Figure 5.2.

| Algorithms | $[I_1, I_2, I_3, \ldots I_n]$ | | | $[d_1, d_2, d_3, \ldots d_n]$ in λ | | | SLL(dB) | FNBW(°) |
|---|---|---|---|---|---|---|---|---|
| PSO[95] | 0.7765 | 0.3928 | 0.6069 | 0.3590 | 0.5756 | 0.2494 | -10.79 | 69.80 |
| | 0.8446 | 1.0000 | 0.7015 | 0.7638 | 0.6025 | 0.8311 | | |
| | 0.9321 | 0.3583 | | 0.7809 | 0.3308 | | | |
| FA[98] | 0.8251 | 0.7018 | 0.9962 | 0.3377 | 0.8274 | 0.8575 | -13.00 | 72.40 |
| | 0.9964 | 0.4933 | 0.5697 | 0.6306 | 0.8538 | 0.7092 | | |
| | 0.4228 | 0.1669 | | 0.2499 | 0.1895 | | | |
| GA[94] | 0.3289 | 0.2537 | 0.7849 | 0.1739 | 0.3144 | 0.6620 | -9.81 | 68.80 |
| | 1.0000 | 0.9171 | 0.5183 | 0.7425 | 0.6297 | 0.8929 | | |
| | 0.6176 | 0.4612 | | 0.4633 | 0.5267 | | | |
| BBO[97] | 1.0000 | 0.6736 | 0.1678 | 0.6341 | 1.0000 | 1.8892 | -12.24 | 38.80 |
| | 1.0000 | 0.9088 | 0.6553 | 0.8456 | 0.5693 | 1.1639 | | |
| | 0.7571 | 1.0000 | | 1.3329 | 1.6367 | | | |
| SA[96] | 0.3047 | 0.4840 | 0.7751 | 0.9997 | 0.7743 | 0.9042 | -12.00 | 54.40 |
| | 0.9867 | 0.3371 | 0.4422 | 0.5652 | 0.8056 | 0.7818 | | |
| | 0.4067 | 0.6807 | | 0.5848 | 0.4594 | | | |
| CSO[100] | 0.2592 | 0.0000 | 0.1459 | 0.3600 | 0.5835 | 0.7781 | -15.77 | 75 |
| | 0.3209 | 0.3342 | 0.1457 | 0.9010 | 0.5432 | 0.9022 | | |
| | 0.2616 | 0.1938 | | 1.3898 | 0.3421 | | | |
| IFPA | 0.5594 | 0.1961 | 0.2477 | 0.3907 | 0.5677 | 0.2186 | -13.90 | 79.20 |
| | 0.9627 | 0.9673 | 0.2840 | 0.7771 | 0.5962 | 0.5094 | | |
| | 0.6448 | 0.4623 | | 0.8910 | 0.4214 | | | |

Table 5.10 Comparison of results obtained by IFPA with other algorithms for N=8 elements

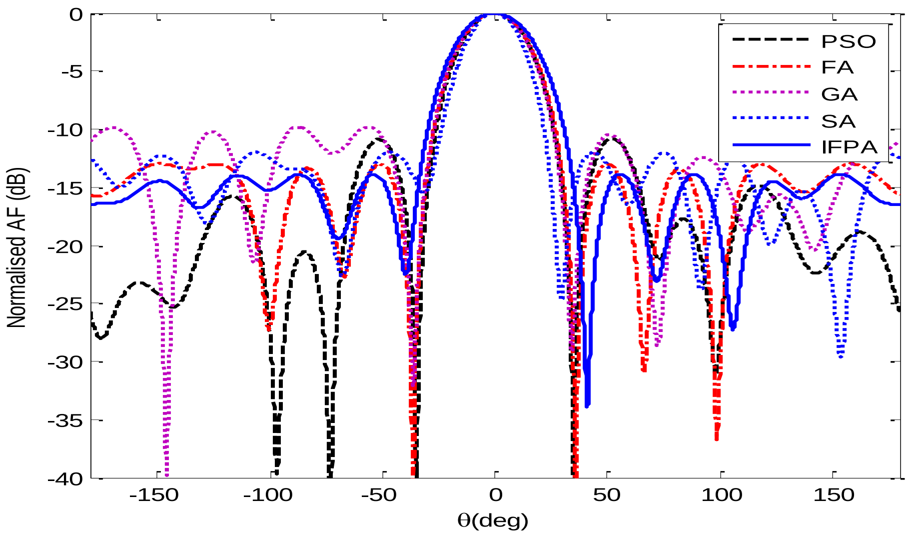



Figure 5.2 Radiation pattern obtained by non-uniform CAA for N=8 elements

For N=10, the parameters which are to be optimized are 20, ten are current excitation weights and ten are inter-element spacing. Table 5.11 shows the reduction of SLL as -13.81 dB as result of optimizing by IFPA. The SLL obtained for PSO [95], FA [98], GA [94], BBO [97], SA [96] and CSO [100] are -12.30 dB, -13.30 dB, -9.81 dB, -13.95 dB and -13.00 dB respectively. IFPA is better in suppressing SLL as compared to other algorithm except BBO and CSO. The circumference of antenna for PSO, FA, GA, BBO, SA, CSO and IFPA are 5.90λ , 6.071λ, 6.08λ 9.23 λ, 8.021λ, 10.63λ and 5.85λ respectively. Again for IFPA antenna array the size of the antenna is less. The radiation pattern for 10-element array of IFPA in comparison with other algorithms are shown in Figure 5.3.

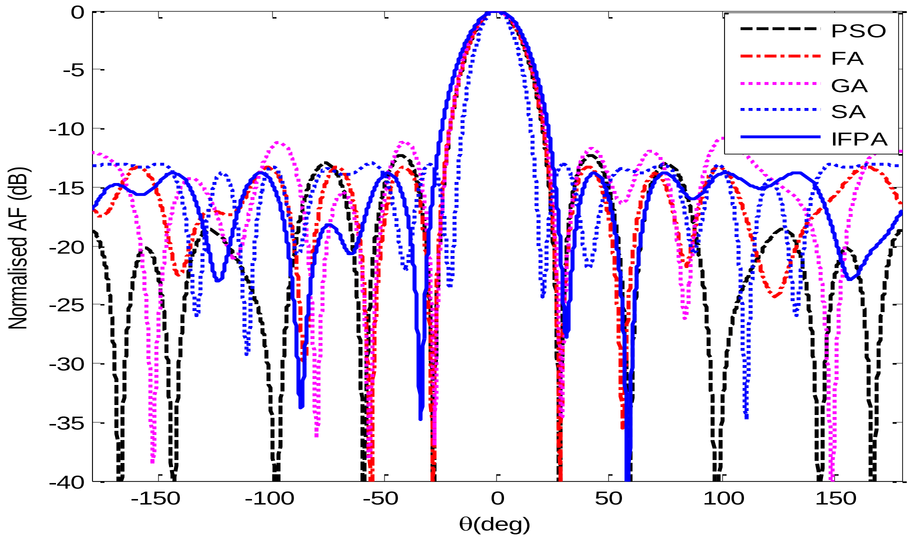



Figure 5.3 Radiation pattern obtained by non-uniform CAA for N=10 elements

For N=12, there are total 24 elements which are to be optimized, 12 elements for current excitation weights and 12 for inter-element spacing. Table 5.12 shows suppression of SLL to -14.84 dB as compared to other algorithms i.e., PSO [95] with -13.67 dB, FA [98] with -14.21 dB, GA [94] having -11.83 dB, BBO [97] having -14.37 dB, SA [96] with 13.91 dB and CSO [100] having -17.47 dB. All algorithms have higher SLL as compared to IFPA except CSO. Circumference of antenna in case of PSO, FA, GA, BBO, SA and CSO are 8.15λ, 7.21λ, 7.77λ, 10.61λ, 7.95 λ, 10.62 λ respectively as compared to IFPA which is having 7.19 λ. Therefore, it can be concluded IFPA has smaller circumference due to which it is having larger FNBW as compared with other algorithms. The radiation pattern for 12-element circular antenna array for IFPA being compared with PSO, GA, SA and IFPA is shown in Figure 5.4.

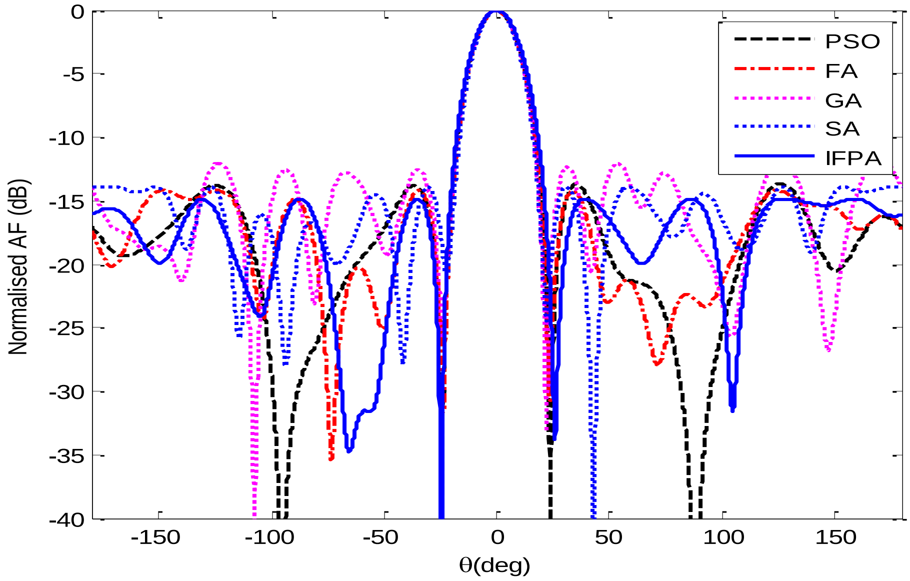



Figure 5.4 Radiation pattern obtained by non-uniform CAA for N=12 elements

| Algorithms | $[I_1, I_2, I_3, \ldots\ldots I_n]$ | | | $[d_1, d_2, d_3, \ldots\ldots d_n]$ in $\lambda$ | | | SLL(dB) | FNBW(°) |
|---|---|---|---|---|---|---|---|---|
| PSO[95] | 1.0000 | 0.7529 | 0.7519 | 0.3170 | 0.9654 | 0.3859 | -12.307 | 56 |
| | 1.0000 | 0.5062 | 1.0000 | 0.9654 | 0.3185 | 0.3164 | | |
| | 0.7501 | 0.7524 | 1.0000 | 0.9657 | 0.3862 | 0.9650 | | |
| | 0.5067 | | | 0.3174 | | | | |
| FA[98] | 0.7081 | 0.2682 | 0.3713 | 0.3810 | 0.7463 | 0.2668 | -13.30 | 56.80 |
| | 0.4100 | 0.8800 | 0.9665 | 0.3142 | 1.0000 | 0.6032 | | |
| | 0.4165 | 0.5813 | 0.7494 | 0.9706 | 0.5713 | 0.8800 | | |
| | 0.5403 | | | 0.3376 | | | | |
| GA[94] | 0.9545 | 0.4283 | 0.3392 | 0.3641 | 0.4512 | 0.2750 | -9.81 | 55.20 |
| | 0.9074 | 0.8086 | 0.4533 | 1.6373 | 0.6902 | 0.9415 | | |
| | 0.5634 | 0.6015 | 0.7045 | 0.4657 | 0.2898 | 0.6456 | | |
| | 0.9718 | | | 0.3282 | | | | |
| BBO[97] | 1.0000 | 1.0000 | 1.0000 | 0.5301 | 1.0603 | 1.3264 | -13.95 | 37.20 |
| | 0.3819 | 0.8970 | 1.0000 | 1.0000 | 0.4307 | 0.4408 | | |
| | 0.7679 | 0.8899 | 0.7246 | 1.5276 | 1.3255 | 1.0000 | | |
| | 1.0000 | | | 0.5904 | | | | |
| SA[96] | 0.6920 | 0.5679 | 0.5937 | 0.6221 | 0.9880 | 0.7777 | -13.00 | 41.40 |
| | 0.6703 | 0.9693 | 0.6014 | 0.9934 | 0.6217 | 0.9514 | | |
| | 0.3575 | 0.3020 | 0.5908 | 0.7626 | 0.5980 | 0.7655 | | |
| | 0.9718 | | | 0.9410 | | | | |
| CSO[100] | 0.3190 | 0.3443 | 0.2061 | 0.6798 | 1.0272 | 1.8979 | -14.93 | 39.6 |
| | 0.5037 | 0.4934 | 0.1444 | 1.5749 | 0.5523 | 1.1355 | | |
| | 0.2666 | 0.0448 | 0.3283 | 1.5075 | 0.8020 | 0.8498 | | |
| | 0.4781 | | | 0.6072 | | | | |
| IFPA | 0.6804 | 0.3636 | 0.3592 | 0.3702 | 0.9693 | 0.5545 | -13.81 | 62.20 |
| | 0.6314 | 0.5825 | 0.8783 | 0.5697 | 0.3274 | 0.3904 | | |
| | 0.2579 | 0.5377 | 0.7430 | 0.8216 | 0.6792 | 0.8509 | | |
| | 0.4520 | | | 0.3260 | | | | |

Table 5.11 Comparison of results obtained by IFPA with other algorithms for N=10 elements

| Algorithms | $[I_1, I_2, I_3, \ldots I_n]$ | | | $[d_1, d_2, d_3, \ldots d_n]$ in λ | | | SLL(dB) | FNBW(º) |
|---|---|---|---|---|---|---|---|---|
| PSO [95] | 0.9554 | 0.6441 | 0.7109 | 0.2569 | 0.8509 | 0.6607 | -13.67 | 48.20 |
| | 0.7769 | 1.0000 | 1.0000 | 0.7057 | 0.8540 | 0.3734 | | |
| | 0.3958 | 0.7162 | 0.6746 | 0.1609 | 0.8321 | 0.6464 | | |
| | 0.7695 | 0.9398 | 0.6145 | 0.7079 | 0.8330 | 0.2682 | | |
| FA [98] | 0.9175 | 0.3153 | 0.5814 | 0.3171 | 0.8105 | 0.5833 | -14.21 | 47.40 |
| | 0.6311 | 0.9629 | 0.9903 | 0.7609 | 0.8946 | 0.4747 | | |
| | 0.3297 | 0.4345 | 0.6820 | 0.9868 | 0.2509 | 0.2932 | | |
| | 0.4397 | 0.7151 | 0.7605 | 0.7748 | 0.6722 | 0.3955 | | |
| GA [94] | 0.2064 | 0.5416 | 0.2246 | 0.4936 | 0.4184 | 1.4474 | -11.83 | 44.60 |
| | 0.6486 | 0.7212 | 0.7913 | 0.7577 | 0.4204 | 0.5784 | | |
| | 0.5277 | 0.3495 | 0.5125 | 0.4520 | 0.8872 | 0.7514 | | |
| | 0.4475 | 0.5233 | 0.8553 | 0.4202 | 0.4223 | 0.7234 | | |
| BBO [97] | 1.0000 | 0.6501 | 0.6224 | 0.6704 | 1.0000 | 1.3046 | -14.37 | 33.40 |
| | 0.5020 | 0.5540 | 1.0000 | 0.8081 | 1.0000 | 0.4031 | | |
| | 0.6683 | 0.7234 | 0.4410 | 0.6183 | 1.1574 | 1.3465 | | |
| | 0.5123 | 0.4793 | 1.0000 | 0.6551 | 1.0000 | 0.6539 | | |
| SA [96] | 0.6231 | 0.3990 | 0.3418 | 0.8315 | 0.7910 | 0.6699 | -13.91 | 44.40 |
| | 0.6054 | 0.9444 | 0.7380 | 0.8087 | 0.7347 | 0.5331 | | |
| | 0.6741 | 0.3001 | 0.4311 | 0.4777 | 0.8960 | 0.4874 | | |
| | 0.5435 | 0.4195 | 0.9795 | 0.8657 | 0.3461 | 0.5105 | | |
| CSO [100] | 0.1261 | 0.0878 | 0.0403 | 0.5398 | 1.1051 | 0.6664 | -17.47 | 50.8 |
| | 0.0625 | 0.1155 | 0.1840 | 1.3622 | 1.1970 | 0.4420 | | |
| | 0.1534 | 0.0705 | 0.0913 | 0.4907 | 1.0990 | 1.3782 | | |
| | 0.0868 | 0.1420 | 0.2177 | 0.7836 | 1.0458 | 0.5192 | | |
| IFPA | 0.4150 | 0.4016 | 0.3016 | 0.3386 | 0.7989 | 0.6974 | -14.84 | 49.20 |
| | 0.2663 | 0.8112 | 0.6628 | 0.6218 | 0.9306 | 0.5004 | | |
| | 0.4068 | 0.1294 | 0.2390 | 0.8680 | 0.3344 | 0.4402 | | |
| | 0.2390 | 0.3252 | 0.5875 | 0.4333 | 0.8642 | 0.3654 | | |

Table 5.12 Comparison of results obtained by IFPA with other algorithms for N=12 elements

## 5.3 SIMULATION RESULTS FOR SYNTHESIS OF CONCENTRIC CIRCULAR ANTENNA ARRAY

In this section, IFPA is being applied for non-uniform concentric circular array having different number of elements. Here, the main goal for the synthesis of antenna is to determine the geometrical properties for CCCA for reducing the SLL and FNBW. This is basically done by the excitation of current for antenna array. The objection function used for the achievement of desired characteristic is same as used for CAA.

Population size=80, Maximum generations=500, $w_1$=10, $w_2$=1.

This section deals with simulation results for CCAA designs attained by IFPA. Here, three arrays of three-ring (M=3) CCAA designs are taken. These rings have maintained for non-uniform excitation and element spacing of 0.55λ, 0.61λ and 0.75λ for inner ring, middle ring and outer ring respectively. The number of elements in first ring (innermost) are denoted by N1, second ring (middle) by N2 and third (outermost) by N3.

| Algorithms | $[I_{1,1}\ I_{1,2}\ I_{1,3}\ I_{1,4}\ ;\ I_{2,1}\ I_{2,2}\ I_{2,3}\ I_{2,4}\ I_{2,5}\ I_{2,6}\ ;\ I_{3,1}\ I_{3,2}\ I_{3,3}\ I_{3,4}\ I_{3,5}\ I_{3,6}\ I_{3,7}\ I_{3,8}]$ | | | | | | SLL(dB) | FNBW(°) |
|---|---|---|---|---|---|---|---|---|
| PSO [103] | 0.4680 | 0.7271 | 0.4161 | 1.0000; | 0.5973 | 0.5422 | -27.18 | 72.24 |
| | 1.0000 | 0.4702 | 0.6336 | 0.9672; | 0.5252 | 0.8711 | | |
| | 0.6403 | 0.2811 | 0.6326 | 0.7948 | 0.5766 | 0.0000 | | |
| PSOCFIWA | 0.5796 | 0.9961 | 0.3999 | 0.9390; | 0.7212 | 0.6671 | -28.48 | 78.1 |
| [103] | 1.0000 | 0.6244 | 0.5629 | 1.0000; | 0.5725 | 1.0000 | | |
| | 0.5420 | 0.2898 | 0.5043 | 0.7267 | 0.5054 | 0.2773 | | |
| EP [103] | 0.0496 | 0.3242 | 0.0283 | 0.3416; | 0.2114 | 0.1923 | -31.84 | 78 |
| | 0.4901 | 0.1876 | 0.1994 | 0.5321; | 0.2555 | 0.3527 | | |
| | 0.2450 | 0.1229 | 0.2294 | 0.3449 | 0.2400 | 0.1204 | | |
| HEP [104] | 0.0192 | 0.4230 | 0.0233 | 0.4009; | 0.2530 | 0.2507 | -32.42 | 78.3 |
| | 0.6606 | 0.2746 | 0.2473 | 0.6098; | 0.2956 | 0.4095 | | |
| | 0.3052 | 0.1664 | 0.3213 | 0.4082 | 0.3124 | 0.1514 | | |
| FA [98] | 0.7025 | 0.1410 | 0.6770 | 0.1215; | 0.9999 | 0.4349 | -33.20 | 78.5 |
| | 0.4084 | 0.9999 | 0.4076 | 0.4305; | 0.2352 | 0.4789 | | |
| | 0.7366 | 0.4831 | 0.2542 | 0.4790 | 0.7172 | 0.4730 | | |
| CSO [100] | 0.0703 | 0.5675 | 0.0824 | 0.5491; | 0.3442 | 0.3516 | -33.39 | 76 |
| | 0.8304 | 0.3463 | 0.3440 | 0.8596; | 0.4199 | 0.5748 | | |
| | 0.4140 | 0.1919 | 0.4040 | 0.5928 | 0.4149 | 0.2097 | | |
| IFPA | 0.1585 | 0.6727 | 0.1159 | 0.6715; | 0.4133 | 0.4279 | -33.52 | 75 |
| | 0.9946 | 0.4115 | 0.3986 | 0.9955; | 0.4854 | 0.7531 | | |
| | 0.4766 | 0.2282 | 0.4813 | 0.6943 | 0.4906 | 0.2335 | | |

Table 5.13 Comparison of results obtained by IFPA with other algorithms for ring (N1=4, N2=6, N3=8)

For the first case (N1=4, N2=6 and N3=8), experimental results are shown in Table 5.13. The experimental results obtained by IFPA are compared with other standard algorithms. It can be seen that IFPA has SLL of -33.52 dB as compared to other algorithms like PSO [103], PSOCFIWA [103], EP [103], HEP [104], FA [98] and CSO [100] having -27.18 dB, -28.48 dB, -31.84 dB, -32.42 dB, -33.20 dB and -33.39 dB respectively. It is clear the IFPA has much lower SLL as being compared with other standard algorithms. It can also be seen that for IFPA, FNBW is also smaller as compared to PSOCFIWA, EP, HEP, FA and CSO while PSO has lesser FNBW than IFPA. The radiation pattern of IFPA in comparison with other algorithms are seen in Figure 5.5.

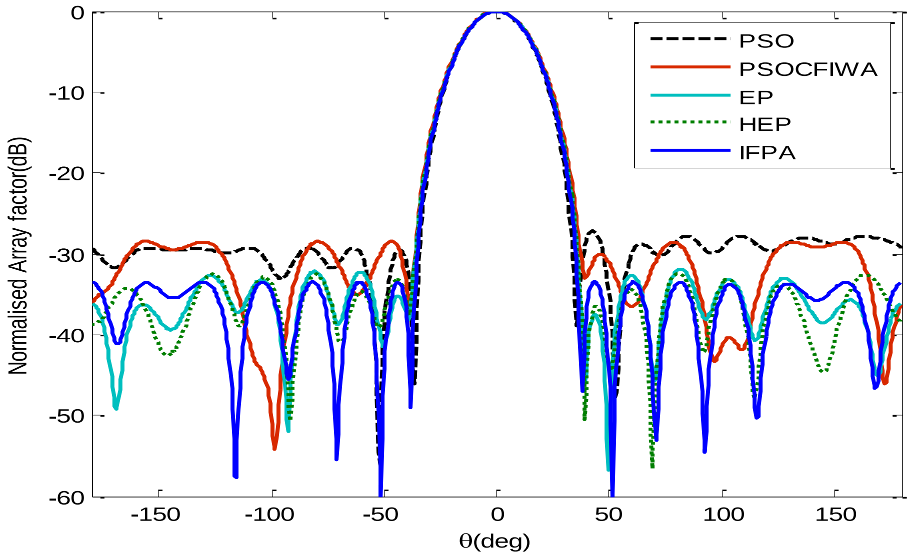



Figure 5.5 Radiation pattern for the comparative study of various algorithms for N1=4, N2=6, N3=8

| Algorithms | $[I_{1,1},\ldots\ldots,$ | $I_{1,6}; I_{2,1},\ldots\ldots..,$ | | $I_{2,8}; I_{3,1},\ldots\ldots\ldots,$ | | $I_{3,10}]$ | SLL(dB) | FNBW(º) |
|---|---|---|---|---|---|---|---|---|
| PSO [103] | 0.4633 | 0.7867 | 1.0000 | 0.5946 | 1.0000 | 0.8731; | -22.76 | 57.15 |
| | 0.7502 | 0.0000 | 0.8257 | 0.5810 | 1.0000 | 0.0188 | | |
| | 0.8705 | 0.9735; | 0.3365 | 0.7593 | 0.7550 | 0.3788 | | |
| | 0.3507 | 0.4310 | 0.9318 | 1.0000 | 0.0420 | 0.0726 | | |
| PSOCFIWA | 0.9370 | 0.5491 | 1.0000 | 0.5833 | 0.8306 | 1.0000; | -23.47 | 62 |
| [103] | 1.0000 | 0.0229 | 0.9453 | 1.0000 | 1.0000 | 0.0022 | | |
| | 1.0000 | 1.0000; | 0.1549 | 0.8472 | 0.6792 | 0.4331 | | |
| | 0.5513 | 0.6437 | 0.6486 | 1.0000 | 0.5610 | 0.5478 | | |
| EP [103] | 0.3201 | 0.1077 | 0.1665 | 0.1586 | 0.2355 | 0.4739; | -25.16 | 57 |
| | 0.4548 | 0.0000 | 0.2516 | 0.4185 | 0.1299 | 0.0007 | | |
| | 0.5280 | 0.5362; | 0.1891 | 0.4027 | 0.3443 | 0.1948 | | |
| | 0.3146 | 0.2293 | 0.2476 | 0.4621 | 0.2266 | 0.1944 | | |
| HEP [104] | 0.2539 | 0.1643 | 0.2535 | 0.1615 | 0.1837 | 0.4012; | -25.96 | 59.3 |
| | 0.4195 | 0.0000 | 0.3601 | 0.5418 | 0.3092 | 0.0000 | | |
| | 0.4333 | 0.5739; | 0.1976 | 0.4085 | 0.4174 | 0.2424 | | |
| | 0.3586 | 0.2639 | 0.3406 | 0.4188 | 0.2275 | 0.2457 | | |
| IFPA | 0.2663 | 0.3291 | 0.4104 | 0.3446 | 0.3178 | 0.5992; | -26.35 | 60 |
| | 0.6223 | 0.0000 | 0.5954 | 0.8589 | 0.5438 | 0.0000 | | |
| | 0.5344 | 0.8050; | 0.4653 | 0.5380 | 0.5959 | 0.2980 | | |
| | 0.4044 | 0.2847 | 0.6050 | 0.5157 | 0.4354 | 0.5158 | | |

Table 5.14 Comparison of results obtained by IFPA with other algorithms for ring (N1=6, N2=8, N3=10)

For the second case, N1=6, N2=8 and N3=10, experimental results of IFPA compared with other algorithms are shown in Table 5.14. It can be observed that IFPA has SLL as -26.35 dB when being compared with PSO [103], PSOCFIWA [103], EP [103] and HEP [104] as -22.76 dB, -23.47 dB, -25.16 dB and 25.96 dB respectively. It is clear that IFPA is better in reducing SLL as compared to other algorithms. FNBW obtained by IFPA is also much closer to other algorithms. The radiation pattern of IFPA in comparison with other standard algorithms are shown in Figure 5.6.

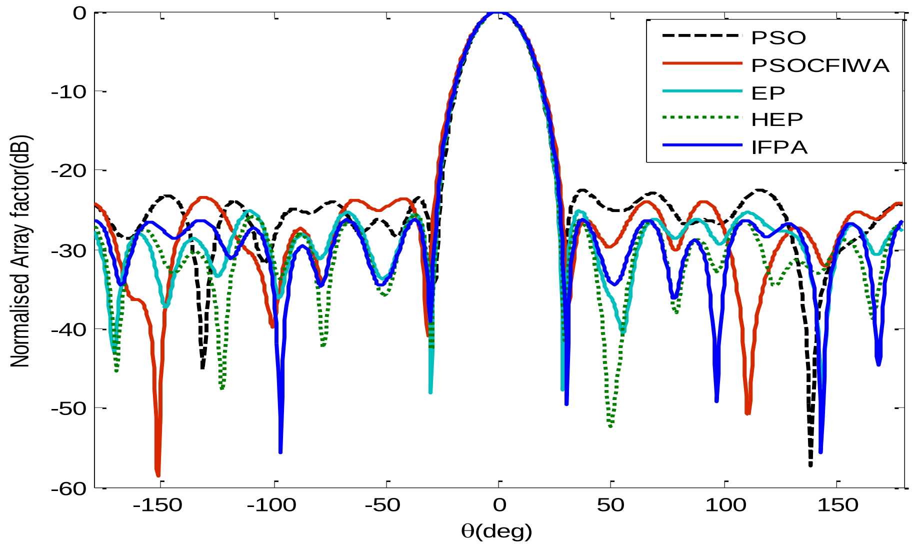



Figure 5.6 Radiation pattern for the comparative study of various algorithms for N1=6, N2=8, N3=10 for CCCA

For the last case (N1=8, N2=10, N3=12), simulation results of IFPA are compared with other algorithms as shown in Table 5.15. IFPA has obtained SLL as -27.69 dB when compared with other algorithms such as PSO [103] with -23.06 dB , PSOCFIWA [103] with -25.8, EP [103] having -26.12, HEP [104] having -27.30, FA [98] with -27.49 and CSO [100] having -27.86 dB. IFPA has also lower SLL as compared with other algorithms expect CSO, so it can be said that IFPA is better in reducing SLL. IFPA has also lower FNBW as compared to other algorithms except PSO and CSO. For the case of PSO and CSO, IFPA is comparable in terms of FNBW. The radiation pattern of IFPA in comparison with other standard algorithms are shown in Figure 5.7.

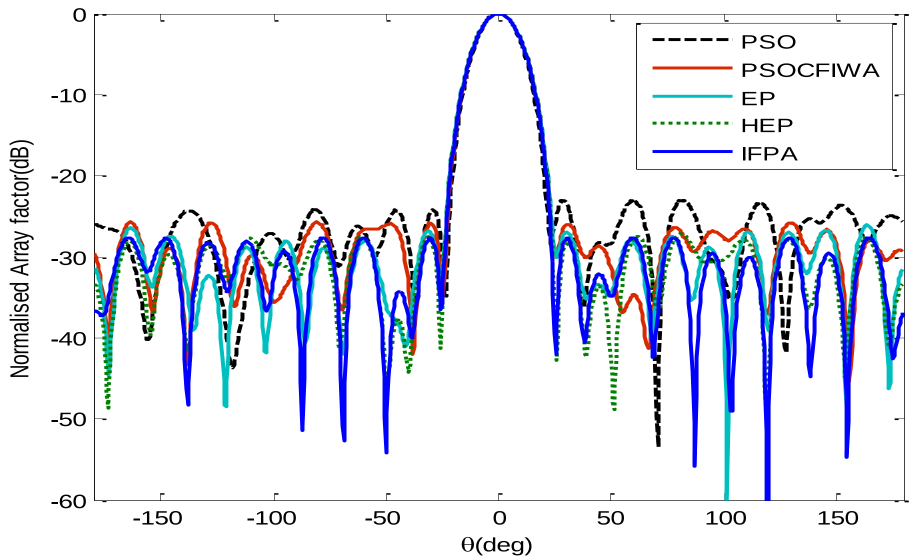



Figure 5.7 Radiation pattern for the comparative study of various algorithms for N1=8, N2=10, N3=12 for CCCA

| Algorithms | $[I_{1,1},\ldots, I_{1,8}; I_{2,1},\ldots, I_{2,10}; I_{3,1},\ldots, I_{3,12}]$ | | | | | | SLL(dB) | FNBW(°) |
|---|---|---|---|---|---|---|---|---|
| PSO [103] | 0.5817 | 0.6087 | 0.7286 | 0.7449 | 0.8850 | 0.4086 | -23.06 | 47.07 |
| | 1.0000 | 1.0000; | 0.8089 | 0.0000 | 0.0000 | 1.0000 | | |
| | 0.5783 | 1.0000 | 0.0000 | 0.0000 | 1.0000 | 0.6361; | | |
| | 0.6352 | 0.1718 | 1.0000 | 0.8064 | 0.1952 | 0.2029 | | |
| | 0.0000 | 1.0000 | 1.0000 | 0.6430 | 0.3959 | 0.3762 | | |
| PSOCFIWA | 0.8361 | 0.1544 | 1.0000 | 1.0000 | 0.6610 | 0.4733 | -25.8 | 50.4 |
| [103] | 0.6552 | 0.6207; | 0.5352 | 0.0061 | 0.0902 | 0.8490 | | |
| | 0.6214 | 0.7821 | 0.2940 | 0.2169 | 0.2752 | 0.2857; | | |
| | 0.4164 | 0.4339 | 1.0000 | 0.3989 | 0.3714 | 0.4873 | | |
| | 0.4445 | 0.2622 | 0.9938 | 0.1467 | 0.3797 | 0.4164 | | |
| EP [103] | 0.2886 | 0.1891 | 0.3336 | 0.5458 | 0.3895 | 0.1000 | -26.12 | 51 |
| | 0.2866 | 0.2242; | 0.1378 | 0.1036 | 0.1000 | 0.4048 | | |
| | 0.2686 | 0.3090 | 0.1000 | 0.1000 | 0.1696 | 0.1595; | | |
| | 0.1183 | 0.1144 | 0.4708 | 0.1685 | 0.2090 | 0.2566 | | |
| | 0.2200 | 0.1000 | 0.4229 | 0.1273 | 0.1020 | 0.2419 | | |
| HEP [104] | 0.3356 | 0.1697 | 0.2629 | 0.2719 | 0.3039 | 0.1342 | -27.30 | 51.3 |
| | 0.3562 | 0.4342; | 0.3223 | 0.0657 | 0.0444 | 0.1575 | | |
| | 0.1418 | 0.1738 | 0.0445 | 0.0721 | 0.3286 | 0.2282; | | |
| | 0.1509 | 0.1533 | 0.3812 | 0.0685 | 0.1776 | 0.1759 | | |
| | 0.1427 | 0.1064 | 0.3736 | 0.1897 | 0.1265 | 0.1526 | | |
| FA [98] | 0.9354 | 0.7716 | 0.3013 | 0.7299 | 0.8924 | 0.7641 | -27.49 | 49.3 |
| | 0.3044 | 0.7999; | 0.5444 | 0.5686 | 0.2124 | 0.1958 | | |
| | 0.5901 | 0.5647 | 0.6322 | 0.1498 | 0.1660 | 0.6379; | | |
| | 0.5044 | 0.4125 | 0.2457 | 0.9673 | 0.2516 | 0.3827 | | |
| | 0.4854 | 0.3444 | 0.3209 | 0.9734 | 0.3290 | 0.365 | | |
| CSO [100] | 0.6739 | 0.3432 | 0.6771 | 0.8039 | 0.7780 | 0.2618 | -27.86 | 49 |
| | 0.7627 | 0.7271; | 0.4706 | 0.1789 | 0.1448 | 0.5577 | | |
| | 0.4837 | 0.6122 | 0.0510 | 0.0895 | 0.6417 | 0.4643; | | |
| | 0.3747 | 0.2329 | 0.9529 | 0.2425 | 0.4582 | 0.4318 | | |
| | 0.3656 | 0.3845 | 0.9701 | 0.4396 | 0.2581 | 0.4045 | | |
| IFPA | 0.7178 | 0.2602 | 0.7148 | 0.7823 | 0.5625 | 0.3581 | -27.69 | 50 |
| | 0.5548 | 0.6962; | 0.5313 | 0.1564 | 0.0662 | 0.6927 | | |
| | 0.4869 | 0.6197 | 0.0956 | 0.1884 | 0.4958 | 0.4773; | | |
| | 0.3204 | 0.3170 | 0.9028 | 0.4556 | 0.3265 | 0.4819 | | |
| | 0.3706 | 0.3367 | 0.8969 | 0.2151 | 0.3644 | 0.4976 | | |

Table 5.15 Comparison of results obtained by IFPA with other algorithms for ring (N1=8, N2=10, N3=12)

# CHAPTER 6
# CONCLUSIONS AND FUTURE SCOPE

## 6.1 CONCLUSION

Nature inspired algorithms are becoming popular in present world due to their ability to solve real-word problems. The most popular and older nature inspired algorithm i.e., evolutionary algorithms have been applied to many practical problems and proven their worth. One of the most famous evolutionary algorithm namely, FPA has been discussed in proposed work. Since, FPA has some disadvantages due to which this algorithm is lacking for its applicability in practical applications. Hence, in this present study, work has been done to overcome its drawbacks by the introduction of modified FPA. This enhanced version of FPA is named as improved flower pollination algorithm (IFPA). Three modifications are proposed for IFPA which make it competitive with other standard algorithms. Firstly, Levy flight is replaced by Cauchy operator for the improvement of exploration. Secondly, effect of current best is considered in local search phase. Lastly, switching probability is improved by exponential decreasing function in order to balance global and local search effectively. The performance of IFPA can be evaluated by testing it with different benchmark functions. IFPA is also compared with other standard algorithms and tested for different population as well as dimension sizes. IFPA has even tested for p-test values that shows the effectiveness of proposed approach.

In order to check the efficiency of IFPA, it has been applied for the synthesis of antenna array. In antenna design, IFPA has been tested for two real world problems namely, for the synthesis of non-uniform CAA and CCAA. The positions and amplitudes of antenna array are adjusted in case of CAA and compared the results with PSO, FA, GA, BBO, SA and CSO. On the other hand, amplitude parameter is controlled in case of CCAA and compared its results with PSO, PSOCFIWA, EP, HEP, FA and CSO. These results demonstrate that IFPA outperforms all algorithms in reducing SLL. Finally, it can be concluded that IFPA is much better than already existed approaches including FPA and even proven its superiority by applying with real world problems.

## 6.2 FUTURE SCOPE

The knowledge gained after working on evolutionary algorithm help us to propose the following future research work:

- FPA can further be modified by replacing other heavy tailed distribution such as gamma distributions, normal distribution or others.

- The switching probability method can also be removed by combining both global search equation and local search equation.
- An enhanced versions of IFPA can be derived by modifying the global search part.
- IFPA can also be hybridized with different other algorithms.
- Further, binary and multi-objective versions of IFPA can be designed as the possible future prospects.
- IFPA can also be applied to other practical applications like robotics, travel salesman problem, vehicle routing problems and so on.
- In the field of communication, IFPA can be extended for Bluetooth, Wi-Fi and so on.
- In signal processing, IFPA can be applied designing of IIR and FIR filters and the work could also be extended to image processing.
- In the field of medical science, work can be extended for various practical problems like cancer treatment, tuberculosis, DNA prediction, etc.

Hence, at last it can be said that IFPA can be applied in different fields like image processing, electronics, wireless communication, antenna design, medical science and so on. Thus, it can be stated the proposed work has proven its worth in antenna designing and by applying further enhancement techniques, work can be extended to most of the practical problems.

# REFERENCES


[1] Deep K and Thakur M (2007). A new mutation operator for real coded genetic algorithms, *Applied mathematics and Computation*, 193(1), 211-230.

[2] Bansal JC *et al.* (2014). Spider monkey optimization algorithm for numerical optimization, *Memetic computing*, 6(1), 31-47.

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

[4] Geem ZW, Kim JH and Loganathan GV (2001). A new heuristic optimization algorithm: harmony search, *Simulation*, 76(2), 60-68.

[5] Tan KC *et al.* (2009). Balancing exploration and exploitation with adaptive variation for evolutionary multi-objective optimization, *European Journal of Operational Research*, 197(2), 701-713.

[6] Yang XS (2011). Review of meta-heuristics and generalized evolutionary walk algorithm, *International Journal of Bio-Inspired Computation*, 3(2), 77-84.

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

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

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

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

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

[12] Beyer HG and Schwefel HP (2002). Evolution strategies–A comprehensive introduction, *Natural computing*, 1(1), 3-52.

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

[14] Kaveh A and Khayatazad M (2012). A new meta-heuristic method: ray optimization, *Computers & structures*, 112, 283-294.

[15] Kaveh A and Talatahari S (2010). A novel heuristic optimization method: charged system search, *Acta Mechanica*, 213(3), 267-289.

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

[17] Erol OK and Eksin I (2006). A new optimization method: big bang-big crunch, *Advances in Engineering software*, 37(2), 106-111.

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

[19] Yang XS (2005). Engineering optimizations via nature-inspired virtual bee algorithms, *Artificial intelligence and knowledge engineering applications: a bioinspired approach*, *IWINAC* [Berlin, Heidelberg: 2005], pp. 317-323.

[20] D. Karaboga, An idea based on honey bee swarm for numerical optimization. Technical Report-TR06: Erciyes University, Engineering Faculty, Computer Engineering Department, Kayseri/Turkiye, 2005.

[21] Oster GF and Wilson EO. *Caste and ecology in the social insects*. Princeton University Press, 1978.

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

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

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

[25] Karaboga D and Basturk B (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, *Journal of global optimization*, 39(3), 459-471.

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

[27] Yang XS (2009). Firefly algorithms for multimodal optimization, *International symposium on stochastic algorithms* [Springer, Berlin: 2009], pp. 169-178

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

[29] 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.

[30] Balanis CA. *Antenna theory: analysis and design*, John Wiley & Sons, 2012.

[31] Beni G and Wang J (1993). Swarm intelligence in cellular robotic systems, *International conference on Robots and Biological Systems: Towards a New Bionics?* [Berlin, Heidelberg: 2009], pp. 703-712.

[32] Liu J, Xu W and Sun J (2005). Quantum-behaved particle swarm optimization with mutation operator, *International Conference on Tools with Artificial Intelligence* [17$^{th}$: Hong Kong, China: 2005], pp. 4-pp.

[33] Gao XY, Sun LQ and Sun DS (2009). An enhanced particle swarm optimization algorithm, *Information Technology Journal*, 8, 1263-1268.

[34] Jensi R and Jiji GW (2016). An enhanced particle swarm optimization with levy flight for global optimization, *Applied Soft Computing*, 43, 248-261.

[35] Bhambu P, Kumar S and Sharma K (2017). Self balanced particle swarm optimization, *International Journal of System Assurance Engineering and Management*, 1-10.

[36] Wang H *et al*. (2013). Diversity enhanced particle swarm optimization with neighborhood search, *Information Sciences*, 223, 119-135.

[37] Arumugam MS, Rao MVC and Tan AW (2009). A novel and effective particle swarm optimization like algorithm with extrapolation technique, *Applied soft computing*, 9(1), 308-320.

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

[39] Yu S *et al.* (2015). Enhancing firefly algorithm using generalized opposition-based learning, *Computing,* 97(7), 741-754.

[40] Wang H *et al*. (2016). Randomly attracted firefly algorithm with neighborhood search and dynamic parameter adjustment mechanism, *Soft Computing*, 1-15.

[41] Gupta A and Padhy PK (2016). Modified Firefly Algorithm based controller design for integrating and unstable delay processes, *Engineering Science and Technology, an International Journal*, 19(1), 548-558.

[42] Meena S and Chitra K (2015). Modified approach of firefly algorithm for non-minimum phase systems, *Indian Journal of Science and Technology*, 8(23).

[43] Ahmadi AH, and Nikravesh SKY (2016). A novel instantaneous exploitation based bat algorithm, *Iranian Conference on Electrical Engineering* [24$^{th}$: Shiraz, Iran: 2016], pp. 1751-1756.

[44] Zhu B *et al.* (2016). A Novel Quantum-Behaved Bat Algorithm with Mean Best Position Directed for Numerical Optimization, *Computational intelligence and neuroscience*, 2016.

[45] Chakri A *et al*. (2017). New directional bat algorithm for continuous optimization problems, *Expert Systems with Applications*, 69, 159-175.

[46] Yılmaz S, Kucuksille EU and Cengiz Y (2014). Modified bat algorithm, *Elektronika ir Elektrotechnika*, 20(2), 71-78.

[47] Xie J, Zhou Y and Chen H (2013). A novel bat algorithm based on differential operator and Lévy flights trajectory, *Computational intelligence and neuroscience*, 2013.

[48] Kielkowicz K and Grela D (2016). Modified Bat Algorithm for Nonlinear Optimization, *International Journal of Computer Science and Network Security (IJCSNS),* 16(10), 46.

[49] Cai X *et al.* (2014). Bat algorithm with Gaussian walk, *International Journal of Bio-Inspired Computation*, 6(3), 166-174.

[50] Afrabandpey H *et al*. (2014). A novel bat algorithm based on chaos for optimization tasks, *Iranian Conference on Intelligent Systems* [Bam, Iran: 2014], pp. 1-6.

[51] Gholizadeh S and Aligholizadeh V (2013). Optimum design of reinforced concrete frames using bat meta-heuristic algorithm, *International Journal of Optimization in Civil Engineering*, 3(3), 483-97.

[52] Kaveh A and Zakian P (2014). Enhanced bat algorithm for optimal design of skeletal structures, *Asian J Civial Eng*, 15(2), 179-212.

[53] Goyal S and Patterh MS (2016). Modified bat algorithm for localization of wireless sensor network, *Wireless Personal Communications*, 86(2), 657-670.

[54] Saha S *at al.* (2013). A new design method using opposition-based BAT algorithm for IIR system identification problem, *International Journal of Bio-Inspired Computation*, 5(2), 99-132.

[55] Latif A and Palensky P (2014). Economic dispatch using modified bat algorithm, *Algorithms*, 7(3), 328-338.

[56] dos Santos Coelho L and Askarzadeh A (2016). An enhanced bat algorithm approach for reducing electrical power consumption of air conditioning systems based on differential operator, *Applied Thermal Engineering*, 99, 834-840.

[57] Wang G and Guo L (2013). A novel hybrid bat algorithm with harmony search for global numerical optimization, *Journal of Applied Mathematics*, 2013.

[58] Fister Jr I, Fister D and Yang XS (2013). A hybrid bat algorithm, *Electrotechnical review*, 80, 1-7.

[59] Fister I *et al.* (2014). A novel hybrid self-adaptive bat algorithm, *The Scientific World Journal*.

[60] Pan TS, Dao TK and Chu SC (2015). Hybrid particle swarm optimization with bat algorithm. *International Conference on Genetic and evolutionary computing* [8th: Nangchang, China: 2015], pp. 37-47.

[61] Pan JS *et al.* (2014). Hybrid bat algorithm with artificial bee colony, *Euro-China Conference on Intelligent Data analysis and its Applications* [1st: Shenzhen, China: 2014], pp. 45-55.

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

[63] Kaveh A and Ghazaan MI (2016). Enhanced Whale Optimization Algorithm for Sizing Optimization of Skeletal Structures, *Mechanics Based Design of Structures and Machines*, 1-18.

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

[65] Horng MF et al. (2017). A Multi−Objective Optimal Vehicle Fuel Consumption Based on Whale Optimization Algorithm, *International Conference on Intelligent Information Hiding and Multimedia Signal Processing* [12th: Kaohsiung, Taiwan: 2017], pp. 371−380.

[66] Dao TK, Pan TS and Pan JS (2016). A multi−objective optimal mobile robot path planning based on whale optimization algorithm, *IEEE International Conference on Signal Processing* [13th: Chengdu, China: 2016], pp. 337−342.

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

[68] Alatas B (2011). Uniform big bang–chaotic big crunch optimization, *Communications in Nonlinear Science and Numerical Simulation*, 16(9), 3696-3703.

[69] Kaveh A and Abbasgholiha H (2011). Optimum design of steel sway frames using Big Bang-Big Crunch algorithm, *Asian J Civ Eng*, 12(3), 293-317.

[70] Kaveh A and Sabzi O (2012). Optimal design of reinforced concrete frames using big bang-big crunch algorithm, *International journal of civil engineering*, 10(3), 189-200.

[71] Sarafrazi S, Nezamabadi-Pour H and Saryazdi S (2011). Disruption: a new operator in gravitational search algorithm, *Scientia Iranica*, 18(3), 539-548.

[72] Rashedi E, Nezamabadi-Pour H and Saryazdi S (2011). Filter modeling using gravitational search algorithm, *Engineering Applications of Artificial Intelligence*, 24(1), 117-122.

[73] Duman S *at al*. (2012). Optimal power flow using gravitational search algorithm, *Energy Conversion and Management*, 59, 86-95.

[74] Reeves CR (1995). A genetic algorithm for flowshop sequencing, *Computers & operations research*, 22(1), 5-13.

[75] Chu PC and Beasley JE (1997). A genetic algorithm for the generalised assignment problem, *Computers & Operations Research*, 24(1), 17-23.

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

[77] Hou ES, Ansari N and Ren H (1994). A genetic algorithm for multiprocessor scheduling, *IEEE Transactions on Parallel and Distributed systems*, 5(2), 113-120.

[78] Wang R and Zhou Y (2014). Flower pollination algorithm with dimension by dimension improvement, *Mathematical Problems in Engineering*, 2014.

[79] Nabil E (2016). A modified flower pollination algorithm for global optimization, *Expert Systems with Applications*, 57, 192-203.

[80] Dubey HM, Pandit M and Panigrahi BK (2015). A biologically inspired modified flower pollination algorithm for solving economic dispatch problems in modern power systems, *Cognitive Computation*, 7(5), 594-608.

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

[82] Regalado JA, Emilio BE and Cuevas E (2015). Optimal power flow solution using modified flower pollination algorithm, *International Autumn Meeting on Power, Electronics and Computing* [Ixtapa, Mexico: 2015], pp. 1-6.

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

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

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

[86] Trivedi IN, Purani SV and Jangir PK (2015). Optimized over-current relay coordination using Flower Pollination Algorithm, *IEEE International Advance Computing Conference* [Bangalore, India: 2015], pp. 72-77.

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

[88] Pambudy MMM, Hadi SP and Ali HR (2014). Flower pollination algorithm for optimal control in multi-machine system with GUPFC, IEEE *International Conference on Information Technology and Electrical Engineering (ICITEE)* [6th: Yogyakarta, Malaysia: 2014], pp.1-6.

[89] Abdel-Baset M and Hezam IM (2015). An Effective Hybrid Flower Pollination and Genetic Algorithm for Constrained Optimization Problems, *Advanced Engineering Technology and Application An International Journal*, 4, 27-27.

[90] Abdel-Raouf O and Abdel-Baset M (2014). A new hybrid flower pollination algorithm for solving constrained global optimization problems, *International Journal of Applied Operational Research-An Open Access Journal*, 4(2), 1-13.

[91] Abdel-Baset M and Hezam I (2016). A Hybrid Flower Pollination Algorithm for Engineering Optimization Problems, *International Journal of Computer Applications*, 140(12).

[92] Chakraborty D, Saha S and Dutta O (2014). DE-FPA: A hybrid differential evolution-flower pollination algorithm for function minimization, IEEE *International Conference on High Performance Computing and Applications* [Bhubaneswar, India: 2014], pp. 1-6.

[93] Abdel-Raouf O, El-Henawy I and Abdel-Baset M (2014). A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles, *International Journal of Modern Education and Computer Science*, 6(3), 38.

[94] Panduro MA *et al.* (2006). Design of non-uniform circular antenna arrays for side lobe reduction using the method of genetic algorithms, *AEU-International Journal of Electronics and Communications*, 60(10), 713-717.

[95] Najjar MSY and Khodier NDM (2008). Design of non–uniform circular antenna arrays using particle swarm optimization, *Journal of electrical engineering*, 59(4), 216-220.

[96] Rattan M, Patterh MS and Sohi BS (2009). Optimization of circular antenna arrays of isotropic radiators using simulated annealing, *International Journal of Microwave and Wireless Technologies*, 1(5), 441.

[97] Singh U and Kamal TS (2011). Design of non-uniform circular antenna arrays using biogeography-based optimization, *IET microwaves, antennas & propagation*, 5(11), 1365-1370.

[98] 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.

[99] Ram G *et al. (*2013). Design of circular antenna arrays for the reduction of side lobe and first null beamwidth using BFO, *International Conference on Microwave and Photonics* [Dhanbad, India: 2013] (pp. 1-6).

[100] Ram G *et al*. (2015). Circular and concentric circular antenna array synthesis using cat swarm optimization, *IETE Technical Review*, 32(3), 204-217.

[101] Das S *et al*. (2010). Optimal angular locations of elements for asymmetric circular array antennas, *International Symposium on Industrial Electronics & Applications* [Penang, Malaysia: 2010], pp. 681-685.

[102] Mandal D, Ghoshal SP and Bhattacharjee AK (2010). Optimal design of concentric circular antenna array using particle swarm optimization with constriction factor approach, *International Journal of Computer Applications*, 1(17), 112-116.

[103] Mandal D, Ghoshal SP and Bhattacharjee AK (2010). Design of concentric circular antenna array with central element feeding using particle swarm optimization with constriction factor and inertia weight approach and evolutionary programing technique, *Journal of Infrared, Millimeter, and Terahertz Waves*, 31(6), 667-680.

[104] Mandal D, Ghoshal SP and Bhattacharjee AK (2010). Optimal synthesis of array pattern for concetric circular antenna array using hybrid evolutionary programming, *International Journal of Research Trends in Engineering and Technology*, 3(3).

[105] Mandal D, Bhattacharjee AK and Ghoshal SP (2009). Application of bio-inspired optimization technique for finding the optimal set of concentric circular antenna array, *World Congress on Nature & Biologically Inspired Computing* [Coimbatore, India: 2009], pp. 1247-1252.

[106] Pal S *et al.* (2010). Concentric circular antenna array synthesis using a differential invasive weed optimization algorithm, *International Conference of Soft Computing and Pattern Recognition* [Paris, France: 2010], pp. 395-400.

[107] Glover B. *Understanding flowers and flowering: An Integrated Appraoch*, Oxford University Press, 2007.

[108] Pavlyukevich I (2007). Lévy flights, non-local search and simulated annealing, *Journal of Computational Physics*, 226(2), 1830-1844.

[109] Chen G *et al.* (2006). Natural exponential inertia weight strategy in particle swarm optimization, *World Congress on Intelligent Control and Automation*, [6th: Dalian, China: 2006], pp. 3672-3675.

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

[111] Derrac J *et al. (*2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, *Swarm and Evolutionary Computation*, 1(1), 3-18.

# LIST OF PUBLICATIONS

[1] Singh D, Salgotra R and Singh U (2017). A novel modified bat algorithm for global optimization, *IEEE International conference on the innovation, embedded and communication systems* [4$^{th}$: Coimbatore, India: 2017]. (Status-Accepted)

[2] Singh D, Singh U and Salgotra S (2017). An extended version of flower pollination algorithm, *In Arabian Journal of Science and Enginnering*. (Status-in communication)

Deepika_14.7.17

ORIGINALITY REPORT

| %5 | %2 | %3 | %2 |
|---|---|---|---|
| SIMILARITY INDEX | INTERNET SOURCES | PUBLICATIONS | STUDENT PAPERS |

PRIMARY SOURCES

| | | |
|---|---|---|
| 1 | Ram, Gopi, Durbadal Mandal, Rajib Kar, and Sakti Prasad Ghoshal. "Circular and Concentric Circular Antenna Array Synthesis Using Cat Swarm Optimization", IETE Technical Review, 2015.<br>Publication | %1 |
| 2 | www.deq.state.ut.us<br>Internet Source | <%1 |
| 3 | Submitted to ABV-Indian Institute of Information Technology and Management Gwalior<br>Student Paper | <%1 |
| 4 | Ram, Gopi, Durbadal Mandal, Rajib Kar, and Sakti Prasad Ghoshal. "Optimal design of non-uniform circular antenna arrays using PSO with wavelet mutation", International Journal of Bio-Inspired Computation, 2014.<br>Publication | <%1 |
| 5 | www.ijcaonline.org<br>Internet Source | <%1 |

| | | |
|---|---|---|
| 6 | www.just.edu.jo<br>Internet Source | <%1 |
| 7 | Submitted to Higher Education Commission Pakistan<br>Student Paper | <%1 |
| 8 | Submitted to University of Witwatersrand<br>Student Paper | <%1 |
| 9 | mcf.gsfc.nasa.gov<br>Internet Source | <%1 |
| 10 | Caughlan, G.R.. "Tables of thermonuclear reaction rates for low-mass nuclei (1 =< Z =< 14)", Atomic Data and Nuclear Data Tables, 198503<br>Publication | <%1 |
| 11 | Submitted to University of Duhok<br>Student Paper | <%1 |
| 12 | Submitted to Savitribai Phule Pune University<br>Student Paper | <%1 |
| 13 | visitor.ics.uci.edu<br>Internet Source | <%1 |
| 14 | Ram, Gopi, Durbadal Mandal, Rajib Kar, and Sakti Prasad Ghoshal. "Hybrid GSADE algorithm for optimization of far-field radiation pattern of circular arrays", annals of telecommunications - annales des | <%1 |

télécommunications, 2015.
Publication

15 Ram, Gopi, Durbadal Mandal, Rajib Kar, and Sakti Prasad Ghoshal. "Opposition-based BAT algorithm for optimal design of circular and concentric circular arrays with improved far-field radiation characteristics : Optimal Design of Circular and Concentric Circular", International Journal of Numerical Modelling Electronic Networks Devices and Fields, 2015.
Publication
<%1

16 Singh, U., and T.S. Kamal. "Design of non-uniform circular antenna arrays using biogeography-based optimisation", IET Microwaves Antennas & Propagation, 2011.
Publication
<%1

17 www.minas.upm.es
Internet Source
<%1

18 Durbadal Mandal. "Design of Concentric Circular Antenna Array with Central Element Feeding Using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach and Evolutionary Programing Technique", Journal of Infrared Millimeter and Terahertz Waves, 03/20/2010
Publication
<%1

19 Shaw, B.. "A novel opposition-based

gravitational search algorithm for combined economic and emission dispatch problems of power systems", International Journal of Electrical Power and Energy Systems, 201202 <%1
Publication

20 Submitted to CONACYT <%1
Student Paper

21 Wu, Huaning Liu, Chao Xie, Xu. "Pattern synthesis of planar nonuniform circular antenna arrays using a chaotic adaptive invasive wee", Mathematical Problems in Engineering, Annual 2014 Issue <%1
Publication

22 Submitted to TechKnowledge Turkey <%1
Student Paper

23 Submitted to Universiti Kebangsaan Malaysia <%1
Student Paper

24 Submitted to University of Mauritius <%1
Student Paper

25 Submitted to University Tun Hussein Onn Malaysia <%1
Student Paper

26 www.docstoc.com <%1
Internet Source

27 Sharaqa, Ashraf, and Nihad Dib. "Circular <%1

antenna array synthesis using firefly algorithm", International Journal of RF and Microwave Computer-Aided Engineering, 2013.
Publication

28 Zhang, Song Liu, Sanyang. "A novel artificial bee colony algorithm for function optimization. (Research Article)(Report)", Mathematical Problems in Engineering, Annual 2015 Issue
Publication
<1%

29 Submitted to IIT Delhi
Student Paper
<1%

30 Adel Younis. "Trends, features, and tests of common and recently introduced global optimization methods", Engineering Optimization, 2010
Publication
<1%

31 www.skads-eu.org
Internet Source
<1%

32 www-nds.iaea.or.at
Internet Source
<1%

33 Submitted to Universiti Tenaga Nasional
Student Paper
<1%

34 Submitted to University of Northumbria at Newcastle
Student Paper
<1%

Submitted to City University of Hong Kong

| | | |
|---|---|---|
| 35 | Student Paper | <%1 |
| 36 | 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<br>Publication | <%1 |
| 37 | Wu, Huaning Liu, Chao Xie, Xu. "Thinning of concentric circular antenna arrays using improved binary invasive weed optimization algo", Mathematical Problems in Engineering, Annual 2015 Issue<br>Publication | <%1 |
| 38 | glucagon.com<br>Internet Source | <%1 |
| 39 | archive.epa.gov<br>Internet Source | <%1 |
| 40 | uvb.nrel.colostate.edu<br>Internet Source | <%1 |
| 41 | www.scholarpedia.org<br>Internet Source | <%1 |

EXCLUDE QUOTES ON

EXCLUDE MATCHES < 10 WORDS

EXCLUDE BIBLIOGRAPHY ON