Reduction of Side Lobe level of Linear Antenna Array Using Optimization Algorithms

A thesis submitted
in partial fulfilment of the requirements for the degree of

MASTER OF TECHNOLOGY
in
Electronics and Communication Engineering

Submitted By
AL-HUSSEIN MOHAMMED JUMAAH
REG. NO. 7141-2021-266

Supervisor	Co-Supervisor
Dr. Sonia Goyal	Dr. Amrit Kaur

Department of Electronics and Communication Engineering,
Punjabi University, Patiala-147002
July 2023

Candidate's Declaration

I hereby certify that the work which is being presented in the thesis entitled "REDUCTION OF SIDE LOBE LEVEL OF LINEAR ANTENNA ARRAY USING OPTIMIZATION ALGORITHMS" by AL-HUSSEIN MOHAMMED JUMAAH in fulfillment of requirement for the award of degree of M. Tech. (Electronics and Communication Engineering) submitted to the department of Electronics and Communication Engineering, Punjabi University, Patiala, is an authentic record of my own work carried out during a period from January to June 2023 under the supervision of Dr. SONIA GOYAL and Dr. AMRIT KAUR, Department of Electronics and Communication Engineering, Punjabi University, Patiala. The matter presented in this thesis has not been submitted in any other University/Institute for the award of any degree.

AL-HUSSEIN MOHAMMED JUMAAH ROLL NO: 12192009

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

Amrit Kaur

Signature of Supervisor Signature of Co-Supervisor

The M.Tech. viva-voce examination of "AL-HUSSEIN MOHAMMED JUMAAH" has been held on 30-06-2023

Signature of Supervisor (s) Signature of External Examiner

PUNJABI UNIVERSITY DIRECTOR INTERNATIONAL AFFAIRS PATIALA

Signature of HOD
Department of Electronics & Communication Engg.
Punjabi University, Patiala-147002

Signature of Dean nominee

THE CULTURAL OFFICE NEW DELHI EMBASSY OF THE REPUBLIC OF IRAQ INDIA

Acknowledgement

I begin this acknowledgment with the name of God, the Most Gracious, the Most Merciful. All praise and thanks are due to Allah, the Lord of all worlds, for His guidance, blessings, and strength throughout the journey of completing this research.

I would like to express my heartfelt gratitude to my supervisor, Dr. SONIA GOYAL, for her unwavering support and valuable guidance. Her insightful feedback and constructive criticism have played a pivotal role in shaping this research.

I am equally indebted to my co-supervisor, Dr. AMRIT KAUR, for her valuable feedback, constructive criticism, and scholarly guidance. Her vast knowledge and insightful suggestions have significantly enriched my research work.

Next, I extend my deepest appreciation to my beloved parents, whose constant encouragement and prayers have been my driving force. Their unwavering belief in my abilities and sacrifices they made for my education have been instrumental in my achievements. I am forever indebted to them for their unconditional love and support.

I would also like to express my gratitude to my dear wife, who has been my pillar of strength and a constant source of motivation throughout this journey. Her patience, understanding, and encouragement have been invaluable. Her unwavering belief in my capabilities has pushed me to overcome challenges and achieve success. I am truly blessed to have her by my side.

Lastly, I would like to acknowledge the support and companionship of my friends. Their unwavering support and encouragement during both challenging and joyful moments are deeply appreciated.

"We raise in degrees whom we will, but over every possessor of knowledge is one more knowing...." (Quran: Surah Yusuf (Joseph): No. 76).

AL-HUSSEIN MOHAMMED JUMAAH

Abstract

This research presents a novel hybrid algorithm that combines the key features of two well-known metaheuristic algorithms: the JAYA Optimization Algorithm and the Artificial Bee Colony (ABC) algorithm. While the ABC algorithm is widely used in the literature, it suffers from certain limitations, such as slow convergence and inadequate exploration versus exploitation. To address these shortcomings, a hybridization of the ABC algorithm with the JAYA algorithm is proposed. This new algorithm, named the Hybrid JAYA and Artificial Bee Colony (JABC) algorithm, incorporates the strengths of both algorithms. The main idea behind the JABC algorithm is to integrate the promising equations of the JAYA algorithm into the ABC algorithm for improved exploitation. Additionally, new mutation operators are introduced to enhance exploration and convergence. The proposed algorithm is evaluated using CEC 2005 benchmark problems as well as real-world synthesis of linear antenna arrays (LAA). We focus on the optimization of position, amplitude, and phase for various LAA configurations, ranging from 10-element to 40-element arrays. To assess the performance of the JABC algorithm, we compare it against other algorithms, including basic ABC, JAYA, Spider Monkey Optimization (SMO), Moth Flame Optimization (MFO), Chameleon Swarm Algorithm (CSA), and others. The experimental results demonstrate that the JABC algorithm outperforms these alternatives significantly, both in terms of benchmark problems and LAA synthesis. Moreover, statistical analysis utilizing Wilcoxon's test and Friedman tests confirms the superior performance of the JABC algorithm compared to the other algorithms. Overall, the proposed hybrid algorithm, JABC, exhibits strong performance and shows promise in addressing the limitations present in the individual ABC and JAYA algorithms.

Table of Contents

List of Publications

(a) International Journals

1. Alhussein M. Alturfi, Sonia Goyal, and Amrit Kaur, "Hybrid Artificial bee colony and JAYA Algorithm for Linear Antenna Array Synthesis", **International Journal of Intelligent Engineering and Systems (SCOPUS INDEXED), Cite Score 2.5,** 2023.

2. Alhussein M. Alturfi, Sonia Goyal, and Amrit Kaur, "Side Lobe Reduction in Linear Antenna Array Using Sine Cosine Algorithm", **Industrial Engineering Journal (UGC INDEXED)**, 2023. (Communicated)

(b) National Journals

1. Alhussein M. Alturfi, Sonia Goyal, and Amrit Kaur, "The Optimum Design of Scanned Linear Antenna Array Using Sine Cosine Optimization Algorithm", **Wasit Journal of Communication and Mathematics**, 2023. (Communicated)

List of Figures

List of Tables

List of Abbreviations

ABC	Artificial bee colony
AF	Array Factor
ALO	Ant Lion Optimizer
BA	Bat Algorithm
BBO	Biogeography-Based Optimization
BFP	Bat Flower Pollinator
CAA	Circular Antenna Array
CABMO	Chaotic adaptive butterfly mating optimization
CCAA	Concentric Circular Antenna Arrays
CFO	Central Force Optimization
CS	Cuckoo Search
CSA	Chameleon Swarm Algorithm
CSCSO	Cuckoo Search–Chicken Swarm Optimization Algorithm
CSO	Cat Swarm Optimization
CVX	Convex Optimization
DE	Differential Evolution
DMO	Dwarf Mongoose Algorithm
EAA	Elliptical Antenna Array
EFA	Enhanced Firefly Algorithm
EM	Electromagnetic Wave
FA	Firefly Algorithm
FNBW	First Null Beam Width
FPA	Flower Pollination Algorithm
GA	Genetic Algorithm
GOA	Grasshopper Optimization Algorithm
GWO	Grey Wolf optimizer
HBA	Honey Badger Algorithm
HHO	Harris Hawks Optimization
ICSO	Improved Cat Swarm Optimization
IWO	Invasive Weed Optimizer
JABC	Hybrid JAYA and Artificial Bee Colony

LAA	Linear Antenna Array
MA	Mayfly algorithm
MFO	Moth Flame Optimization
MSMO	Modified Spider Monkey Optimization
MSOA	Modified Seagull Optimization Algorithm
NMRA	Naked Mole-Rat Algorithm
PABC	Parallel artificial bee colony
PSBL	Peak Sideband Level
PSO	Particle Swarm Optimization
RCGA	Real Coded Genetic Algorithm
RSA	Reptile Search Algorithm
SADE	Self-Adaptive Differential Evolution
SCA	Sine Cosine Algorithm
SLL	Side Lobe Level
SMO	Spider Monkey Optimization
SOS	Symbiotic Organism Search
SSA	Slap Swarm Algorithm
STD	Standard deviation
TMSLA	Time-Modulated Sparse Linear Array
TOM	Taguchi Optimization Method
TS	Tabu Search

Chapter 1
INTRODUCTION

1.1 Introduction

Wireless communication systems, which have expanded quickly, rely on antennas and antenna systems as their eyes and ears. Antenna technology is said to have contributed to some of this development. This is comparable to people whose productivity and everyday activities are primarily controlled by the effectiveness of their eyes and hearing. A radio antenna is " usually a metallic device (as a rod or wire) for radiating or receiving radio waves", according to Webster's Dictionary. Essentially, an antenna is a transducer. It converts an RF signal into an electromagnetic wave (EM) with the same frequency. It is a component of the receiver and transmitter circuits. Resistance, inductance, and capacitance are present in its equivalent circuit, which defines it. A charge creates an electrostatic field, whereas a current creates a magnetic field. In turn, these two produce an induction field [1].

Antennas are the backbone of modern wireless devices, including mobiles, radios, radars, and satellites. In these technologies, a set of antennas are used instead of a single antenna because of their capabilities in controlling the main lobe, radiation pattern and adjusting parameters such as positions, excitation phases, and excitation currents. Antenna arrays, in general, help in reducing the power consumption, side lobe level, and enhance signal-to-noise ratio; and can have different geometries, namely rectangular, elliptic, linear, hexagonal and circular. Among all, Linear antenna arrays (LAAs) are widely used and considered the most basic type of antenna arrays. In an LAA, the elements are placed along a single axis, owing to steer in a particular direction and provide an omnidirectional radiation pattern for diversity in one plane.

To achieve an optimal radiation pattern that balances both the array's directivity and interference limitations, it is necessary to minimize the Side Lobe Level (SLL) while simultaneously increasing the array's directivity. This is crucial in improving communication quality and avoiding interference with other systems operating within the same frequency range [2]. However, designing an antenna array with both low SLL and a narrow First Null Beam Width (FNBW) poses a challenge, as reducing the SLL may result in a wider FNBW, and vice versa.

Over the past three decades, many optimization algorithms have been proposed and have been exploited for almost every research problem, from medicine to management, business to electronics, operation research, routing problems, and others. The major reason for using these algorithms is due to their simple and linear structure, lesser known parameters for tuning, and better convergence results, among others. These algorithms are commonly known as Nature Inspired Algorithms (NIAs) and are categorized into Evolutionary Algorithms (EAs) and Swarm Intelligent Algorithms (SIAs). EAs are based on the Darwinian theory of natural selection. Among EAs, the Genetic Algorithm (GA) is the oldest known algorithm [3], and Differential Evolution (DE) [4] is another important algorithm. SIAs, on the other hand, are based on the swarming behavior of various animal species, and some of the algorithms include the Naked Mole-Rat Algorithm (NMRA) [5] based on the mating patterns of naked mole-rats, Grey Wolf optimizer (GWO) [6] based on swarming behavior of grey wolves found in nature. Other algorithms include Sine Cosine Algorithm (SCA) [7], Harris Hawks Optimization (HHO) [8], Aquila Optimizer [9], Reptile Search Algorithm (RSA) [10], Dwarf Mongoose Algorithm (DMO) [11] and others.

All major algorithms include initialization, global search (exploration), local search (exploitation), and selection as their basic operations. Apart from that, the performance of these algorithms is based on various aspects, including scaling factor, mutation probability, crossover rate, and population, among others [12]. Initial studies showed that parameter tuning is a crucial step in any algorithm and can be time-consuming if based on a trial-and-error approach. Premature convergence, local optima stagnation, higher parameter dependence, and other issues make an algorithm prone to poor solutions, and hence can result in reduced performance [13]. This proves that NIAs are prone to various different problems, and new algorithms must be designed to solve the problem under consideration.

Synthesis of LAA is a very intuitive subject and has been exploited vastly in the literature. Many optimization algorithms have been applied for LAA synthesis, including Grasshopper Optimization Algorithm (GOA) [14], Ant Lion Optimizer (ALO) [15], Flower Pollination Algorithm (FPA) [16], Symbiotic Organism Search (SOS) [17], Biogeography Based Optimization [18], Particle Swarm Optimization (PSO) [19], Taguchi's Optimization Method (TOM) [20], Cuckoo Search (CS) [21], Enhanced Firefly Algorithm (EFA) [22], Bat Flower Pollinator (BFP) [23], Invasive Weed Optimizer (IWO) [2], Tabu Search (TS) [24], Hybrid algorithm [25], Slap Swarm Algorithm (SSA) [26], Modified Seagull Optimization Algorithm (MSOA)[27], Cuckoo Search–Chicken Swarm Optimization Algorithm (CSCSO)[28] and others.

Since introducing the Sine Cosine algorithm (SCA), it has demonstrated its ability to address various optimization problems, namely feature selection, robot route planning, economic dispatch, image processing, and many more [29]. In this study, the SCA method is employed for the first time for antenna array optimization, and it was suggested in [30] as a method for solving optimization problems. SCA may be a more appropriate choice than other methods for solving various optimization problems. On the other hand, this method not better than other algorithms on a particular set of issues. However, it is worth testing and applying to issues in other fields. Thus, the SCA method is provided to researchers in several domains [29].

Artificial bee colony (ABC) is one such algorithm introduced in the recent past [31]. The algorithm is based on the foraging patterns of bees found in nature. The algorithm is based on the division of bees into three groups; employed bees, onlooker bees, and scouts. The bee colony is divided into two parts, where the first half has employed bees, and the second part consists of onlooker bees. For each of these phases, there is only one artificial bee for one food source and an employed bee whose food source becomes exhausted; that artificial bee becomes the scout. The algorithm follows many iterations to provide the final best solution. Each search process consists of three steps: i) the employed and onlooker bees move onto the food, and the nectar amount is calculated, ii) the best scout bee is identified, and iii) the scout bee is directed toward the food. Each position of the food source acts as the potential solution of the problem [31].

Many articles have been proposed on the modification and application of the ABC algorithm. Some of these include time varying ABC [32], parallel ABC [33], discrete ABC for cloud services [34], hybrid DE with ABC [35], ABC for binary and integer problems [36], ABC for engineering design problems [37], application of ABC in medical imaging [38], for combinatorial problems [39], among others. Most of the articles discussed above are review articles and provide in-depth details on the variants and applications of ABC. But it has been found in the literature that the algorithm still suffers from various problems, including local optima stagnation, poor exploration, and slow convergence, among others [40]. Also, very little work has been done to improve the parameters of the algorithm, and simultaneous efforts need to be done. Most of the work done on ABC is concentrated around equation modifications or simple applications to the problem, and less effort has been made to make it a generic problem solver. In order to deal with these problems, JAYA algorithm [41] based modification is added to ABC. The JAYA algorithm is simple in structure and is based on the concepts of moving the solution obtained for the problem towards the best solution and simultaneously avoiding the worst solution. The algorithm is found to be highly reliable due to its parameter less nature and hence is found to provide viable solutions. But as the problem complexity increases, modifications must be added to make it suitable for challenging problems.

The above said limitations of existing algorithms have motivated the authors to propose a new algorithm. The new algorithm has been named as hybrid ABC and JAYA algorithms (JABC) and is meant to overcome the problems of ABC as well as JAYA. The modifications are added by enhancing the equations of ABC for both the employed and onlooker phase, and for the scout phase JAYA based equations are used. Modifications are added in the generalized parameters (including simulated annealing based parametric adjustments) of the employed and onlooker phase. This helps the algorithm performing extensive exploration and exploitation of the search space, along with a balanced operation. Introducing JAYA into the scout phase makes the scout phase parameter independent and also helps to provide better convergence properties. More details about the proposed modifications are presented in consecutive subsections. For performance evaluation, we use CEC 2005 benchmark problems [42], consisting of unimodal, multimodal, and some fixed dimension test problems. These test functions are highly challenging and are used for the performance evaluation of most of the newly proposed algorithms. Apart from the benchmark problems, the linear antenna array synthesis is also done using the proposed JABC algorithm.

For comparative study, ABC [31], JAYA [41], and other algorithms have been used. A statistical analysis using Wilcoxon's ranksum and Friedman test [37] is also done to prove the significance of the proposed algorithm.

1.2 Research objectives

- To study the various existing methods for side lobe level (SLL) reduction of Linear Antenna Array (LAA).
- To optimize the linear antenna array by varying excitation amplitude, phase, and distance between the elements using optimization algorithms.
- To evaluate and compare the performance of optimized linear antenna array with state of art techniques.

1.3 Scope of work

This research focused on the challenging task of finding a suitable optimization method to effectively solve real-world problems. Among the existing algorithms, Artificial Bee Colony (ABC) has gained popularity and widespread usage in the literature. However, despite its popularity, the ABC algorithm suffers from inherent drawbacks such as slow convergence and poor exploration versus exploitation balance.

To address these limitations, we propose a hybrid approach that combines the strengths of the ABC algorithm with the JAYA algorithm. By incorporating the main properties of both algorithms, we aim to enhance the performance and overcome the drawbacks observed in ABC. The resulting algorithm is named JABC (JAYA and Artificial Bee Colony) algorithm.

In addition to combining the key characteristics of ABC and JAYA, new mutation operators are introduced, designed to improve exploration and convergence. These mutation operators enhance the algorithm's ability to explore the search space effectively, allowing for a more thorough exploration of potential solutions. Moreover, they contribute to achieving better convergence by guiding the algorithm towards optimal or near-optimal solutions.

The JABC algorithm not only inherits the beneficial properties of both ABC and JAYA but also incorporates novel enhancements through the introduction of new mutation operators. This hybridization aims to address the drawbacks of the original ABC algorithm and improve its overall performance in terms of convergence speed, exploration capabilities, and solution quality.

1.4 Structure of dissertation

Chapter 1 describes introduction and Scope, in this introductory chapter, a comprehensive overview of the research topic is provided. The chapter begins by presenting an overview of antennas, optimization algorithms, and introduces the ABC algorithm along with its application. Furthermore, it addresses the challenges associated with the ABC and JAYA algorithms and outlines the scope and objectives of the dissertation. This chapter sets the stage for the subsequent chapters by providing a clear context for the research.

Chapter 2 focuses on conducting a thorough literature survey, examining previous studies and research related to antenna array synthesis. The chapter highlights the intelligent optimization algorithms that have been utilized in this field, showcasing their applications and discussing their effectiveness. This review of the existing literature helps establish the current state-of-the-art and provides a foundation for the subsequent chapters.

Chapter 3 describes algorithms and problem formulation, and divided into three sections, each serving a specific purpose. The first section provides a detailed introduction to the basic concepts of SCA (Shuffled Complex Algorithm), JAYA, ABC, and JABC (a combination of ABC and JAYA). It explains why and how these algorithms were proposed, offering insights into their underlying principles. The second section presents a performance evaluation of these algorithms using the CEC 2005 test functions, highlighting their strengths and weaknesses.

Finally, the third section delves into the synthesis of linear antenna arrays, discussing the problem formulation and the specific challenges faced in this area.

Chapter 4 presents the results obtained from the synthesis of Linear Antenna Arrays (LAA) using the proposed optimization algorithms. The chapter includes a comprehensive analysis and discussion of these results, examining the performance of the antenna arrays in terms of SLL reduction and other relevant metrics. The findings are carefully interpreted and compared with the objectives and expectations outlined in earlier chapters, providing insights into the effectiveness of the proposed methods.

Chapter 5, the final chapter of the dissertation summarizes the main findings, conclusions, and implications drawn from the research. It highlights the key contributions and their significance in the context of the broader field of antenna array synthesis. Additionally, this chapter discusses possible future prospects and areas for further exploration and development, suggesting avenues for potential research and advancements in the field.

Chapter 2
SURVEY OF LITERATURE

2.1 Literature Survey

In recent years, the synthesis of linear antenna arrays (LAA) has gained significant attention in the field of antenna design and optimization. The synthesis of LAA involves the determination of optimal positions, amplitudes, and phases of individual antenna elements to achieve desired radiation patterns and performance characteristics. This process plays a crucial role in various applications, including wireless communications, radar systems, and smart antennas.

Several optimization algorithms have been proposed and applied for the synthesis of LAA. These algorithms aim to find the best set of parameters that minimize side lobe levels (SLL), maximize directivity, achieve narrow beam widths, and meet other design specifications. The performance of these algorithms heavily depends on their ability to balance exploration and exploitation, efficiently search the solution space, and overcome challenges such as local optima and high-dimensional optimization problems. The intelligent optimization strategies utilized for antenna array synthesis are illustrated in this chapter.

Yan and Lu [43] proposed a Genetic algorithm for the synthesis of the linear antenna array (LAA) and circular antenna array (CAA) by optimizing their excitation current for the pattern synthesis of an antenna array.

Tian and Qian [44] used GA to improve LAA synthesis by optimizing excitation current and array element position simultaneously. They also designed the fitness function to place a soft constrain on the searching space to compress the optimum range and speed convergence to improve GA's global searching characteristic. Moreover, they used adaptive searching mechanisms and immune operators to compensate for it being trapped in local optimum and save computation time. Even while GA has a competitive edge in terms of global search capability, it falls short due to its tendency to become trapped at local optimums and its poor computing speed.

Shuang Liang et al. (2017) utilized cuckoo search–chicken swarm optimization (CSCSO) to optimize the excitation current of an LAA as well as the excitation current and array element position of a CAA. Cuckoo search (CS) and chicken swarm optimization (CSO) are combined in CSCSO. The simulation results show that CSCSO performs better than the standard CSO, CS, and Particle Swarm Optimization (PSO) regarding solution accuracy and convergence rate in the radiation beam pattern optimization[28].

Reference [45] recommended the use of the firefly algorithm (FA) to minimize peak SLL by optimizing the amplitude current of LAA. In contrast, reference [46] used FA to design LAA with non-uniform spacing. In both circumstances, FA performs better than PSO in terms of convergence rate and global optimal solution. This is because FA can identify both global and local optimums efficiently.

Sharaqa and Dib use the Biogeography-Based Optimization (BBO) technique [18] to reduce the maximum SLL of LAA's and elliptical antenna array. Three optimization cases are investigated: positions, excitation phases, and excitation currents optimization.

The Genetic Algorithm (GA), Particle Swarm Optimization (PSO), and Weightless Swarm Algorithm (WSA) are three meta-heuristic algorithms used to reduce the maximum SLL and increase the capacity, as described in reference [47]. The results demonstrate that WSA is superior to PSO and GA in reducing the maximum SLL when dealing with small clusters, peak SLL is reduced by a greater amount, which improves capacity at an unintended receiver.

Saxena and Kothari [48] used the Grey Wolf Optimization technique to optimize the position and amplitudes of an antenna array to produce the required array design with least SLL and null placement in specified directions. It was also shown that the near side lobe may be suppressed while the other side lobes were controlled. The findings show that, compared with a uniform array and synthesis produced from other optimization approaches, significant improvements may be achieved when LAA is optimized using GWO.

The performance of the Enhanced Firefly method (EFA) method has been evaluated on eleven test functions and contrasted with other well-known algorithms. In terms of finding the global optimum, simulation results demonstrate that the EFA algorithm is better than the Artificial Bee Colony (ABC), Firefly Algorithm (FA), Flower Pollination Algorithm (FPA) and some others. Convergence profiles and statistical analysis are also used to verify EFA's superior performance. It also applied for the synthesis of non-uniform LAA by optimizing their excitation current and position for the pattern synthesis of an antenna array. EFA achieves peak SLL values of -33.62 dB and -37.36 dB for the synthesis of the excitation amplitudes of 16 and 26 elements, respectively, while achieving a peak SLL of -23.90 dB for the synthesis of positions of 32 elements, with a null value less than -60 dB [22].

The implementation of the Slap Swarm Algorithm (SSA) technique on antenna array was carried out by Luo, Z. et al. (2021) [26] for the design of both linear and planar sparse antenna arrays with optimum SLL by optimizing their excitation currents and positions. The SSA algorithm has a very low convergence rate compared to existing heuristic algorithms, but it is simple to construct and only needs a few input parameters. This technique performs better regarding the SLL reduction of sparse arrays, according to numerical findings for both linear and planar arrays.

Majid Khodier (2019) successfully used the Cuckoo search (CS) method to control essential antenna array radiation properties, including SLL, First Null Beam Width (FNBW), and nulls, using the array's element's positions, excitation phases, and excitation currents optimization. When compared to other nature-based algorithms, the CS algorithm sometimes produced outcomes ranging from slightly better, slightly worse, or approximately the same. Therefore, CS is considered generic and robust for many optimization problems compared with other nature-inspired algorithms [21].

Dib. [17] implemented Symbiotic Organism Search (SOS) for the design of LAA by optimizing several array parameters (position, amplitude, and phase), it is possible to reduce the maximum SLL and impose nulls at certain angles for isotropic LAA. , the results drawn indicate the performance of the SOS method is better than Particle Swarm Optimization (PSO) , Genetic Algorithm (GA), Firefly Algorithm (FA), and others. SOS achieves maximum SLL values of -25.28 dB, -33.39 dB, and -39.37 dB for the synthesis of the excitation currents of 10, 16, and 24 elements, respectively. It also achieves a peak SLL of -20.93 dB for synthesizing positions of 37 elements and -18.02 dB for synthesizing the excitation phases. However, compared to SOS, BBO, PSO, and Taguchi, the synthesis result produced by Moth Flame Optimization (MFO) [49] for both LAA and CAA optimization improves the SLL.

Liang et al. proposed a novel approach for the design of a time-modulated sparse linear array (TMSLA) by combining particle swarm optimization (PSO) and convex optimization (CVX) techniques [50]. Their algorithm aimed to minimize the peak side lobe level (SLL), peak sideband level (PSBL), and grating lobes of the LAA. By integrating the strengths of PSO and CVX, the authors were able to achieve significant reductions in these undesired parameters, improving the overall performance of the antenna array.

Liang et al. [51] suggested an improved cat swarm optimization (ICSO) for reducing SLL in linear, circular, and random antenna arrays and compared the ICSO's results to those of seven present heuristic techniques. The optimization performance was improved as a result of the updated local search factor, weighting factor, and global search factor.

The utilization of the Mayfly algorithm (MA) [52] offers an effective approach for optimizing the excitation current and element spacing in a linear antenna array (LAA). In the absence of null control, the MA algorithm demonstrates impressive results in minimizing the peak side lobe level (SLL) across different LAA configurations. Specifically, it achieves remarkable peak SLL values of -26.70 dB, -48.27 dB, -34.90 dB, and -35.73 dB for the excitation current synthesis of 10, 16, 20, and 32 LAA elements, respectively. Additionally, in the absence of null control, MA achieves a peak SLL of -22.79 dB, 24.97 dB, and -23.24 dB for the position synthesis of 10, 20, and 32 LAA elements, respectively. These outcomes surpass the performance of other optimization algorithms such as Grey Wolf Optimization (GWO) [48], Invasive Weed Optimizer (IWO) [2], Firefly Algorithm (FA) [53], and Moth Flame Optimization (MFO)[49].

Asem S. et al. suggested in [25] the use of a hybrid optimization algorithm based on the Grasshopper Optimization Algorithm (GOA) and Ant Lion Optimizer (ALO) for minimizing the maximum SLL, controlling the FNBW and imposing nulls at particular angles in some designs, which are accomplished by optimizing various array parameters (position, amplitude, and phase). In contrast, in [54], the optimal design of scanned LAA was performed by reducing SLL with the constraint of a fixed FNBW using the same proposed algorithm in [25]. In both circumstances, the Hybrid algorithm performs better than ALO [15], SOS [17], FA [53], and others in terms of convergence rate and optimal global solution.

For the optimizations of LAAs, reference [38] proposes a method that is based on Grasshopper Optimization Algorithm (GOA). The algorithm is applied to multiple LAA pattern synthesis and broadband optimization of whip antenna for minimizing the maximum SLL, controlling the FNBW, and imposing nulls at particular angles in some designs, which are accomplished by optimizing the position and amplitudes of the antenna array elements, the GOA algorithm is superior to Particle Swarm Optimization (PSO) [19], Cat Swarm Optimization (CSO) [55], Spider Monkey Optimization (SMO) [56], Flower Pollination Algorithm (FPA) [16], Cuckoo Search (CS) [21], Invasive Weed Optimizer (IWO) [2], Real Coded Genetic Algorithm (RCGA) [56], Modified Spider Monkey Optimization (MSMO) [57] , Enhanced Firefly Algorithm (EFA) [22], and other algorithms in reducing SLL.

Table 2.1 presented below, offers a comprehensive summary of the studies that were included in the review. These studies, spanning the period from 2017 to 2022, have been carefully examined and analyzed to provide valuable insights and findings for this research.

Table 2.1: Summary of Literature Survey

Year	Author	Title	Journal	Optimization algorithm	Finding			
Jan,15,2022 [25]	Al-Zoubi AS, Amaireh AA, Dib NI.	comprehensive study of linear antenna arrays' synthesis.	International Journal of Electrical and Computer Engineering (IJECE). (Scopus)	hybrid optimization algorithm based on GOA and ALO.	**Varying**	**N (elements)**	**Peak SLL (dB)**	**FNBW**
					I_n (amplitude)	16	-33.36	23.2◦
					x_n (position)	40	-20.89	-
					α_n (phase)		-18.13	6.4◦
Jan, 2022 [58]	Vegesna N, Yamuna G, Kumar TS.	Linear Antenna Array synthesis using Rao and Jaya algorithms.	International Journal of Knowledge-based and Intelligent Engineering Systems. (Scopus)	Rao and Jaya algorithms	**Varying**	**N (elements)**	**Peak SLL (dB)**	
							JAYA	**RAO-1**
					I_n (amplitude)	20	-52.24	-41.40
						40	-45.39	-42.56
						60	-41.43	-30.64
						80	-40.65	-37.71
Dec.,12, 2021 [27]	Kurt E, Basbug S, Guney K.	Linear antenna array synthesis by modified seagull optimization algorithm	The Applied Computational Electromagnetics Society Journal (ACES).	Modified Seagull Optimization Algorithm (MSOA)	**Varying**	**N (elements)**	**Peak SLL (dB)**	
					I_n (amplitude)	10	−27.52	
						16	−33.53	
					x_n (position)	12	−19.25	

<table>
<tr><td rowspan="5">Jan,20,2020 [14]</td><td rowspan="5">Wang H, Liu C, Wu H, Li B, Xie X.</td><td rowspan="5">Optimal pattern synthesis of linear array and broadband design of whip antenna using grasshopper optimization algorithm</td><td rowspan="5">International Journal of Antennas and Propagation. (Scopus)</td><td rowspan="5">Grasshopper Optimization Algorithm (GOA)</td><td>Varying</td><td>N (elements)</td><td colspan="2">Peak SLL (dB)</td></tr>
<tr><td rowspan="2">$\boldsymbol{I_n}$ (amplitude)</td><td>10</td><td colspan="2">-21.31</td></tr>
<tr><td>16</td><td colspan="2">-24.61</td></tr>
<tr><td rowspan="2">$\boldsymbol{x_n}$ (position)</td><td>10</td><td colspan="2">-27.36</td></tr>
<tr><td>16</td><td colspan="2">-28.10</td></tr>
<tr><td rowspan="6">October, 2021 [59]</td><td rowspan="6">Durmus, A.</td><td rowspan="6">Novel metaheuristic optimization algorithms for sidelobe suppression of linear antenna array</td><td rowspan="6">International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT). (IEEE Xplore)</td><td rowspan="6">Honey Badger Algorithm (HBA) and Chameleon Swarm Algorithm (CSA) algorithms in</td><td rowspan="2">Varying</td><td rowspan="2">N (elements)</td><td colspan="2">Peak SLL (dB)</td></tr>
<tr><td>HBA</td><td>CSA</td></tr>
<tr><td rowspan="4">$\boldsymbol{I_n}$ (amplitude)</td><td>10</td><td>-26.98</td><td>-26.99</td></tr>
<tr><td>16</td><td>-40.12</td><td>-40.25</td></tr>
<tr><td>24</td><td>-40.52</td><td>-40.90</td></tr>
<tr><td>32</td><td>-23.02</td><td>-22.96</td></tr>
<tr><td rowspan="5">Jul,2019 [21]</td><td rowspan="5">Khodier M.</td><td rowspan="5">Comprehensive study of linear antenna array optimization using the cuckoo search algorithm</td><td rowspan="5">IET Microwaves, Antennas & Propagation. (Scopus)</td><td rowspan="5">Cuckoo Search Algorithm (CS)</td><td>Varying</td><td>N (elements)</td><td colspan="2">Peak SLL (dB)</td></tr>
<tr><td rowspan="3">$\boldsymbol{I_n}$ (amplitude)
FNBW Varying</td><td>10</td><td colspan="2">-24.42</td></tr>
<tr><td>11</td><td colspan="2">-25.30</td></tr>
<tr><td>24</td><td colspan="2">-31.42</td></tr>
<tr><td>$\boldsymbol{I_n}$ (amplitude)
FNBW Constant</td><td>10</td><td colspan="2">-15.99</td></tr>
</table>

					x_n (position)		10	-19.72	
					α_n (phase)		20	-16.27	
							40	-17.59	
					I_n, x_n (amplitude, position)		10	-19.67	
Apr, 2020 [54]	Amaireh, A.A., Al-Zoubi, A.S. and Dib, N.I.,	The optimal synthesis of scanned linear antenna arrays.	International Journal of Electrical and Computer Engineering (IJECE). (Scopus)	Hybrid optimization algorithm based on GOA and ALO.	**Varying**	$\emptyset_d$ (angle at which the major lobe is oriented)	**N (elements)**	**Peak SLL (dB)**	**FNBW**
					I_n (amplitude)	30 ∘	20	-15.66	29.44
						45 ∘	26	-16.32	-
						60 ∘	30	-16.20	8.96
Mar, 2019 [22]	Singh U, Salgotra R.	Synthesis of linear antenna arrays using enhanced firefly algorithm	Arabian Journal for Science and Engineering. (Scopus)	Enhanced Firefly Algorithm (EFA)	**Varying**		**N**	**Peak SLL (dB)**	
					I_n (amplitude)		16	-33.62	
							24	-37.36	
							32	-23.90	
Oct,2017 [60]	Amaireh AA, Alzoubi A, Dib NI.	Design of linear antenna arrays using antlion and grasshopper optimization algorithms	Jordan conference on applied electrical engineering and computing technologies (AEECT)	Antlion (ALO) and Grasshopper Optimization Algorithms (GOA)	**Varying**		**N (elements)**	**Peak SLL (dB)**	
								GOA	**ALO**
					I_n (amplitude)		10	-25.24	-25.23
							16	-33.34	-33.36
							24	-39.09	-39.42

The literature survey in Table 2.1 provides valuable insights into the performance of various optimization algorithms for LAA synthesis. The comparative analysis reveals the strengths and weaknesses of each algorithm in terms of peak SLL reduction, main lobe steering, FNBW, and convergence rate. Based on the findings, it is evident that no single algorithm excels in all aspects, and the choice of optimization algorithm depends on the specific objectives and requirements of the LAA system. Further research and experimentation are needed to explore hybrid approaches and improve the performance of optimization algorithms for LAA synthesis. Addressing the limitations of the literature survey, such as sample sizes and experimental variations, will enhance the reliability and generalizability of the results.

Overall, this literature survey serves as a valuable resource for researchers and practitioners in the field of antenna array optimization. It guides the selection of suitable algorithms and provides directions for future studies aimed at advancing the design and performance of LAA systems.

2.2 Research Gap

After conducting a comprehensive literature survey, several research gaps in the synthesis of linear antenna arrays (LAA) have been identified, highlighting areas for further investigation and improvement. The specific research gaps are as follows:

1. The existing research by Vegesna N. et al. (2022) has achieved a noteworthy reduction in the side lobe level (SLL) of 20 element arrays, reaching a value of -52.243 dB [58]. However, there is still potential for further minimizing the SLL through the utilization of appropriate optimization algorithms. The aim is to explore and develop optimization techniques that can effectively enhance the performance of LAA by achieving even lower SLL values.

2. It has been observed that while reducing the SLL, there is often a trade-off resulting in a wider beam width. This implies that narrower beam widths do not necessarily lead to lower side lobe levels. Therefore, it is crucial to maintain a constant first null beam width (FNBW) while minimizing the SLL. The research should focus on developing methodologies that strike a balance between reducing the SLL and preserving a narrow beam width, thus achieving optimal antenna array performance.

3. While various optimization algorithms have been implemented to minimize the SLL of linear antenna arrays (LAA), there is a significant scope for further advancements. Hybrid optimization algorithms, which combine the strengths of multiple techniques, offer promising potential for reducing the SLL even further. Exploring and implementing such hybrid optimization algorithms can lead to enhanced performance and improved SLL reduction in LAA designs.

Addressing these research gaps will contribute to the advancement of LAA synthesis methodologies, enabling the development of more efficient and effective antenna arrays with reduced SLL and maintained FNBW.

Chapter 3

Synthesis of Linear Antenna Array Using Optimization Algorithms

In this research, four optimization algorithms were implemented for the synthesis of a Linear Antenna Array (LAA). Firstly, the Sine Cosine Algorithm (SCA), Artificial Bee Colony (ABC) algorithm, and JAYA algorithm were implemented for this objective. However, these algorithms have some inherent drawbacks, including slow convergence, poor exploration versus exploitation, among others. In order to deal with these problems, this research proposed a new hybrid algorithm that enhances the performance of the ABC and JAYA algorithms by combining the main characteristics of both techniques. This hybrid algorithm has been named the Hybrid JAYA and Artificial Bee Colony (JABC) algorithm. This chapter is divided into three sections. The first section provides a brief introduction to the basic SCA, JAYA, ABC, and JABC algorithms, explaining why and how these algorithms were proposed. In the second section, we provide a performance evaluation for the CEC 2005 test functions [42]. The third part gives a basic overview of the geometry of the Linear Antenna Array (LAA) and addresses the problem formulation.

3.1 Optimization Algorithms

In this research, four optimization algorithms implemented for synthesis Linear Antenna Array (LAA), and they are Sine Cosine Algorithm (SCA), JAYA, Artificial Bee Algorithm (ABC) algorithm, and hybrid JAYA and Artificial Bee Colony (JABC) algorithm.

3.1.1 Sine Cosine Algorithm (SCA)

Sine Cosine Algorithm (SCA) is a meta-heuristic technique created by Mirjalili in 2016 that utilizes the mathematical properties of the sine and cosine functions and has attracted the interest of investigators because of its fair execution time, strong convergence acceleration rate, and great efficiency in comparison to other well-respected optimization algorithms present in the literature [29].

The first step deals with initialization of sources in the random manner for the dimension of the problem. This is mathematically given as:

$$x_{i,k} = x_{min,k} + r(0,1) \times (x_{min,k} - x_{max,k}) \tag{3.1}$$

where, $i \in [1, 2, \ldots.. n], k \in [1, 2, \ldots d]$, $x_{i,k}$ represents , i^{th} solution in k^{th} dimension; r(0, 1) is a random number; $x_{min,k}$ and $x_{max,k}$ represents the lower bound and the upper bounds of the problem. This step is the same for the whole algorithm and provides the initial set of solutions for the performance evaluation.

Like other meta-heuristic methods, SCA begins with a collection of distributed solutions. The following equations are then used to update each solution:

$$x_i^{t+1} = x_i^t + r_1 sin\,(r_2) \times |r_3 P_i^t - x_i^t| \tag{3.2}$$

$$x_i^{t+1} = x_i^t + r_1 cos\,(r_2) \times |r_3 P_i^t - x_i^t| \tag{3.3}$$

Where, x_i^t and x_i^{t+1} represents the i^{th} position of the present solution at iteration *t* and *t+1*, respectively. r_1, r_2, r_3 represents random numbers, P_i^t the fittest solution in i^{th} position within the solution set. Eq. (3.2) and (3.3) are employed in SCA in the following way:

$$x_i^{t+1} = \begin{cases} x_i^t + r_1 \times \sin(r_2) \times |r_3 P_i^t - x_i^t|, r_4 < 0.5 \\ x_i^t + r_1 \times \cos(r_2) \times |r_3 P_i^t - x_i^t|, r_5 \geq 0.5 \end{cases} \tag{3.4}$$

Where, r_4 is an interval-based random number (0,1). According to the equations above, SCA has four predominant parameters (r_1, r_2, r_3, and r_4). The parameter r_1 determines the subsequent location region, which may or may not be present in the region bounded by the solution and the destination. Examining and utilizing a search area and striking the correct balance between the two are further advantages of this parameter. The first half of the maximum amount of iterations is dedicated to diversification, while the second half is committed to the intensification of a search area that is accessible [25]. r_1 may be described mathematically as follows:

$$r_1 = a - a \times \frac{t}{T} \tag{3.5}$$

Where $'a'$ is a constant, $'t'$ is the present iteration, and $'T'$ represents the highest value of iterations. The r_2 parameter determines the direction of a present solution's moment. The random parameter r_3 is for the weighting of P_i^t. Finally, the parameter r_4 a controllable variable that toggles between sine and cosine transitions of Eq. (3.3) [29]. The tuning parameter and flowchart of the SCA algorithm are shown in Table 3.1 and Figure 3.1, respectively.

Table 3.1: The selected range for the SCA tuning parameters.

Parameter	Population	Iterations	r_2	r_3	r_4	a
Value	50	150-1000	$[0,2\pi]$	[0,2]	[0,1]	0.15

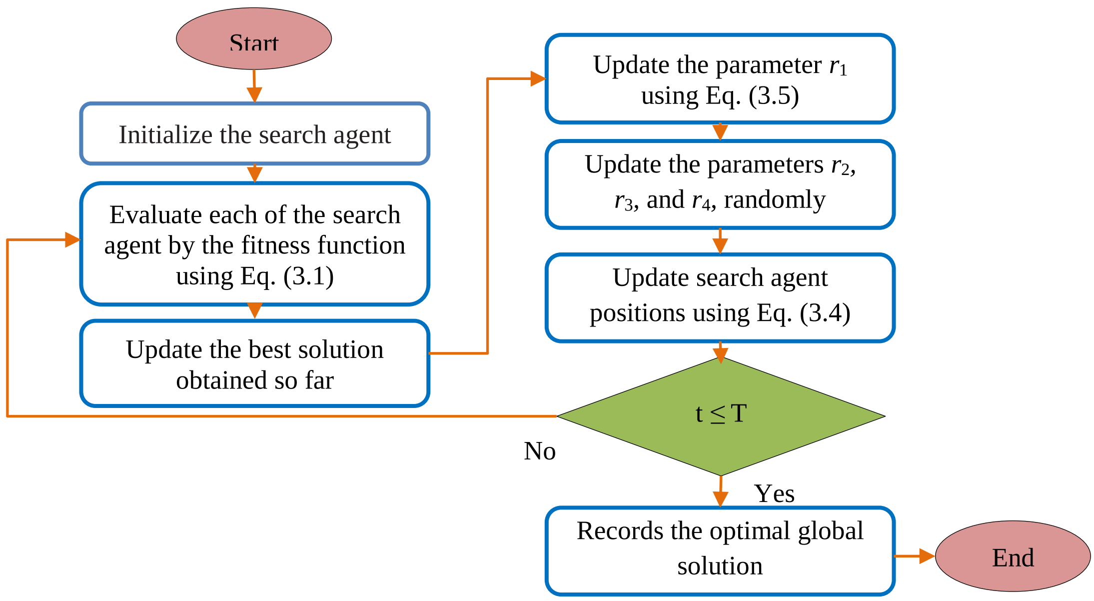

Figure 3.1: Flowchart of the Sine Cosine Algorithm (SCA)

3.1.1.1Effect of variation parameter "a" for Sine Cosine Algorithm (SCA)

In order to analyze the impact of a specific parameter, denoted as "a," on the overall fitness of the system, we conducted a series of experiments while keeping all other parameters constant. The value of "a" was varied at three different settings: 0.15, 0.5, and 2. Table 3.2 presents the fitness values obtained for each variant of "a." The fitness values provide a measure of how well the system performs in achieving its objectives. In this case, we are interested in finding the optimal value of "a" that yields the best average outcomes across all design case studies.

Table 3.2: The optimal fitness function value, as determined by the SCA technique, for the scanned LAA in response to a change in parameter "a".

Fitness value for 20 elements with $\emptyset_d = 30^{\circ}$ in dB			
a	0.15	0.5	2
Worst	-16.5012	-16.5585	-16.4041
Best	**-16.7023**	-16.6235	-16.5450
Average	-16.6345	-16.5818	-16.4949
Fitness value for 26 with $\emptyset_d = 45^{\circ}$ elements in dB			
a	0.15	0.5	2
Worst	-17.9342	-17.8570	-17.7011
Best	**-18.0550**	-17.9078	-17.7946
Average	-17.9973	-17.8776	-17.7456
Fitness value for 30 elements with $\emptyset_d = 60^{\circ}$ in dB			
a	0.15	0.5	2
Worst	-17.9296	-17.8685	-17.6778
Best	**-18.1399**	-17.9552	-17.7440
Average	-18.0090	-17.9259	-17.7074

After analyzing the results, it is evident that setting "a" to 0.15 consistently leads to the best average outcomes for all the design cases. This implies that a value of 0.15 for "a" effectively balances the system's performance and helps achieve the desired objectives more effectively than the other tested values. By understanding the influence of "a" on the fitness values, we gain valuable insights into the parameter's role in optimizing the system's performance. This knowledge can guide future decision-making processes and assist in fine-tuning the system for improved results.

3.1.2 JAYA Algorithm

Rao has created an algorithm called the Jaya algorithm which uses a swarm-based method to solve both constrained and unconstrained optimization problems. During each iteration, the search agents avoid the worst solutions (failures) and move closer to the goal. Only the best solutions are kept while all the other positions are discarded [41]. The positions of the search agents are mathematically adjusted using a specific equation. This equation is detailed in Eq. (3.6).

$$x_i^{t+1} = x_i^t + r_1(x_{best} - x_i^t) - r_2(x_{worst} - x_i^t) \tag{3.6}$$

Where, x_i^t and x_i^{t+1} represents the i^{th} position of the present solution at iteration *t* and *t+1*, respectively. r_1 and r_2 are random numbers, x_{best} and x_{worst} are the best and the worst solutions, $(x_{best} - x_i^t)$ indicates the capability of each solution to move towards the best solution and $(x_{worst} - x_i^t)$ represents the capability to move away from the worst solution.

The Jaya algorithm is an effective optimization technique that is easy to implement and requires only minimal tuning of common control parameters such as population size and number of iterations. No additional parameters are needed for successful implementation. The algorithm primarily follows an iterative approach and is further depicted in the flowchart of Figure 3.2. It can be used to solve a wide range of optimization problems in both single-objective and multi-objective variants. Various cases, such as global optimization of non-linear functions, antenna design and multi-objective scheduling have been reported as successful applications of the Jaya algorithm.

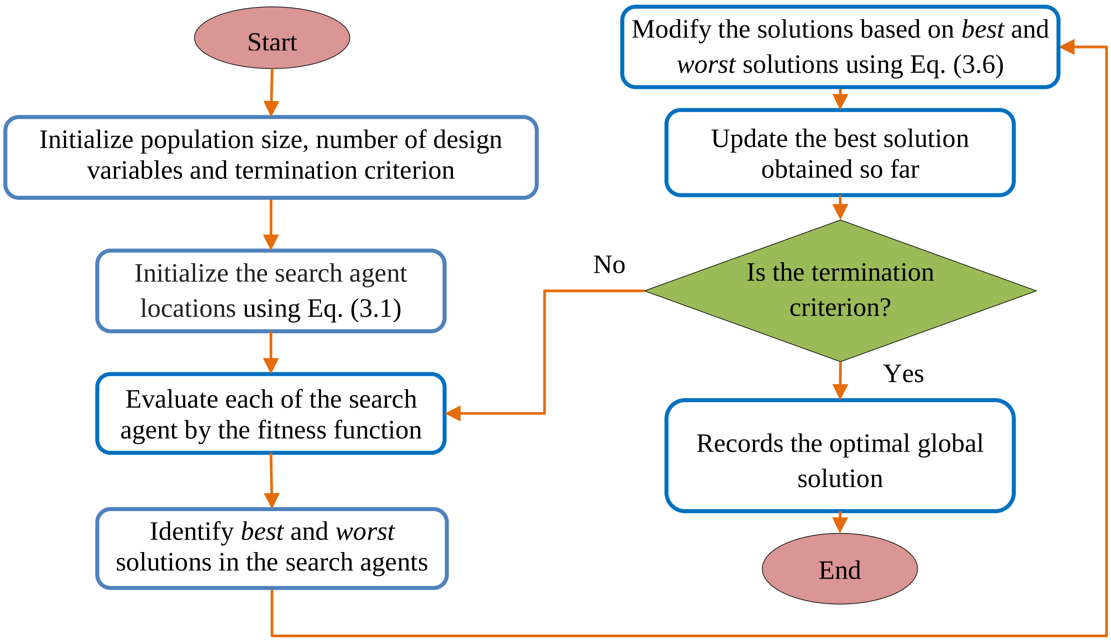

Figure 3.2: Flowchart of the JAYA Algorithm.

3.1.3 Artificial Bee Colony (ABC) Algorithm

The Artificial Bee Colony (ABC) algorithm was developed by Karaboga [31], it is a population-based metaheuristic algorithm inspired by the intelligent behavior of honey bees. It is an iterative algorithm that simulates the foraging behavior of bees, allowing them to search for nectar sources while simultaneously optimizing a problem. The algorithm works by assigning each solution a fitness value, which is then used to guide the search in the next iteration. This approach allows the algorithm to explore the entire search space in an efficient manner. The ABC algorithm is a relatively simple and efficient optimization technique that can be used to solve a variety of problems. The ABC colony is composed of three types of bees: the employed, the onlooker, and the scout. The employed bees are responsible for searching for food sources in the surrounding area of their hive and storing the related information in their memories. Onlooker bees, in turn, collect this information from the employed bees in the hive to identify potential food sources for further extraction of nectar. If a food source has a low quantity of nectar or has been exhausted, the scout bee will randomly search for a new food source in the search space. The algorithmic procedure of the ABC model follows these steps sequentially:

3.1.3.1 Initialization of the swarm

The first phase involves the initialization of N food sources in a random manner for the given problem dimension. The mathematical expression for this phase is represented by Eq. (3.1).

3.1.3.2 Employed bee phase

The second phase of the ABC algorithm is the employed phase. The phase consists of employed bees that searches for food sources (v_i^k) with more amount of nectar among the neighboring food sources (x_i^k). The generalized equation for this phase is given by Eq. (3.7).

$$v_i^t = x_i^t + \varphi(x_i^t - x_j^t) \tag{3.7}$$

Where, x_j^t is a random food source in the i^{th} direction, and φ is a random number generated in the range of [0, 1]. After generating a new food source v_i , its fitness is compared with respect to x_i. The fitness fit_i for the solution x_i corresponding to the $f_i(x_i)$ objective function is given by Eq. (3.8).

$$fit_i = \begin{cases} \frac{1}{1+f_i(x_i)} & if\ f_i(x_i) \geq 0 \\ 1 + |f_i(x_i)| & if\ f_i(x_i) < 0 \end{cases} \tag{3.8}$$

3.1.3.3 Onlooker bee phase

This phase is governed by the unemployed bees. These unemployed bees take information of food sources from the employed bees and choose the best food source for collecting nectar. Each food source is selected based on a certain probability p_m and this probability is chosen by using Eq. (3.9).

$$p_m = \frac{fit_i}{\sum fit_i} \tag{3.9}$$

After choosing a food source x_i , new neighbouring solutions are found by Eq. (3.7), and its fitness value is evaluated using greedy selection.

3.1.3.4 Greedy selection

In this phase, we find the best solution. This phase generally compares the current best solution with the previous best and choses the best among the two. This best solution is then updated over the course of iterations and after a certain set of iteration or until the stopping criteria is achieved, the final best solution is retained. The generalized equation for this phase is given by Eq. (3.10).

$$v_i^{t+1} = \begin{cases} v_i^t & if\ \ f(v_i) < \ f(x_i^t) \\ x_i^t & \quad otherwise \end{cases} \tag{3.10}$$

Where, v_i^{t+1} is the current iterative best solution, x_i^t is the previous best iterative solution and $f(x_i^t)$ is the fitness corresponding to x_i^t solution.

3.1.3.5 Scout bee phase

If a food source (v_i^k) cannot be further improved through a predetermined number of trials limit, the food source is assumed to be abandoned, and the corresponding employed bee becomes a scout. The scout produces a food source randomly using Eq. (3.1).

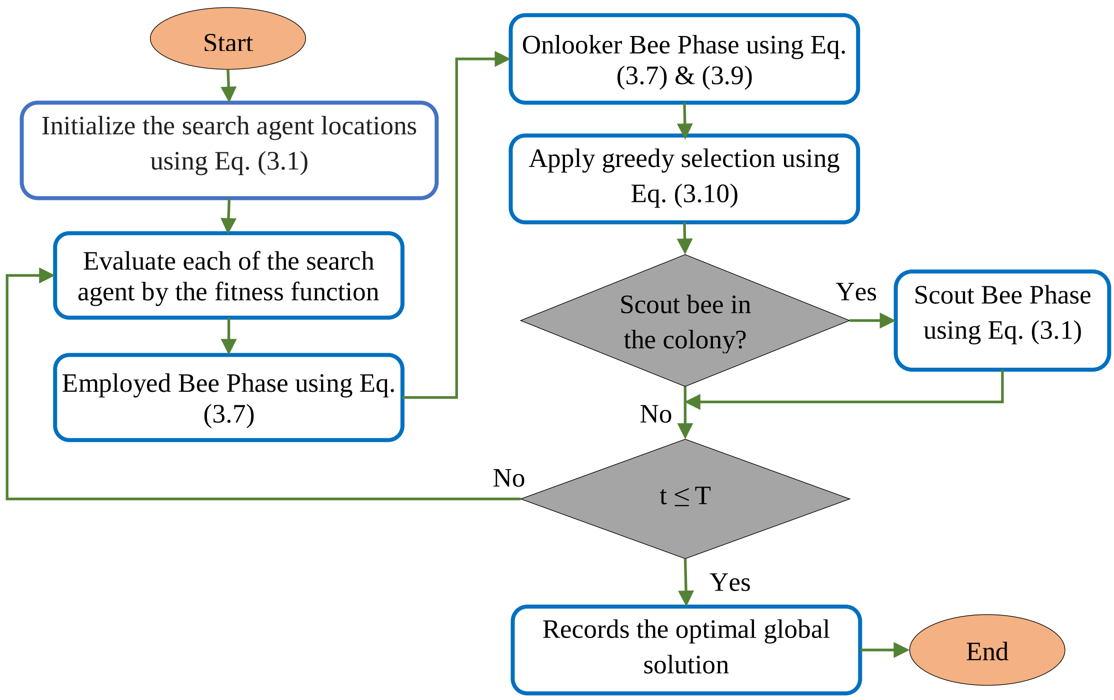

Figure 3.3: Flowchart of the Artificial Bee Colony (ABC) Algorithm.

3.1.4 Hybrid Artificial Bee Colony and JAYA Algorithm (JABC)

Artificial bee colony (ABC) is a population-based stochastic search technique inspired by the foraging behaviour of honey bees. The algorithm simulates the behaviour of three types of bees: employed, onlooker, and scout bees. Employed bees search for food sources, onlooker bees select promising food sources based on the dance of employed bees, and scout bees search for new food sources. The ABC algorithm solves optimization problems by iteratively adjusting the solutions represented by employed bees. The algorithm has been successfully applied to a wide range of optimization problems, including classification, clustering, and feature selection, among others. It is one among the most effective algorithms and this can be better understood from the fact that the basic ABC paper has received more than 8800 citations [31]. The suitability of the algorithm for large scale problems and high dimensions is still a matter of concern. Apart from that, it suffers from poor exploration, poor exploitation, and has degraded convergence patterns [40]. Many papers have been proposed to improve its performance and for application to specific problems. Apart from that, the no free lunch theorem states that no algorithm can be best fit for all optimization problems and new work must be done to make the algorithm perform better for specific research problems. So, the above said problems motivate the authors to propose new prospective algorithms for their problems. In this section, we provide details about the proposed JABC algorithm. Since JABC uses the basic structure of both ABC and JAYA, no explicit details on the basics of ABC and JAYA are presented in this paper. For implementation details of both of these algorithms, the readers can refer to [29, 39]. The aim of the proposed JABC algorithm is to mitigate poor exploitation and exploration, slow convergence speed, and unbalanced local and global search.

3.1.4.1 Employed bee phase

The first phase of the proposed algorithm is the employed phase and is similar to the basic ABC algorithm with added modifications. The generalized equation for this phase is given by Eq. (3.7)

In the present case, a simulated annealing based mutation operator is used to formulate new values of φ. The mathematical formulation for simulated annealing based φ is given by:

$$\varphi = \gamma_{min} + (\gamma_{max} - \gamma_{min}) \times r^{k-1} \tag{3.11}$$

Where, γ_{max} = 0.95, γ_{min}= 0.45, & k= rand [0,1] are the parameters of simulated annealing mutation operator, this mutation operator helps to provide better exploration operation, and helps in improved convergence patterns of the proposed algorithm [61]. After generating a new food source v_i , its fitness is compared with respect to x_i. The fitness fit_i for the solution x_i corresponding to the $f_i(x_i)$ objective function is given by Eq. (3.8)

3.1.4.2Onlooker bee phase

This phase is similar to the basic ABC algorithm with insert simulated annealing that added in the employed phase of the JABC algorithm. The equation for simulated annealing using φ is expressed mathematically in Eq. (3.11)

3.1.4.3 Scout bee phase based JAYA algorithm

Those unemployed bees who randomly select food sources are scouts. This phase is activated if the solution quality does not improve after a certain number of trials. If x_i is abandoned, the new solution becomes the employed bee and is generated by using JAYA algorithm [41]. The major reason for the use of JAYA algorithm in this phase is the parameter less nature of this algorithm. Apart from that, due to the movement of new solutions toward the best and away from the worst solution, the added modification helps the algorithm in local optima avoidance problem. The generalized equation for this phase is given by Eq. (3.6)

The overall movement of these steps helps the algorithm in local optima avoidance and hence provides better convergence. The pseudocode of JABC algorithm is given in Algorithm 1, the flowchart of the propose algorithm shown in Figure 3.4.

Algorithm 1 Pseudocode of JABC algorithm

1: **Begin**
2: Define the size of the population (N)
3: Define stopping criteria and problem dimension (Dim)
4: Initialize the population of solutions using Equation (3.1)
5: Set t = 1 as the current iteration and T as the maximum number of iterations
6: Perform the Employed Bee phase using Equations (3.7) and (3.11)
7: Perform the Onlooker Bee phase using Equations (3.7), (3.9), and (3.11)
8: Evaluate fitness and perform greedy selection using Equation (3.10)
9: Perform the Scout phase using Equation (3.6)
10: Update φ using Equation (3.11)
11: Close
12: Update the final best solution
13: **End**

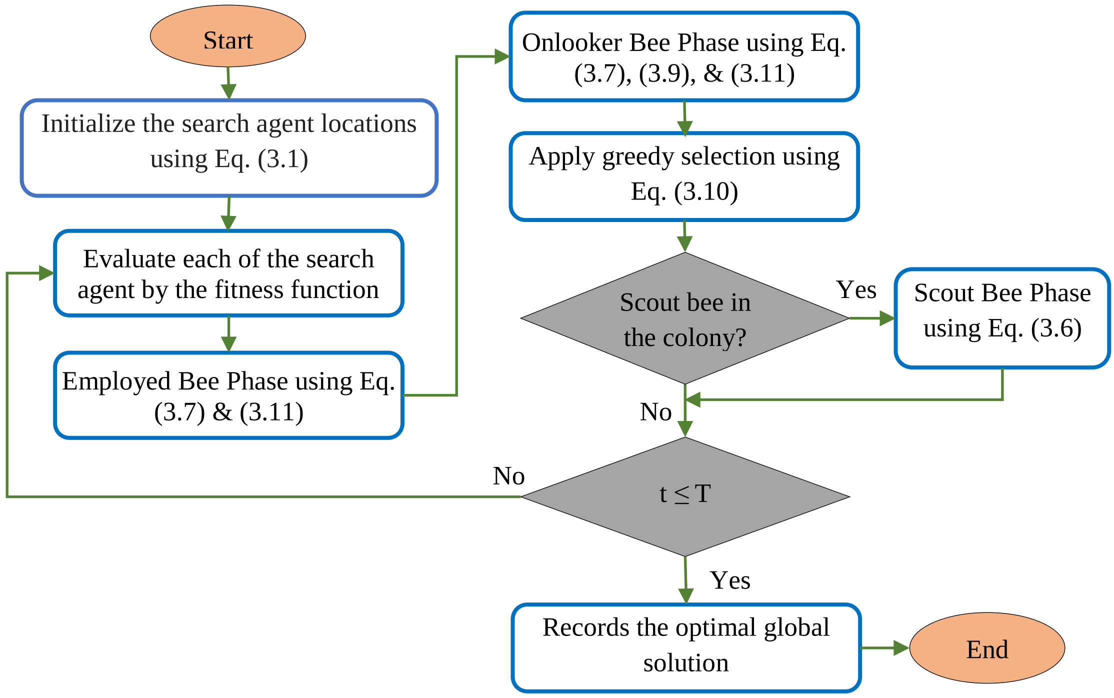

Figure 3.4: Flowchart of the hybrid Jaya and Artificial Bee Colony (JABC) Algorithm.

3.2 Performance evaluation for CEC 2005 test functions

The performance of JABC is tested on eight benchmark problems and a comparison with respect to ABC [32], Bat Algorithm (BA) [62], Firefly Algorithm (FA) [63], and BBO [64]. These algorithms are competitive and have been found to provide viable solutions for the problems under test. The description of these benchmarks is given in Table 3.3, and the parameters corresponding to each of the algorithms is given in Table 3.4. The results for comparison are presented in Table 3.5, and Table 3.6 gives the statistical results for each of the algorithms under comparison.

Table 3.3: Benchmark functions used in the simulation.

Test Problems	Objective Function	Search Range	Optimum Value	D
Hartmann function 3	$f_1(x) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{3} A_{ij}(x_j - P_{ij})^2]$	[0, 1]	−3.86278	3
Hartmann function 6	$f_2(x) = -\sum_{i=1}^{4} \alpha_i \exp[-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2]$	[0, 1]	−3.32237	6
Shekel function 5	$f_3(x) = -\sum_{j=1}^{5} [\sum_{i=1}^{4} ((x_i - C_{ij})^2 + \beta_j)^{-1}]$	[0, 10]	−10.1532	4
Shekel function 7	$f_4(x) = -\sum_{j=1}^{7} [\sum_{i=1}^{4} ((x_i - C_{ij})^2 + \beta_j)^{-1}]$	[0, 10]	−10.4029	4

Shekel function 10	$f_5(x) = -\sum_{j=1}^{10}[\sum_{i=1}^{4}((x_i - C_{ij})^2 + \beta_j)^{-1}]$	[0, 10]	−10.5364	4
Rastrigin function	$f_6(x) = 10D + \sum_{i=1}^{D}[x_i^2 - 10\cos(2\pi x_i)]$	[−5.12, 5.12]	0	30
Six Hump Camel function	$f_7(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	[−5, 5]	−1.0316	2
Goldstein & price function	$f_8(x) = (1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3{x_1}^2 - 14x_2 + 6x_1x_2 + 3{x_2}^2))(30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12{x_1}^2 + 48x_2 - 36x_1x_2 + 27{x_2}^2))$	[−2, 2]	3	2

Table 3.4: Parameter settings of various algorithms

Algorithm	Parameters	Values
FA	Number of fireflies	20
	Alpha (α)	0.25
	Beta (β)	0.20
	Gamma (γ)	1
	Maximum number of iterations	500
	Stopping Criteria	Max Iteration.
BA	Population size	20
	Loudness	0.5
	Pulse rate	0.5
	[Qmin, Qmax]	[0,1]
	Maximum number iterations	1000
	Stopping Criteria	Max Iteration.
ABC	Colony size (SN)	20
	Number of food sources	SN/2
	Limit	100
	Maximum number iterations	500
	Stopping Criteria	Max Iteration.
BBO	Population Size	20
	Mutation probability	0.25
	Habitat modification probability	1
	Maximum number of iterations	500
	Stopping Criteria	Max. Iteration.
JABC	Colony size (SN)	20
	Number of food sources	SN/2
	Limit	100
	Maximum number of iterations	500
	Stopping Criteria	Max Iteration.

3.2.1 Experimental results

From the results in Table 3.5, best values are shown in the bold text. For functions, f_2, f_5, f_6, f_7 and f_8 the standard deviation of JABC is much better except for f_1 in which FA is better, f_3 where ABC is better and f_4 where BA is better. Mean for seven function is better except for only f_2 and f_3 where FA is better. As far as best value is concerned, JABC gives best for most of the test function except for f_6 where BA is better. The results show that JABC algorithm performs better than ABC, BBO, BA and FA for most of the test functions. The proposed algorithm is also able to achieve better mean and standard deviation values than competing algorithms.

Table 3.5: Simulation Results

Objective Function	Algorithm	Best	Worst	Mean	Standard Deviation
$f_1(x)$	ABC	−3.77541	−2.41101	−3.23971	4.151E−01
	BBO	−3.22341	−0.00242	−0.96732	9.551E−01
	BA	−3.86282	−3.08983	-3.78554	0.23793
	FA	**−3.86283**	**−3.86284**	**-3.86284**	**3.312E-007**
	JABC	**−3.86284**	−3.79512	−3.85485	1.682E−02
$f_2(x)$	ABC	−2.19631	−0.72901	−1.38161	4.643E−01
	BBO	−3.14523	−1.90592	−2.75012	3.023E−01
	BA	**−3.32243**	**−3.20313**	−3.26273	0.06121
	FA	**−3.32221**	−3.19153	**−3.26724**	0.06262
	JABC	**−3.32242**	−3.09414	−3.25145	**8.673E−03**
$f_3(x)$	ABC	−10.10731	−2.59283	−6.61141	**3.09561**
	BBO	**−10.15322**	−2.63042	−6.14442	3.47912
	BA	−10.15251	**−2.63051**	−5.01863	3.18792
	FA	−10.15281	−2.63042	−6.78843	3.81552
	JABC	**−10.15322**	**−2.63054**	**−7.90952**	3.51641
$f_4(x)$	ABC	−10.50543	−1.66804	−5.90551	3.22573
	BBO	−10.40284	**−2.76595**	−7.60972	3.54632
	BA	−10.40293	−1.83763	−4.02643	**2.49083**
	FA	−10.40282	−2.75192	−9.25424	2.80251
	JABC	**−10.40291**	−2.75191	**−9.25673**	2.79952
$f_5(x)$	ABC	−10.46422	−1.85082	−5.35521	3.42953
	BBO	−10.53631	−2.80663	−7.32432	3.65851
	BA	**−10.53642**	−1.67662	−4.19373	3.30052
	FA	−10.53623	−10.53471	−10.53554	4.85E-004
	JABC	**−10.53644**	**−10.53642**	**−10.53645**	**7.292E−06**
$f_6(x)$	ABC	4.22E+01	9.20E+01	6.76E+01	1.371E+01
	BBO	9.67411	2.20E+01	1.95E+01	3.104249
	BA	**8.10E-09**	12.92344	4.07932	3.194044
	FA	3.63E-06	1.10E-04	4.06E-05	3.222E-05
	JABC	1.31E-08	**2.49E-07**	**1.01E-07**	**8.553E-08**
$f_7(x)$	ABC	**−1.03161**	−1.02611	−1.03052	1.501E−03
	BBO	−1.02342	−0.04792	−0.73143	3.452E−01
	BA	**−1.03163**	−0.21553	−0.78683	0.38372
	FA	**−1.03163**	**−1.03162**	**−1.03163**	1.221£-006
	JABC	**−1.03161**	**−1.03162**	**−1.03161**	**4.582E−09**
$f_8(x)$	ABC	3.00032	3.09043	3.01902	2.523E−02
	BBO	**3.00002**	**3.00003**	**3.00003**	**0.00E+00**
	BA	**3.00003**	84.00004	16.50002	25.53942
	FA	**3.00004**	**3.00005**	**3.00001**	1.482E-05
	JABC	**3.00000**	**3.00000**	**3.00000**	2.442E−08

3.2.2 Statistical testing

Wilcoxon's rank-sum test [65] and Friedman rank (f-rank) [66] are used to validate the applicability of the proposed JABC statistically. Wilcoxon's ranksum test is done to provide details of results in terms of *win(w), loss(l)* and *tie(t)*. Here *w* given as " + " means that the proposed algorithm is better than the algorithm under comparison, *l* given by " − " means the proposed algorithm does not provide better results than the test algorithm, and *t* given by " = " stands for equality in results. The results in Table 3.6 shows that for most of the cases, our proposed JABC is better and significant with respect to others.

Table 3.6: Wilcoxon's ranksum and Freidman test results.

Test function		Algorithm				
		ABC	BBO	BA	FA	JABC
$f_1(x)$	p-rank	-	-	-	+	N/A
	f-rank	4	5	3	1	2
$f_2(x)$	p-rank	-	-	-	-	N/A
	f-rank	5	4	3	2	1
$f_3(x)$	p-rank	-	-	-	-	N/A
	f-rank	5	2	4	3	1
$f_4(x)$	p-rank	-	-	+	-	N/A
	f-rank	5	4	1	3	2
$f_5(x)$	p-rank	-	-	-	-	N/A
	f-rank	4	5	3	2	1
$f_6(x)$	p-rank	-	-	-	-	N/A
	f-rank	5	4	3	2	1
$f_7(x)$	p-rank	-	-	-	-	N/A
	f-rank	3	4	5	2	1
$f_8(x)$	p-rank	-	+	-	-	N/A
	f-rank	4	1	5	3	2
w/l/t		0/8/0	1/7/0	1/7/0	1/7/0	
overall f-rank		35	29	27	18	10
Average f-rank		5	4	3	2	1

3.3 Synthesis of Linear Antenna Arrays (LAAs)

A linear antenna array (LAA) refers to a configuration of antenna elements arranged in a straight line. The arrangement typically consists of (2N) elements symmetrically positioned along the line, as illustrated in Figure 3.5. The popularity of LAA stems from its straightforward implementation and simplicity, which have led to extensive investigations in various research works. The practicality and versatility of LAA make it a compelling choice for many applications, as it offers ease of deployment and efficient signal propagation characteristics.

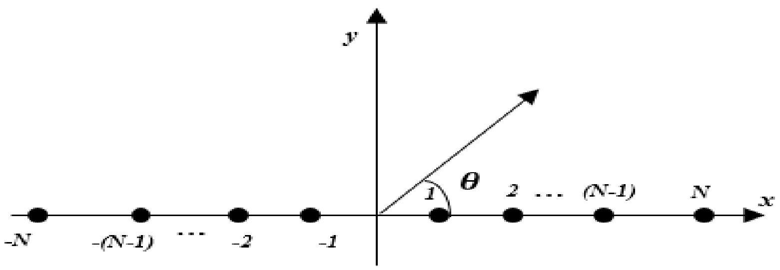

Figure 3.5: Geometry of a systematic LAA.

3.3.1 Problem formulation

The array factor for the geometry with odd and even numbers of elements is given by Eq. (3.12) and (3.13), respectively [14].

$$AF\ (\theta) = I_o exp(j\theta_0)\ + 2\sum_{n=-N,n\neq 0}^{N} I_n\ exp(j\ [kx_n cos\ (\theta) + \alpha_n] \tag{3.12}$$

$$AF\ (\theta) = 2\sum_{n=-N,n\neq 0}^{N} I_n\ exp(j\ [kx_n cos\ (\theta) + \alpha_n] \tag{3.13}$$

Where α_n and I_n are the excitation phase and amplitude, respectively, for the n^{th}element feeding current, $x_n = \sum_{i=1}^{n} d_i$ is the position and d_i is inter-element spacing, having k = 2π/λ as the wave number. After solving the above equations for $\alpha_n = \alpha_{-n}$, $x_n = x_{-n}$ & $I_n = I_{-n}$, the array factor becomes:

$$AF\ (\theta) = I_o exp(j\theta_0)\ + 2\sum_{n=1}^{N} I_n\ cos(j\ [kx_n cos\ (\theta) + \alpha_n] \tag{3.14}$$

$$AF\ (\theta) = 2\sum_{n=1}^{N} I_n\ cos(j\ [kx_n cos\ (\theta) + \alpha_n] \tag{3.15}$$

The array variables (α_n, I_n, and x_n) are optimized in the present case, by suppressing the side lobe level (SLL). The fitness function for minimization is given by [14]:

$$\text{Fitness function} = min\left[max\ \left(20\ log\frac{|AF(\Phi)|}{max|AF(\theta)|}\right)\right] \tag{3.16}$$

The side lobe region of a linear antenna array (LAA) is typically defined within the range of [0, Φ], where Φ depends on the number of elements in the array. In this study, specific values have been chosen to represent the side lobe region for different LAA configurations. For instance, the range [0, 74°] is selected for the 10-element array, [0, 80°] for the 16-element array, [0, 83°] for the 24-element array, [0, 84°] for the 28-element array, and [0, 87°] for the 40-element array. These ranges correspond to the desired side lobe characteristics for each respective array configuration.

In Figure 3.6, a simplified flowchart is presented to illustrate the process of synthesizing a linear antenna array using optimization algorithms. The flowchart outlines the key steps involved in the synthesis procedure. It provides a visual representation of the sequence of actions, from defining objectives and constraints to evaluating performance metrics. By following this flowchart, one can navigate through the synthesis process efficiently and effectively, ultimately achieving an optimized linear antenna array configuration.

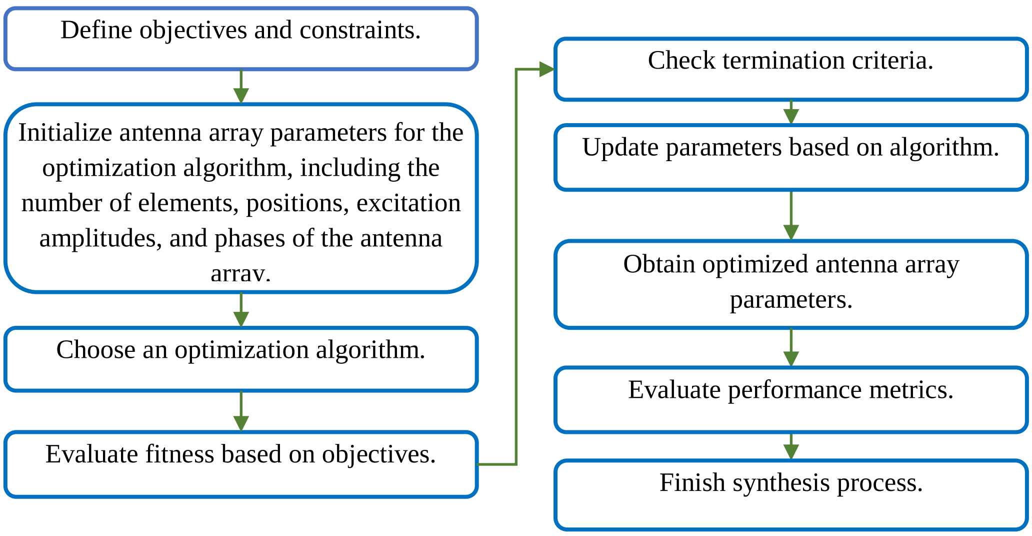

Figure 3.6: Flowchart for synthesizing a linear antenna array using optimization algorithms.

Chapter 4

RESULTS AND DISCUSSION

This chapter deals with the analysis of the proposed JABC algorithm with respect to synthesis of linear antenna array problems. The whole chapter provides extensive results on the synthesis of LAA using proposed algorithms for reduction peak sidelobe levels (SLLs) of linear antenna arrays (LAA), and maintain first null beam width (FNBW) constant, which is achieved by optimizing excitation amplitude, phases, and positions array elements parameters of LAA.

All the simulations are performed on a Windows 7, 64-bit operating system with 8GB RAM, Intel Core i3 processor, and MATLAB 2022a.

4.1 Synthesis of Linear Antenna Arrays (LAAs)

This section provides extensive results of the proposed algorithm for the synthesis of LAA. The number of elements used in this chapter are 10-element, 16-element, 20-element, 24-element, 26-element, 28 element, 30-element, and 40-element LAA for optimization.

Table 4.1 outlines the key parameters used in the optimization of linear antenna array (LAA) synthesis using various algorithms: hybrid Jaya and Artificial Bee Colony (JABC), Artificial Bee Colony (ABC), JAYA, and Sine Cosine Algorithm (SCA).

Table 4.1: Parameter settings of various algorithms.

Algorithm	Parameters	Values
ABC	Colony size (SN)	50
	Number of food sources	SN/2
	Limit	100
	Maximum number iterations	150-1000
	Stopping Criteria	Max Iteration.
SCA	Population Size	50
	Parameter ‘a’	0.15
	Maximum number of iterations	150-1000
	Stopping Criteria	Max. Iteration.

JAYA	Population Size Maximum number of iterations Stopping Criteria	50 150-1000 Max. Iteration.
JABC	Colony size (SN) Number of food sources Limit Maximum number of iterations Stopping Criteria	50 SN/2 100 150-1000 Max Iteration.

Table 4.2 presents the parameter values used for the optimization of Linear Antenna Array (LAA). Various parameters have been considered to optimize the performance of LAA, including the number of elements, element spacing, amplitude weighting, phase shifting, and main lobe steering angle.

Table 4.2: Parameter Values for Linear Antenna Array (LAA)

Parameters	**Values**
Number of elements (N)	10, 16, 20, 24, 26, 28, 30, 40
Element phase (α_n)	0-180
Element position (x_n)	$\frac{0.25\lambda}{2} < x_n < (N - \frac{1}{2})\lambda$
Element amplitude (I_n)	0-1
Steering angles (degree)	0, 30, 45, 60

4.1.1 Optimization of element amplitude (I_n) without FNBW constraints

In this section, by selecting the ideal amplitude for each component of the LAA, the optimization process' primary goal is to obtain the lowest peak SLL in the radiation pattern. The uniform array's fixed parameters are used, i.e. α_n= 0, and the distance between the elements is equal to λ/2. The amplitudes will initially take values between (0,1). For optimizing amplitude, Eq. (4.1) & (4.2) are used for even and odd elements, respectively.

$$AF(\theta)=2\sum_{n=1}^{N} I_n \cos\left[(n-0.5)\pi\cos(\theta)\right] \tag{4.1}$$

$$AF(\theta)=I_o + 2\sum_{n=1}^{N} I_n \cos\left[n\pi\cos(\theta)\right] \tag{4.2}$$

In this case, 10-element, 16-element, and 24-element LAAs have been used. A comparison with some of the recent algorithms is performed.

4.1.1.1 Case 1: 10 Elements LAA

This case shows optimizing amplitudes of 10 elements. Table 4.3 compares normalized optimization amplitudes obtained with a uniform linear array before optimization to the results obtained with the JABC, ABC, JAYA, SCA, and other optimization algorithms. Table 4.4 shows the performance of the suggested algorithms over 20 runs compared to other algorithms. Figures 4.1 & 4.2 show the azimuth radiation pattern and the convergence curve over 150 iterations of JABC, ABC, JAYA, and SCA algorithms, respectively. The box-and-whisker plots over 20 independent runs of the proposed methods are presented in Figure 4.3.

Table 4.3: Optimized amplitude for a 10-element LAA obtained with suggested algorithms compared to other techniques without constraint of FNBW.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \ldots, I_5$	Peak SLL (dB)	FNBW
JABC	1.0000, 0.8824, 0.6798, 0.4449, 0.2808	-28.94	34.42°
ABC	1.0000, 0.8817, 0.6797, 0.4455, 0.2793	-28.89	34.42°
JAYA	1.0000, 0.8836, 0.6822, 0.4369, 0.2894	-27.78	34.42°
SCA	1.0000, 0.8824, 0.6800, 0.4452, 0.2792	-28.88	34.59°
GOA [14]	1.0000, 0.8892, 0.6962, 0.4684, 0.3208	-27.36	33.08°
MFO [49]	1.0000, 0.8962, 0.6966, 0.4935, 0.2965	-26.07	32.59°
MA [52]	1.0000, 0.8922, 0.7036, 0.4791, 0.3400	-26.70	32.59°
MSOA [27]	1.0000, 0.8887, 0.6944, 0.4657, 0.3154	-27.52	33.59°
CSA [59]	0.9992, 0.8901, 0.6996, 0.4738, 0.3312	-26.99	32.59°
ALO [15]	1.0000, 0.8959, 0.6957, 0.4935, 0.2966	-26.08	32.96°
FPA [16]	1.0000, 0.8979, 0.7178, 0.5002, 0.3833	-25.33	31.47°
SOS [17]	1.0000, 0.8985, 0.7189, 0.5017, 0.3856	-25.28	31.40°
BBO [18]	1.0000, 0.8988, 0.7189, 0.5025, 0.3862	-25.21	31.40°
PSO [19]	1.0000, 0.9010, 0.7255, 0.5120, 0.4088	-24.62	30.80°
TOM [20]	1.0000, 0.8999, 0.7228, 0.5077, 0.3994	-24.87	31.20°
CS [21]	1.0000, 0.9019, 0.7273, 0.5153, 0.4157	-24.43	30.80°
Uniform	1.0000, 1.0000, 1.0000, 1.0000, 1.0000	-12.97	23.00°

Table 4.4: Performance of JABC algorithm with 10 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Worst SLL (dB)	Mean (dB)	STD (dB)
JABC	-28.9401	-28.9402	-28.9402	0
ABC	-28.8923	-28.8305	-28.7213	0.0470
JAYA	-27.8181	-27.3687	-27.0247	0.2764
SCA	-28.8817	-28.6610	-28.7839	0.0908
GOA [14]	-27.3600	-27.2200	-27.3200	0.0528
FPA [16]	-25.3300	-25.3000	-25.3100	0.0630
SOS [17]	-25.2791	-25.1842	-25.2645	0.0216
BBO [18]	-25.2100	-24.0763	-24.9704	0.2596

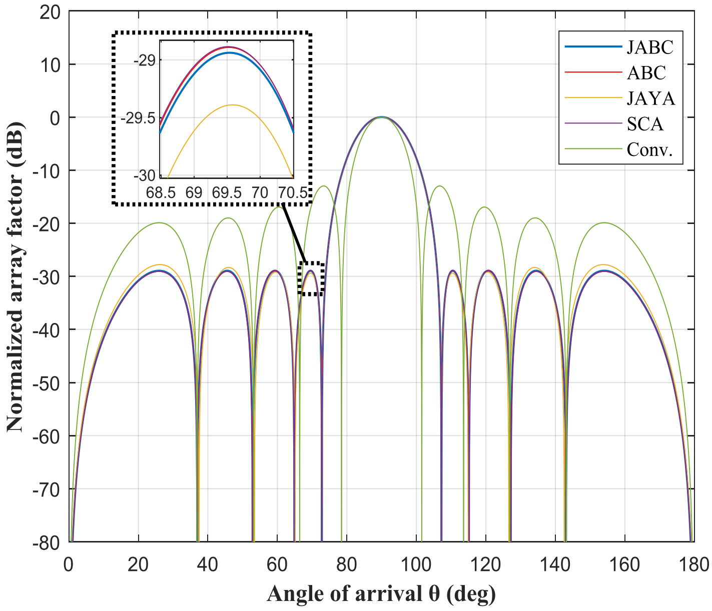

Figure 4.1: Radiation pattern obtained by suggested algorithms for 10-element linear array synthesis without FNBW constraint.

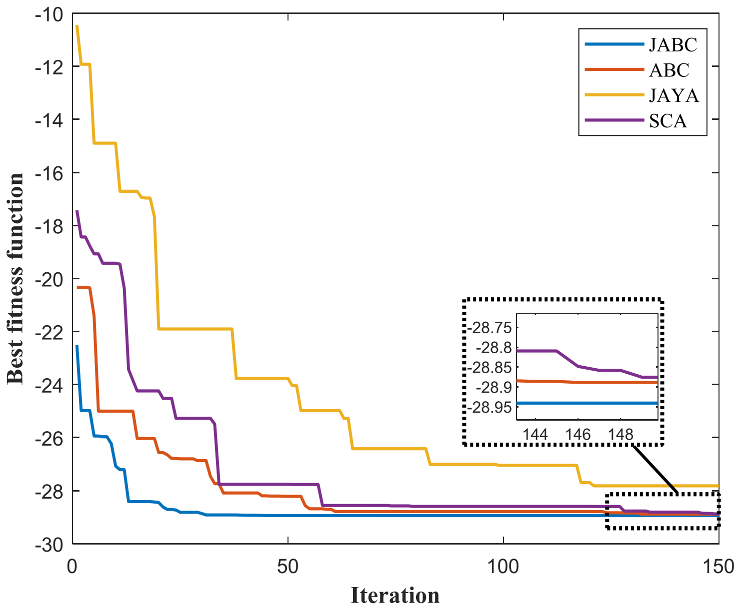

Figure 4.2: Convergence curve obtained by suggested algorithms for 10-element LAA over 150 iterations without FNBW constraint.

As can be observed from Table 4.3, JABC achieved a minimum peak SLL of -28.94 dB, as compared to other techniques. It can be observed from Table 4.4 also that the Standard deviation (STD) for JABC algorithm is lower than the STD for ABC, JAYA, SCA, GOA, FPA, and SOS, which demonstrates the accuracy and robustness of the JABC technique.

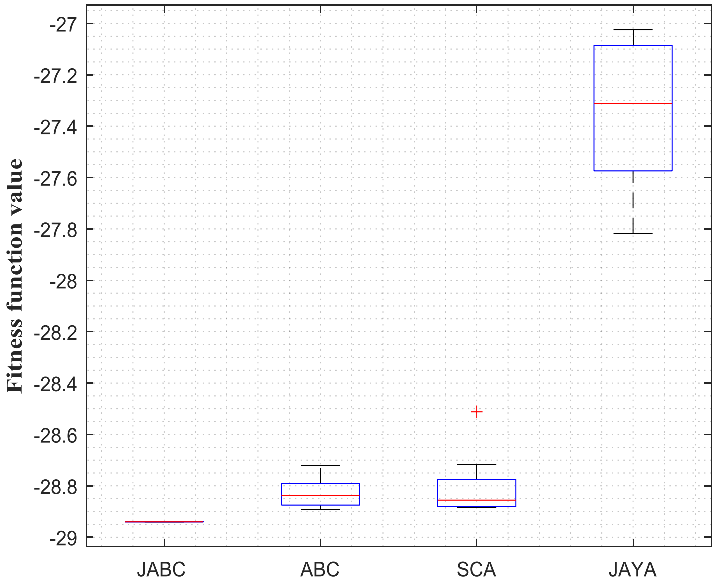

Figure 4.3: Box and whisker plot of 10-element LAA in 20 runs without FNBW constraint.

4.1.1.2 Case 2: 16 Elements LAA

Using the equation of fitness function associated with the array factor for 16 elements linear array, the proposed algorithm's code was executed for 20 independent trials. Table 4.5 and Figure 4.4 show the best optimum amplitudes and the radiation pattern obtained using the JABC, SCA, ABC, and JAYA, respectively. Figure 4.5 show the convergence curve over 150 iterations of JABC, ABC, JAYA, and SCA algorithms, respectively. Table 4.6 and Figure 4.6 show the performance of the suggested algorithms and the box-and-whisker plots over 20 runs compared to other algorithms, respectively.

Table 4.5: Optimized amplitude for a 16-element LAA obtained with suggested algorithms compared to other techniques without constraint of FNBW.

Evolutionary algorithm	**Optimized element amplitudes** $I_1, I_2, \ldots, I_8$	**Peak SLL (dB)**	**FNBW**
JABC	1.0000, 0.9399, 0.8287, 0.6820, 0.5197, 0.3613, 0.2234, 0.1447	-37.30	25.17°
ABC	1.0000, 0.9601, 0.8339, 0.6798, 0.5212, 0.3701, 0.2192, 0.1366	-36.56	25.17°
SCA	1.0000, 0.9609, 0.8238, 0.6838, 0.5199, 0.3595, 0.2290, 0.1216	-36.09	25.57°
JAYA	1.0000, 0.9521, 0.8037, 0.6826, 0.5170, 0.3537, 0.2208, 0.1201	-35.34	24.98°
EFA [22]	1.0000, 0.9464, 0.8460, 0.7118, 0.5593, 0.4061, 0.2667, 0.2038	-33.62	23.6°
BFP [23]	1.0000, 0.9464, 0.8459, 0.7119, 0.5594, 0.4060, 0.2667, 0.2037	-33.62	23.6°

SOS [17]	1.0000, 0.9466, 0.8475, 0.7137, 0.5624, 0.4094, 0.2697, 0.2088	-33.39	23.2°
Hybrid [25]	1.0000, 0.9468, 0.8474, 0.7139, 0.5623, 0.4093, 0.2699, 0.2087	-33.36	23.2°
ALO [25]	1.0000, 0.9466, 0.8475, 0.7136, 0.5624, 0.4093, 0.2698, 0.2088	-33.36	23.2°
GOA [25]	1.0000, 0.9466, 0.8475, 0.7136, 0.5625, 0.4093, 0.2699, 0.2088	-33.35	23.2°
BBO [18]	1.0000, 0.9402, 0.8487, 0.7104, 0.5596, 0.4115, 0.2697, 0.2035	-33.06	23.2°
TOM [20]	1.0000, 0.9500, 0.8575, 0.7317, 0.5861, 0.4381, 0.2988, 0.2552	-31.21	22.2°
SADE [20]	1.0000, 0.9515, 0.8586, 0.7333, 0.5889, 0.4404, 0.3020, 0.2616	-31.01	22°
Cheby [19]	1.0000, 0.9515, 0.8602, 0.7364, 0.5933, 0.4457, 0.3069, 0.2713	-30.70	21.8°
PSO [19]	1.0000, 0.9521, 0.8605, 0.7372, 0.5940, 0.4465, 0.3079, 0.2724	-30.63	22°
IWO [2]	1.0000, 0.9760, 0.9310, 0.7930, 0.6600, 0.6440, 0.4000, 0.4090	-26.39	18.78°
TS [24]	1.0000, 0.9627, 0.8766, 0.7560, 0.6105, 0.4833, 0.2957, 0.3426	-26.18	21°
MSMO [57]	1.0000, 0.9613, 0.7249, 0.8346, 0.5556, 0.3977, 0.2842, 0.1844	-25.94	22.4°
CABMO [67]	1.0000, 0.8080, 0.6410, 0.6200, 0.6610, 0.4840, 0.3660, 0.3010	-25.87	19.49°
CS [23]	1.0000, 0.8660, 0.7910, 0.8010, 0.5670, 0.3660, 0.3530, 0.3360	-25.01	21.25°
FA [23]	1.0000, 0.9070, 0.8800, 0.7530, 0.5960, 0.5000, 0.3660, 0.3970	-24.27	19.91°
Uniform	1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000	-13.15	14.54°

Table 4.6: Performance of JABC algorithm with 16 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Worst SLL (dB)	Mean (dB)	STD (dB)
JABC	-37.3013	-37.2602	-37.3060	0.0116
ABC	-36.5621	-34.0377	-35.3222	0.7009
JAYA	-35.3492	-33.5699	-34.8861	0.6134
SCA	-36.0919	-34.7195	-35.3932	0.5394
SOS [17]	-33.3914	-32.9152	-33.3418	0.0216
BBO [18]	-33.0600	-29.5565	-32.0106	0.2596
IWO [2]	-26.5700	-25.3500	-26.4100	0.0550
CABMO [67]	-25.8700	-	-	0.5347
CS [23]	-26.0800	-25.0100	-25.2800	0.2293
FA [23]	-25.3400	-24.6100	-24.6100	0.3180

JABC obtained a maximum SLL of −37.30 dB, which is far better compared to EFA, BFP, SOS, ABC, and JAYA, among others. Again, this case shows that JABC gives results that are as good as those obtained by other well-known optimization techniques.

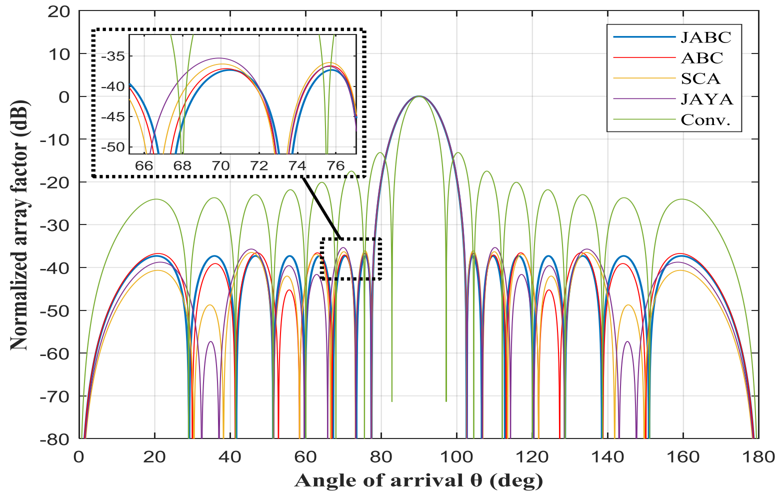

Figure 4.4: Radiation pattern obtained by suggested algorithms for 16-element linear array synthesis without FNBW constraint.

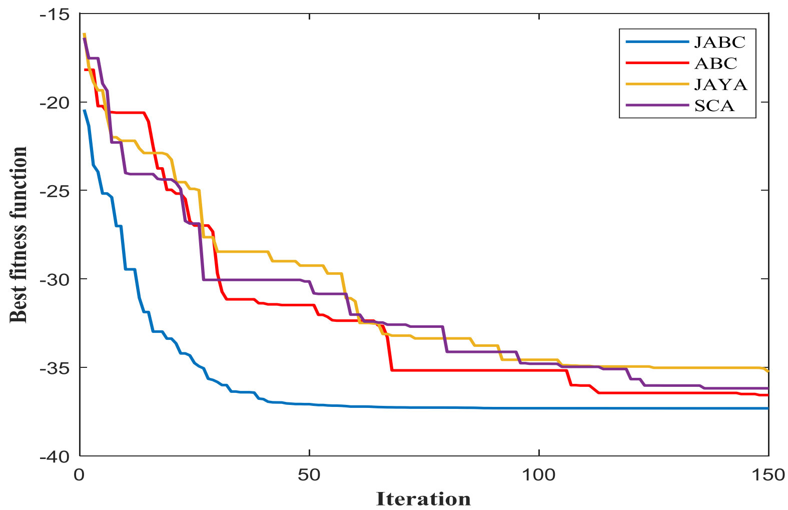

Figure 4.5: Convergence curve obtained by suggested algorithms for 16-element LAA over 150 iterations without FNBW constraint.

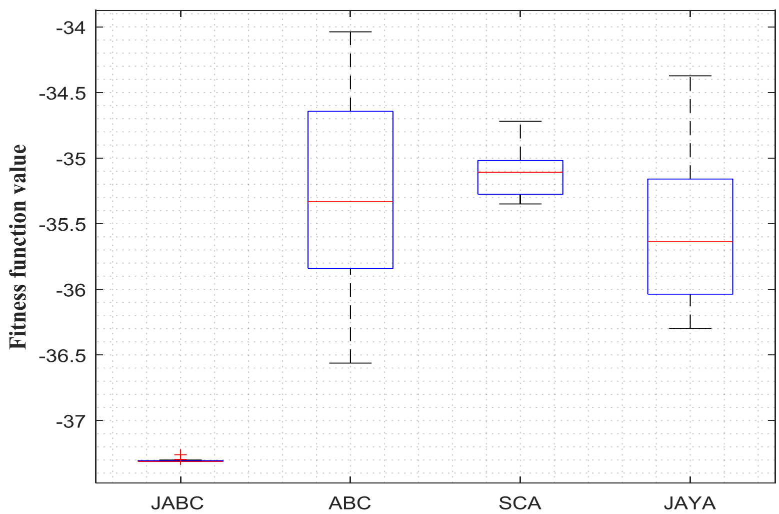

Figure 4.6: Box and whisker plot of 16-element LAA in 20 runs without FNBW constraint.

4.1.1.3 Case 3: 24 Elements LAA

The third case uses proposed algorithms to reduce peak SLL of 24 element LAA. Table 4.7 shows the peak SLL, FNBW, and optimum element amplitudes determined using the suggested algorithms, while Table 4.8 shows the effectiveness of the JABC algorithm and others over 20 runs. Figure 4.8 and Figure 4.9 show the convergence curve over 500 iterations of the suggested algorithms, and the box-and-whisker plots over 20 runs compared to other algorithms, respectively.

Table 4.7: Optimized amplitude for a 24-element LAA obtained with suggested algorithms compared to other techniques without constraint of FNBW.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \ldots, I_{12}$	Peak SLL (dB)	FNBW
JABC	1.0000, 0.9642, 0.8958, 0.8011, 0.6879, 0.5655,0.4439, 0.3288, 0.2302, 0.1468, 0.0865, 0.0502	-49.09	20.55°
ABC	1.0000, 0.9718, 0.9061, 0.8287, 0.7193, 0.6180, 0.4981, 0.3911, 0.2851, 0.1884, 0.1240, 0.0932	-41.66	18.55°
JAYA	1.0000, 0.9429, 0.8744, 0.8112, 0.7299, 0.5544, 0.4148, 0.3445, 0.2173, 0.1673, 0.0991, 0.0590	-36.53	20.55°
SCA	1.0000, 0.9777, 0.8957, 0.7955, 0.6880, 0.5567, 0.4328, 0.3062, 0.2239, 0.1313, 0.0816, 0.0272	-45.93	20.55°
SOS [17]	1.0000, 0.9699, 0.9143, 0.8387, 0.7420, 0.6368,0.5273, 0.4145, 0.3149, 0.2243, 0.1515, 0.1236	-39.37	17.54°
CSA [59]	0.9968, 0.9737, 0.9062, 0.8395, 0.7276, 0.6332, 0.5093, 0.4048, 0.3031, 0.2076, 0.1449, 0.1052	-40.90	17.54°
BBO [68]	0.9796, 1.0000, 0.9011, 0.8581, 0.7375, 0.6103,0.5205, 0.4463, 0.3016, 0.2236, 0.1495, 0.0957	-37.14	17.54°
TOM [20]	1.0000, 0.9731, 0.9283, 0.8585, 0.7745, 0.6758,0.5772, 0.4686, 0.3719, 0.2764, 0.1995, 0.2026	-35.02	15.54°
PSO [19]	1.0000, 0.9712, 0.9226, 0.8591, 0.7812, 0.6807,0.5751, 0.4768, 0.3793, 0.2878, 0.2020, 0.2167	-34.46	15.54°
Uniform	1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000,1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000	-13.18	9.53°

Table 4.8: Performance of JABC algorithm with 24 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Worst SLL (dB)	Mean (dB)	STD (dB)
JABC	**-49.1068**	-47.8123	-45.8123	1.4131
ABC	-41.6604	-31.0435	-36.5684	3.0380
JAYA	-36.5324	-27.8377	-29.7476	2.1971
SCA	-45.0911	-33.1479	-37.1126	2.8600

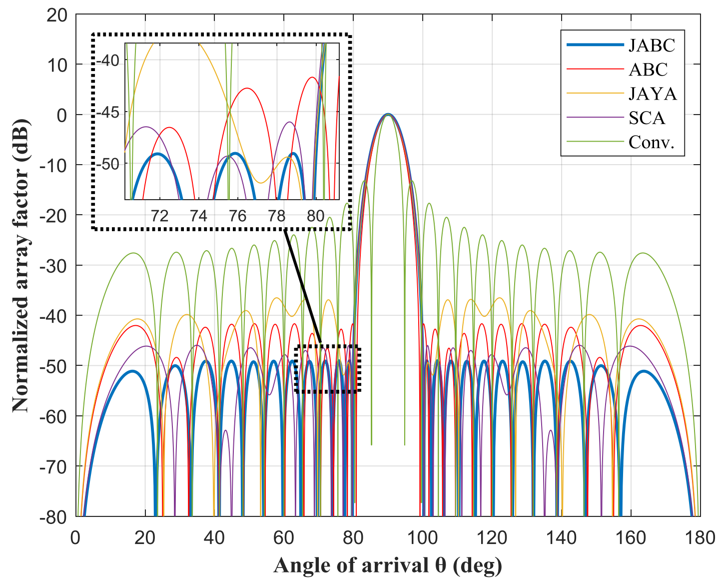

Figure 4.7: Radiation pattern obtained by suggested algorithms for 24-element linear array synthesis without FNBW constraint.

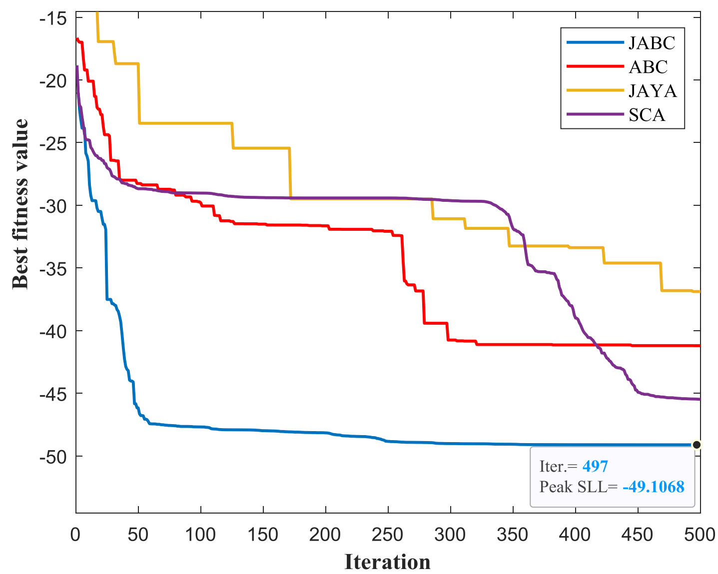

Figure 4.8: Convergence curve obtained by suggested algorithms for 24-element LAA over 500 iterations without FNBW constraint.

According to the results in Table 4.7, the peak SLL achieved by JABC has been minimized from -13.18 dB to -9.09 dB as compared to uniform array, which is 7.43 dB, 12.56 dB, 12.56 dB, 3.16 dB, 9.72dB, 11.95 dB, 14.07 dB, and 14.63 dB less than ABC, JAYA, SCA, SOS [17], BBO [68], TOM [20], and PSO [19] algorithms, respectively.

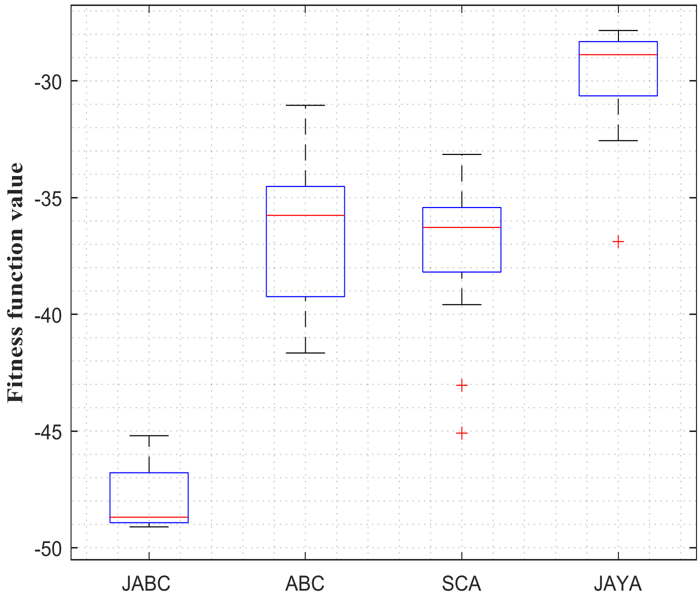

Figure 4.9: Box and whisker plot of 24-element LAA in 20 runs without FNBW constraint.

4.1.2 Optimization of element amplitude (I_n) with FNBW constraints

As observed from section 4.1.1, that decreased SLL comes at the trade-off of wide beam width. As a result, it can be stated that arrays with narrow beam widths do not provide smaller SLL. To have a fair comparison between the optimized array and the uniform array, the first null beam width (FNBW) of the optimized array should be the same as that of the uniform array. This can be achieved by fixing the outermost element's amplitudes at $I_{\pm N} = 1$ and optimizing the other element's amplitudes. In this case, the array factor becomes as Eq. (4.3) & (4.4) for even and odd elements, respectively.

$$\mathrm{AF}\ (\theta) = 2cos[(N - 0.5)\pi cos\ (\theta)] + 2\sum_{n=1}^{N-1} I_n\ cos\ [(n - 0.5)\pi cos\ (\theta)] \tag{4.3}$$

$$\mathrm{AF}\ (\theta) = I_o + 2cos[N\pi cos\ (\theta)] + 2\sum_{n=1}^{N-1} I_n\ cos\ [n\pi cos\ (\theta)] \tag{4.4}$$

4.1.2.1 Case 4: 10 Elements LAA

For a 10-element LAA with FNBW at 23°, the optimum values for each of the excitation phases is given in Table 4.9. We can see that the SLL values for all the cases, including BBO, JAYA, ABC, DE, and JABC is almost similar and there is not much difference and is close to -15.96 to -15.99dB for all the cases. But the radiation patterns and convergence curves in Figure 4.10 & 4.11 show that JABC performs better and this is due to better convergence speed of the JABC algorithm.

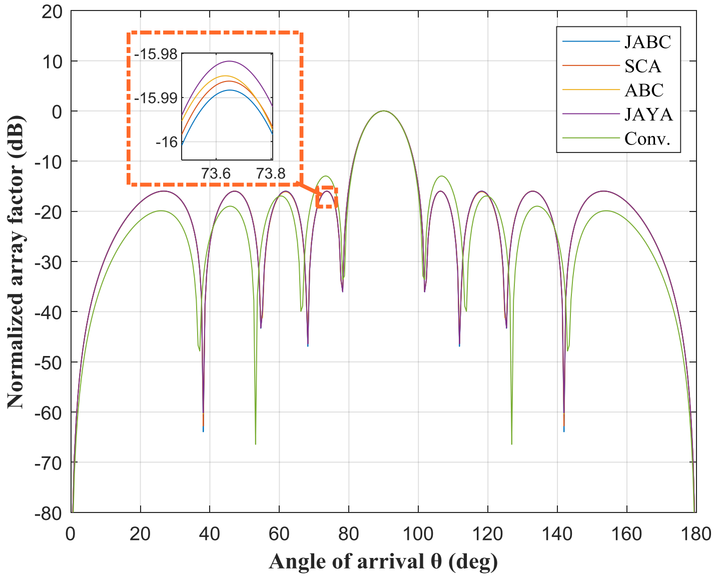

Figure 4.10: Radiation pattern obtained by suggested algorithms for 10-element linear array synthesis with FNBW constraint.

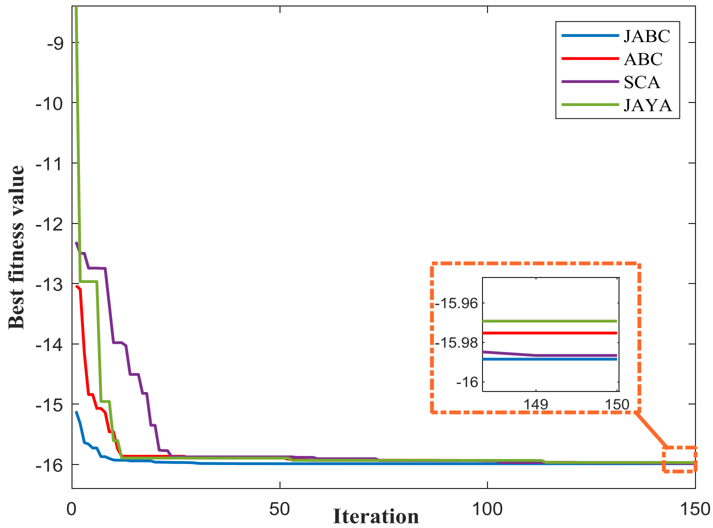

Figure 4.11: Convergence curve obtained by suggested algorithms for 10-element LAA over 150 iterations with FNBW constraint.

Table 4.9: Optimized amplitude for 10-element LAA obtained with suggested algorithms compared to other techniques with FNBW constraint.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \dots, I_5$	Peak SLL (dB)	FNBW
JABC	1.0000, 0.9391, 0.8255, 0.6740, 1.0000	-15.99	23°
ABC	1.0000, 0.9448, 0.8317, 0.6739, 1.0000	-15.98	23°
JAYA	1.0000, 0.9389, 0.8268, 0.6739, 1.0000	-15.97	23°
SCA	1.0000, 0.9400, 0.8267, 0.6740, 1.0000	-15.99	23°
CS [21]	1.0000, 0.9392, 0.8257, 0.6741, 1.0000	-15.99	23°
BBO [18]	1.0000, 0.9382, 0.8258, 0.6733, 1.0000	-15.97	23°
CFO [69]	1.0000, 0.9690, 0.8590, 0.6760, 1.0000	-15.93	23°
DE [69]	1.0000, 0.9390, 0.8270, 0.6760, 1.0000	15.96	23°

4.1.2.2Case 5: 16 Elements LAA

In this case, the nulls are placed at 15.86° and the corresponding SLL and nulls are given in Table 4.10. From the results, it can be seen that the proposed JABC algorithm achieves highly competitive when compared to other algorithms. The maximum SLL achieved equals −18.83 dB. The radiation pattern and convergence curve for this antenna is given in Figure 4.12& 4.13.

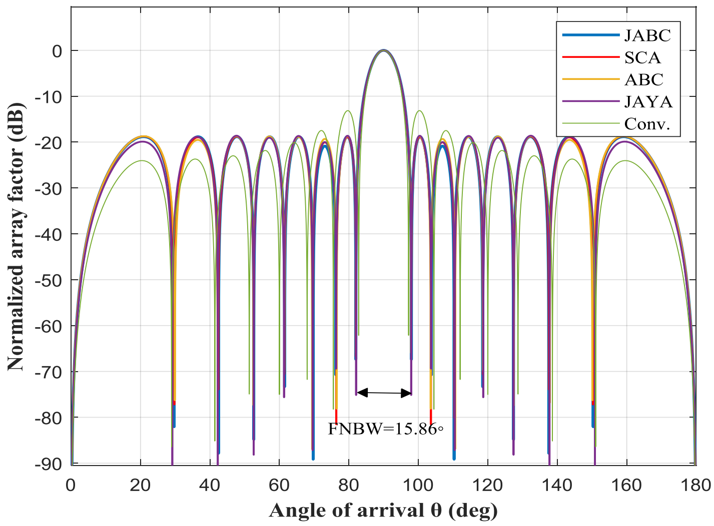

Figure 4.12: Radiation pattern obtained by suggested algorithms for 16-element linear array synthesis with FNBW constraint.

Table 4.10: Optimized amplitude for 16-element LAA obtained with suggested algorithms compared to other techniques with FNBW constraint.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \dots, I_8$	Peak SLL (dB)	FNBW
JABC	1.0000, 0.9982, 0.9753, 0.9091, 0.7939, 0.6479, 0.5043, 1.0000	-18.83	15.86°
ABC	1.0000, 0.9984, 0.9199, 0.8812, 0.7633, 0.6466, 0.5370, 1.0000	-18.71	15.86°
JAYA	1.0000, 1.0000, 0.9767, 0.8719, 0.8073, 0.6354, 0.5486, 1.0000	-18.61	15.86°
SCA	0.9993, 0.9953, 0.9527, 0.8879, 0.7737, 0.6454, 0.5147, 1.0000	-18.75	15.86°

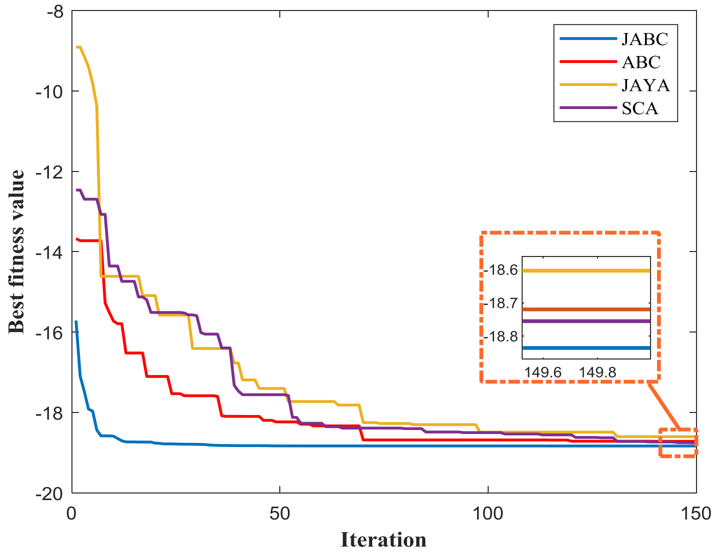

Figure 4.13: Convergence curve obtained by suggested algorithms for 16-element LAA over 150 iterations with FNBW constraint.

4.1.2.3 Case 6: 24 Elements LAA

For a 24-element LAA with FNBW at 11°, the optimum peak SLL for JABC is -21.39dB which is better when compared to ABC and JAYA having an SLL of −21.20 and −21.19 dB, respectively. Figure 4.14 shows the radiation pattern and convergence curve of 24-element with FNBW constraint. Here also, it can be seen that JABC is comparatively better and shows significant enhancement over ABC and JAYA algorithms.

Table 4.11: Optimized amplitude for 24-element LAA obtained with suggested algorithms compared to other techniques with FNBW constraint.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \ldots, I_{12}$	Peak SLL (dB)	FNBW
JABC	1.0000, 0.9999, 0.9769, 0.9292, 0.8727, 0.8117, 0.7429, 0.6702, 0.5980, 0.5176, 0.4251, 1.0000	-21.3982	11°
ABC	1.0000, 1.0000, 0.9920, 0.9125, 0.8829, 0.8274, 0.7300, 0.6813, 0.5909, 0.5319, 0.4428, 1.0000	-21.2003	11°
JAYA	1.0000, 0.9999, 0.9988, 0.9472, 0.8851, 0.8065, 0.7614, 0.6847, 0.6097, 0.5203, 0.4275, 1.0000	-21.1971	11°
SCA	1.0000, 1.0000, 0.9754, 0.9236, 0.8825, 0.8100, 0.7420, 0.6810, 0.5879, 0.5164, 0.4347, 1.0000	-21.3272	11°

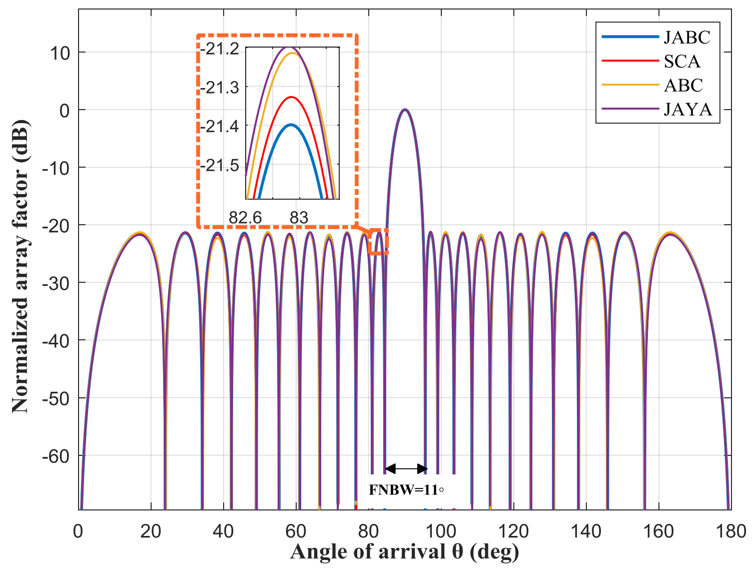

Figure 4.14: Radiation pattern obtained by suggested algorithms for 24-element linear array synthesis with FNBW constraint.

4.1.3 Optimization of Elements Positions (x_n) without FNBW constraint

In this section, by selecting the optimum positions for each LAA element, the minimum peak SLL is achieved. In order to achieve this, the amplitudes and phases should be fixed, i.e. ($\alpha_n = 0$ and $I_n = 1$) so the array factor becomes:

$$\text{AF}\ (\theta) = 2\sum_{n=1}^{N} cos\ [kx_n cos\ (\theta)\ ] \qquad (4.5)$$

The appropriative position of elements is important. Because mutual coupling effects can occur if antennas are placed too closely together, grating lobes result if placed too far away. Thus, to overcome the drawbacks indicated, the following conditions [55] must be achieved:

$$\left|x_i - x_j\right| > 0.25 \qquad (4.6)$$

$$\text{minimum}\ \{x_i\} > 0.125\lambda \quad \text{i=1,2, 3,..,N.}\ \ \text{i} \neq \text{j} \qquad (4.7)$$

4.1.3.1 Case 7: 10 Elements LAA

In this case, JABC, ABC, JAYA, and SCA algorithms are used for optimizing positions of a 10-element LAA to minimize peak SLL in $\theta \in [0°, 74°] \cup [106°, 180°]$. Each element position is used as a variable for proposed algorithms. Table 4.12 shows the peak SLL, FNBW, and optimum element positions determined using the suggested algorithms, while Table 4.13 shows the effectiveness of the JABC algorithm and others over 20 runs. The azimuth radiation pattern, along with the suggested algorithms and conventional LAA, is shown in Figure 4.15. The convergence curve of proposed algorithms for a 10-element LAA over 300 iterations is shown in Figure 4.16. The optimal peak SLLs for the 20 trials are shown in Figure 4.17. The box-and-whisker plots over 20 runs compared to other algorithms are shown in Figure 4.18.

Table 4.12: Optimized positions for 10-element LAA obtained with suggested algorithms compared to other techniques without FNBW constraint.

Evolutionary algorithm	Optimized element positions	Peak SLL (dB)	FNBW
JABC	0.1492λ, 0.3992λ, 0.7817λ, 1.0747λ, 1.6599λ	-23.36	38.6°
ABC	0.1367λ,0.4076λ, 0.7759λ, 1.0749λ, 1.6548λ	-23.22	38.6°
JAYA	0.1478λ, 0.3985λ, 0.7830λ, 1.0681λ, 1.6514λ	-23.01	38.6°
SCA	0.1380λ, 0.4050λ, 0.7780λ, 1.0740λ, 1.6570λ	-23.31	38.6°
CSO [55]	0.1510λ, 0.4110λ, 0.7890λ, 1.1040λ, 1.6840λ	-22.89	37.8°
MA [49]	0.2915λ, 0.5567λ, 0.9456λ, 1.2654λ, 1.8722λ	-22.79	30.6°
GOA [14]	0.3360λ, 0.4190λ, 1.0120λ, 1.4160λ, 2.1000λ	-21.31	29.5°
PSO [55]	0.2600λ, 0.5100λ, 1.0180λ, 1.4690λ, 2.1400λ	-20.72	28.5°
SMO [56]	0.2360λ, 0.5280λ, 1.0070λ, 1.4710λ, 2.1260λ	-20.25	28.5°
Uniform	0.2500λ, 0.7500λ, 1.2500λ, 1.7500λ, 2.2500λ	-12.96	23°

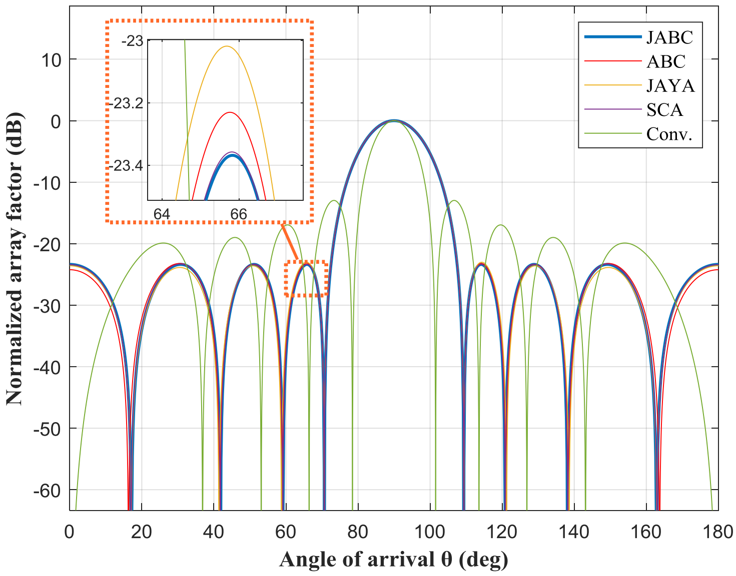

Figure 4.15: Radiation pattern obtained by suggested algorithms for 10-element linear array synthesis without FNBW constraint.

Table 4.13: Performance of JABC algorithm with 10 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Mean (dB)	Worst SLL (dB)	STD (dB)
JABC	-23.3685	-23.1746	-23.3685	0.000
ABC	-23.2157	-21.8215	-20.3967	1.0389
JAYA	-23.0170	-22.3404	-18.3562	1.4596
SCA	-23.3087	-22.0638	-20.5322	1.0177
GOA [14]	-21.31	-21.15	-20.94	0.1150

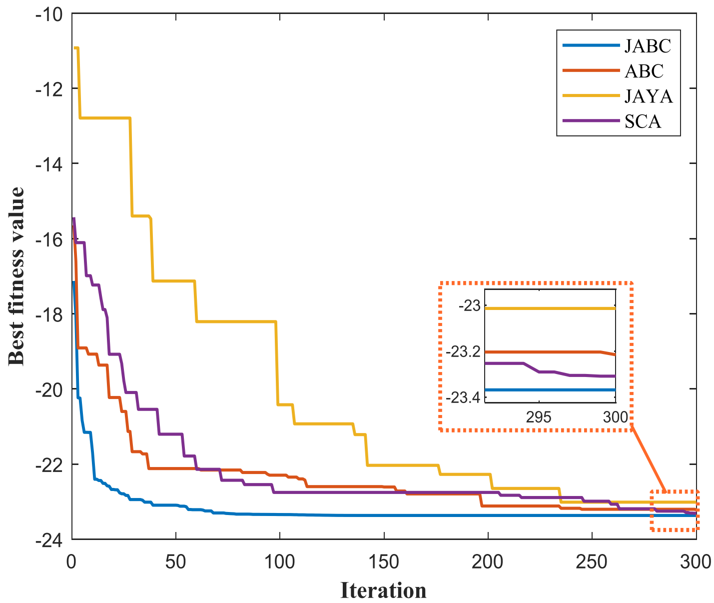

Figure 4.16: Convergence curve obtained by suggested algorithms for 10-element LAA over 300 iterations without FNBW constraint.

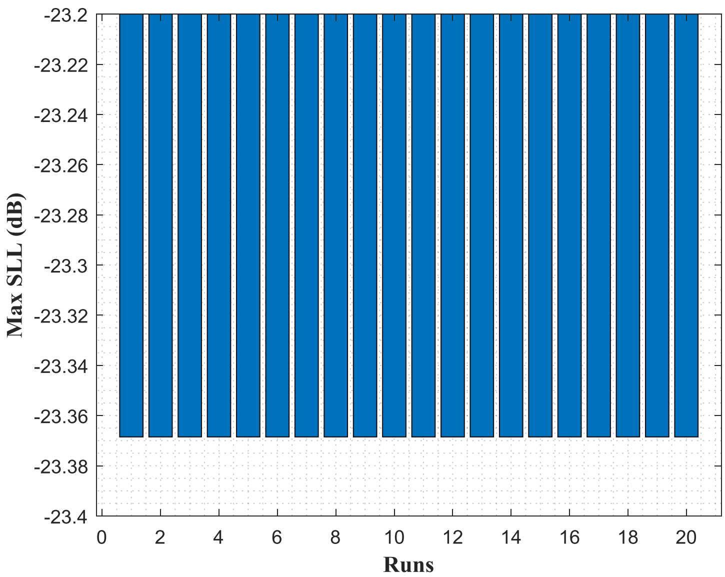

Figure 4.17: The maximum SLL obtained by JABC for 10 elements in 20 independent trials.

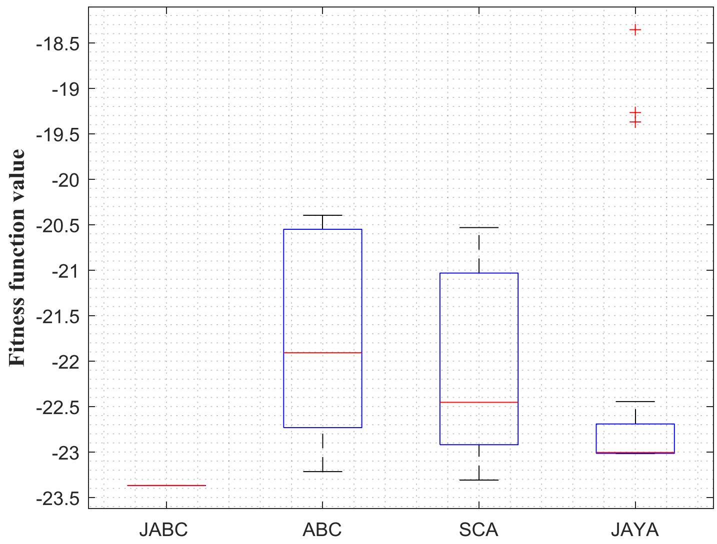

Figure 4.18: Box and whisker plot of 10-element LAA in 20 runs without FNBW constraint.

The results presented in Table 4.12 demonstrate the superior performance of the JABC algorithm compared to other optimization methods. The peak Sidelobe Level (SLL) achieved with JABC is -23.36 dB, which outperforms the results obtained using ABC, JAYA, SCA, CSO, GOA, PSO, SMO, and the conventional uniform LAA method by 0.14 dB, 0.35 dB, 0.05 dB, 0.47 dB, 2.05 dB, 2.64 dB, 3.11 dB, and 10.4 dB, respectively. This significant improvement in SLL indicates the effectiveness of the JABC algorithm in reducing unwanted side lobes and improving the overall performance of the linear antenna array (LAA). Furthermore, Figure 4.15 illustrates the impact of the JABC algorithm on the first null beam width (FNBW). It can be observed that the FNBW increased from 23° to 38.6°, indicating a broader main lobe coverage and improved beam steering capabilities. This wider FNBW allows for better control and flexibility in directing the main lobe of the LAA, leading to enhanced beam shaping and target tracking capabilities.

These findings highlight the significant advantages of utilizing the JABC algorithm in optimizing the performance of linear antenna arrays. By achieving lower SLL values and wider

FNBW, the JABC algorithm enables improved signal quality, increased interference rejection, and enhanced system performance in various real-world applications.

4.1.3.2 Case 8: 28 Elements LAA

For a 28 element LAA, the optimization of element positions without FNBW constraint is done in this case. Table 4.14 shows the peak SLL, FNBW, and optimum element positions determined using the suggested algorithms, while Table 4.15 shows the effectiveness of the JABC algorithm and others over 20 runs. The azimuth radiation pattern, along with the suggested algorithms and conventional LAA, is shown in Figure 4.19. The convergence curve of proposed algorithms over 500 iterations is shown in Figure 4.20. The optimal peak SLLs for the 20 trials are shown in Figure 4.21. the boxplot of results for each proposed algorithm over 20 runs compared to other algorithms are shown in Figure 4.22.

Table 4.14: Optimized positions for 28-element LAA obtained with suggested algorithms compared to other techniques without FNBW constraint.

Evolutionary algorithm	Optimized element positions	Peak SLL	FNBW
JABC	0.2339λ, 0.4994λ, 0.8938λ, 1.2786λ, 1.5790λ, 2.0768λ, 2.3728λ, 2.8697λ, 3.2608λ, 3.8343λ, 4.3208λ, 4.9977λ, 5.8325λ, 6.6690λ	-25.8	10.8°
ABC	0.2390λ, 0.4890λ, 0.9119λ, 1.1619λ, 1.6726λ, 1.9547λ, 2.3402λ, 2.7994λ, 3.2090λ, 3.7500λ, 4.2910λ, 4.8917λ, 5.6927λ, 6.4138λ	-24.06	10.9°
JAYA	0.2330λ, 0.5084λ, 0.9095λ, 1.3151λ, 1.6198λ, 2.1014λ, 2.4002λ, 2.9195λ, 3.3246λ, 3.8986λ, 4.4290λ, 5.0937λ, 5.9251λ, 6.7293λ	-24.61	10.6°
SCA	0.1899λ, 0.5975λ, 0.9345λ, 1.3616λ, 1.6962λ, 2.1626λ, 2.5776λ, 3.0175λ, 3.4987λ, 4.0743λ, 4.5971λ, 5.3344λ, 6.1652λ, 6.9720λ	-24.51	10.5°
CSO [55]	0.2344λ, 0.5280λ, 0.9224λ, 1.2965λ, 1.6549λ, 2.1427λ, 2.4387λ, 2.9369λ, 3.3753λ, 3.9280λ, 4.4091λ, 5.1167λ, 5.9188λ, 6.7422λ	-24.53	10.5°
PSO 55]	0.1703λ, 0.6430λ, 0.9509λ, 1.4245λ, 1.7849λ, 2.0397λ, 2.4511λ, 3.0522λ, 3.0522λ, 3.6249λ, 4.0476λ, 4.6302λ, 5.2984λ, 6.7118λ	-21.89	10°
Uniform	0.2500λ, 0.7500λ, 1.2500λ, 1.7500λ, 2.2500λ, 2.7500λ, 3.2500λ, 3.7500λ, 4.2500λ, 4.7500λ, 5.2500λ, 5.7500λ, 6.2500λ, 6.7500λ	-13.27	8°

Table 4.15: Performance of JABC algorithm with 28 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Mean (dB)	Worst SLL (dB)	STD (dB)
JABC	-25.8239	-23.8046	-22.0355	1.5324
ABC	-24.0627	-22.4751	-20.9831	0.7996
JAYA	-24.6182	-22.3143	-19.1872	1.6118
SCA	-24.5153	-23.5635	-22.1911	0.5772

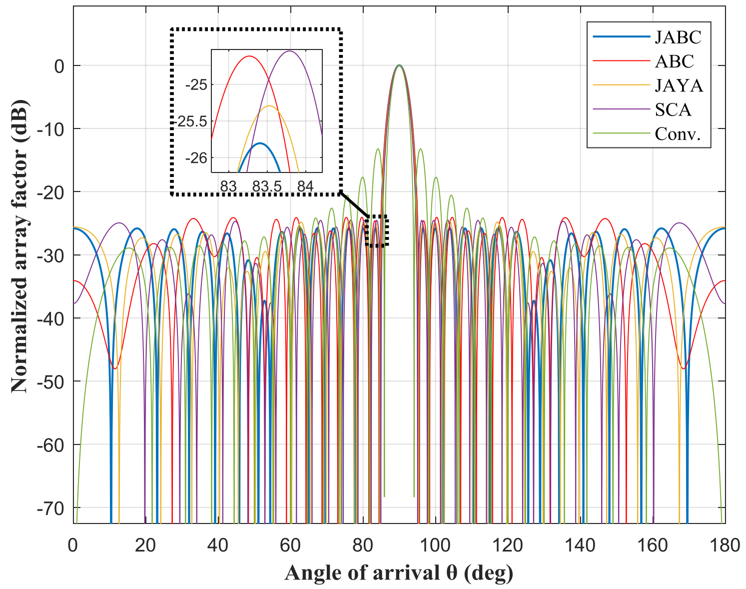

Figure 4.19: Radiation pattern obtained by suggested algorithms for 28-element linear array synthesis without FNBW constraint.

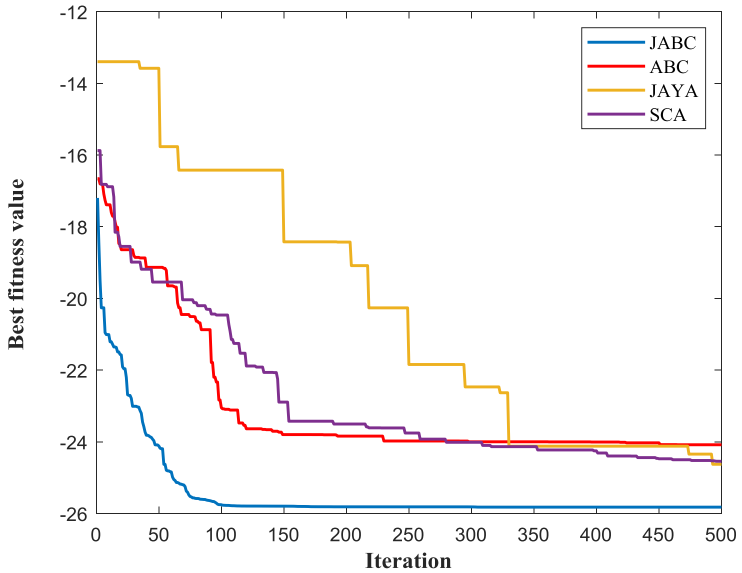

Figure 4.20: Convergence curve obtained by suggested algorithms for 28-element LAA over 500 iterations without FNBW constraint.

The results show that for an FNBW at 10.8°, the maximum SLL is achieved by JABC and equals −25.8 dB, which is comparatively better compared to ABC (−24.06dB), JAYA (−24.61dB), CSO (−24.53dB), and others.

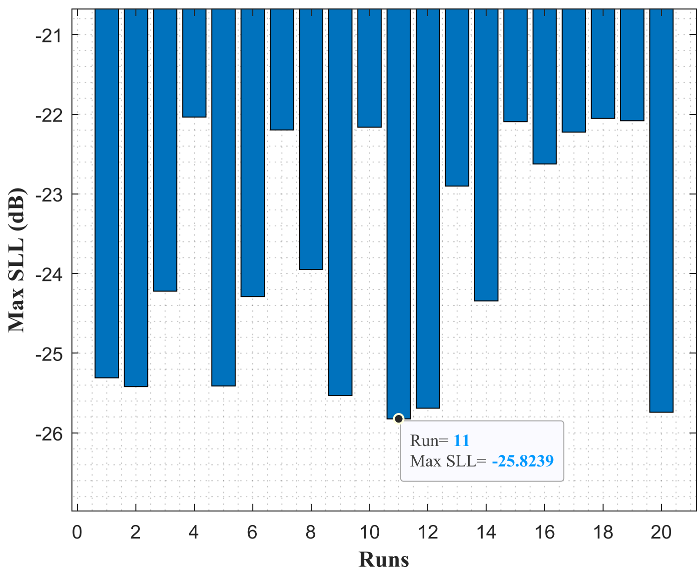

Figure 4.21: The maximum SLL obtained by JABC for 28 elements in 20 independent trials.

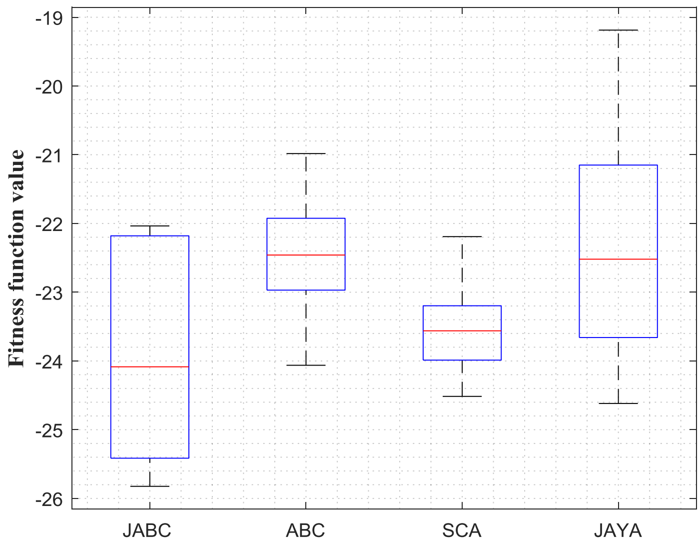

Figure 4.22: Box and whisker plot of 28-element LAA in 20 runs without FNBW constraint.

4.1.4 Optimization of Elements Positions (x_n) with FNBW constraint

This section focuses on reducing the Sidelobe Level (SLL) while keeping the First Null Beam Width (FNBW) within a tolerance range of ±5%. Achieving a lower SLL is important for minimizing interference and improving signal quality in antenna systems. However, it is equally crucial to maintain the FNBW within the specified tolerance to preserve the main lobe coverage and beam steering capabilities of the antenna array.

The use of the JABC algorithm enables us to explore the solution space effectively and identify the best LAA configuration. Through iterative adjustments and leveraging the algorithm's hybridization, we can enhance interference rejection while preserving the desired beam characteristics.

The results obtained from this section will provide valuable insights into the trade-offs and performance characteristics of the optimized LAA designs. They contribute to advancing antenna design methodologies and aid in the development of efficient communication systems that meet real-world requirements.

4.1.4.1 Case 9: 10 Elements LAA

In this case, JABC, ABC, SCA, and JAYA algorithms are used for optimizing positions of a 10-element LAA to minimize peak SLL with FNBW constraint. The azimuth radiation pattern along with the proposed algorithms, is shown in Figure 4.23 as compared to conventional uniform LAA methods. Table 4.16 shows the peak SLL, FNBW, and optimized element positions obtained by proposed algorithms and compared with other algorithms. The convergence curve for a 10-element LAA over 300 iterations is shown in Figure 4.24. The optimal peak SLLs and the boxplot of results for each proposed algorithm over 20 trials are shown in Figure 4.21 & 4.22, respectively. Table 4.17 shows the effectiveness of the JABC algorithm and others over 20 runs.

Table 4.16: Optimized positions for 10-element LAA obtained with suggested algorithms compared to other techniques with FNBW constraint.

Evolutionary algorithm	Optimized element positions	Peak SLL (dB)	FNBW
JABC	0.2060λ, 0.6482λ, 1.1074λ, 1.6784λ, 2.3468λ	-19.17	23.8°
ABC	0.2062λ, 0.6475λ, 1.1054λ, 1.6764λ, 2.3454λ	-19.09	23.8°
JAYA	0.2114λ, 0.6822λ, 1.1576λ, 1.7658λ, 2.4508λ	-18.75	23.3°

SCA	0.2057λ, 0.6482λ, 1.1070λ, 1.6778λ, 2.3452λ	-19.15	23.8°
CSO [55]	0.2081λ, 0.6670λ, 1.1340λ, 1.7230λ, 2.4030λ	-19.09	23.2°
PSO [55]	0.2040λ, 0.6270λ, 1.6640λ, 1.4560λ, 2.3030λ	-17.82	23.8°
Uniform	0.2500λ, 0.7500λ, 1.2500λ, 1.7500λ, 2.2500λ	-13.23	23°

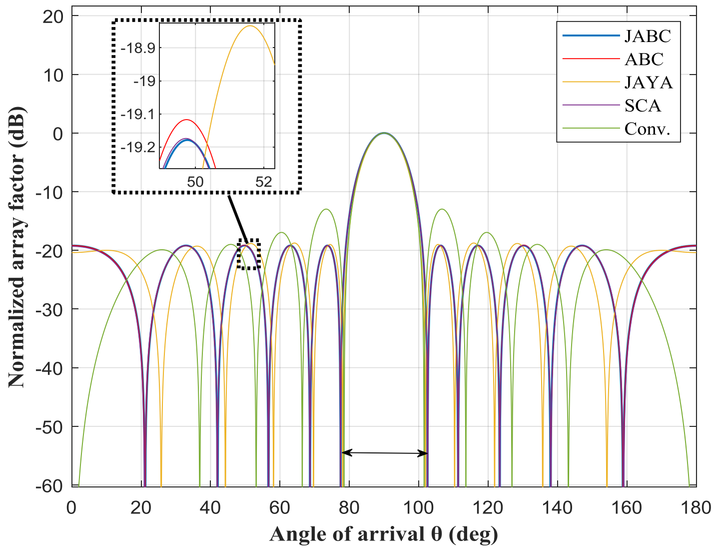

Figure 4.23: Radiation pattern obtained by suggested algorithms for 10-element linear array synthesis with FNBW constraint.

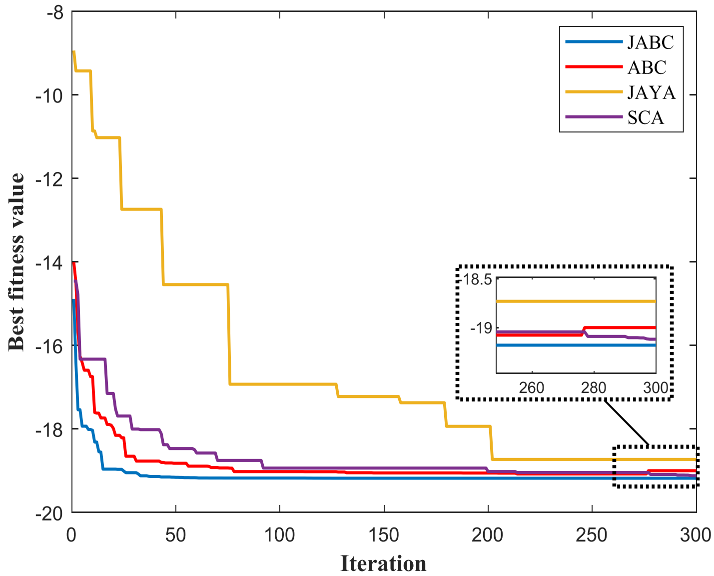

Figure 4.24: Convergence curve obtained by suggested algorithms for 10-element LAA over 300 iterations with FNBW constraint.

Table 4.17: Performance of JABC algorithm with 10 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Mean (dB)	Worst SLL (dB)	STD (dB)
JABC	-19.1798	-19.1192	-19.0667	0.0565
ABC	-19.0943	-19.0321	-18.9375	0.0349
JAYA	-18.7526	-18.7180	-17.6352	0.2836
SCA	-19.1537	-19.0956	-19.0282	0.0337

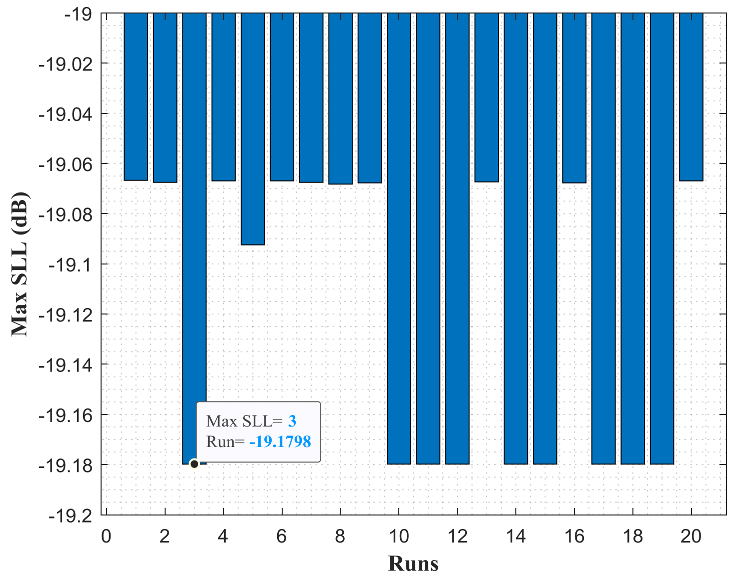

Figure 4.25: The maximum SLL obtained by JABC for 10 elements in 20 independent trials.

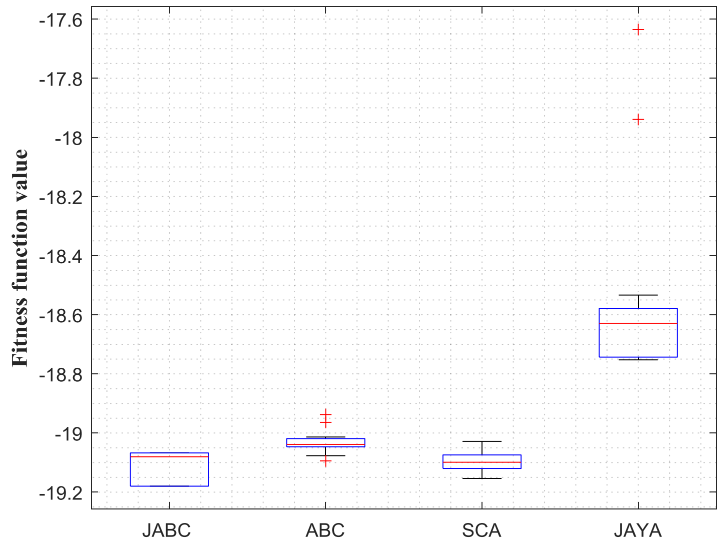

Figure 4.26: Box and whisker plot of 10-element LAA in 20 runs with FNBW constraint.

According to the results in Table 4.16, the proposed JABC has a maximum SLL of −19.17dB at 23.8° FNBW, which is better than SCA (-19.15 dB), ABC (−19.09dB), JAYA (-18.75 dB), CSO (−19.09dB), PSO (−17.82dB) and uniform array (−13.23dB).

4.1.4.2 Case 10: 28 Elements LAA

This case deals with the optimization of a 28 element LAA with FNBW at 8.1°. For this case also JABC provides better maximum SLL at −24dB, which is −0.62dB better than ABC (−23.38dB) and −1.81dB higher than JAYA. For CSO, the maximum SLL is −20.32dB, for PSO is −17.22dB and for a uniform array it is −13.27dB. The radiation patterns and convergence curves for this case are given in Figure 4.27& 4.28, and it shows that JABC is significantly better compared to other algorithms under comparison.

Table 4.18: Optimized positions for 28-element LAA obtained with suggested algorithms compared to other techniques with FNBW constraint

Evolutionary algorithm	Optimized element positions	Peak SLL (dB)	FNBW
JABC	0.3020λ, 0.5520λ 1.2044λ, 1.4931λ, 2.0428 λ, 2.4649λ, 3.0042λ, 3.5055λ, 4.1377λ, 4.6279λ, 5.3754λ, 6.2116λ, 7.1368λ, 8.0250λ	-24	8.1°
ABC	0.2601λ, 0.5387λ, 1.1806 λ, 1.5598λ, 2.0798λ, 2.4336λ, 3.0120λ, 3.5685λ, 4.1101λ, 4.6942λ, 5.3669λ, 6.2867λ, 7.1092λ, 8.0035λ	-23.38	8.1°
JAYA	0.2726λ, 0.5963λ, 1.1102λ, 1.5255λ, 2.0168λ, 2.4626λ, 2.9568λ, 3.4468λ, 3.9887λ, 4.5815λ, 5.2776λ, 6.0859λ, 7.0282λ, 7.7562λ	-22.19	8.1°
SCA	0.2500λ, 0.6228λ, 1.1642λ, 1.5056λ, 2.0172λ, 2.4376λ, 3.0187λ, 3.5381λ, 4.0346λ, 4.6429λ, 5.3507λ, 6.2253λ, 7.0320λ, 7.9267λ	-23.33	8.1°
CSO [55]	0.2437λ, 0.6445λ, 1.0230λ, 1.5095λ, 1.8444λ, 2.3974λ, 2.8835λ, 3.2657λ, 3.8500λ, 4.4726λ, 5.1068λ, 5.8367λ, 6.5065λ,6.9999λ	-20.32	8.1°
PSO [55]	0.3270λ, 0.5771λ, 0.2089λ, 1.5145λ, 2.1417λ, 2.3939λ, 2.8792λ, 3.4450λ, 4.3046λ, 4.8928λ, 5.1472λ, 5.9070λ, 6.4275λ, 6.9999λ	-17.22	8.1°
Uniform	0.2500λ, 0.7500λ, 1.2500λ, 1.7500λ, 2.2500λ, 2.7500λ, 3.2500λ, 3.7500λ, 4.2500λ, 4.7500λ, 5.2500λ, 5.7500λ, 6.2500λ, 6.7500λ	-13.27	8°

Table 4.19: Performance of JABC algorithm with 28 elements compared to other algorithms over 20 runs.

Evolutionary algorithm	Best SLL (dB)	Mean (dB)	Worst SLL (dB)	STD (dB)
JABC	-24.0039	-23.3572	-21.2026	0.6759
ABC	-23.3898	-21.8986	-20.9522	0.6215
JAYA	-22.1962	-21.6175	-19.9214	0.7336
SCA	-23.3301	-22.9053	-22.1618	0.3181

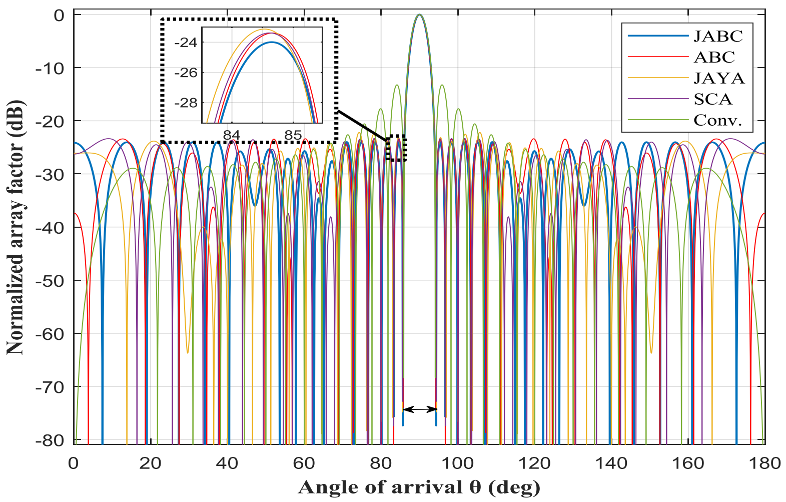

Figure 4.27: Radiation pattern obtained by suggested algorithms for 28-element linear array synthesis with FNBW constraint.

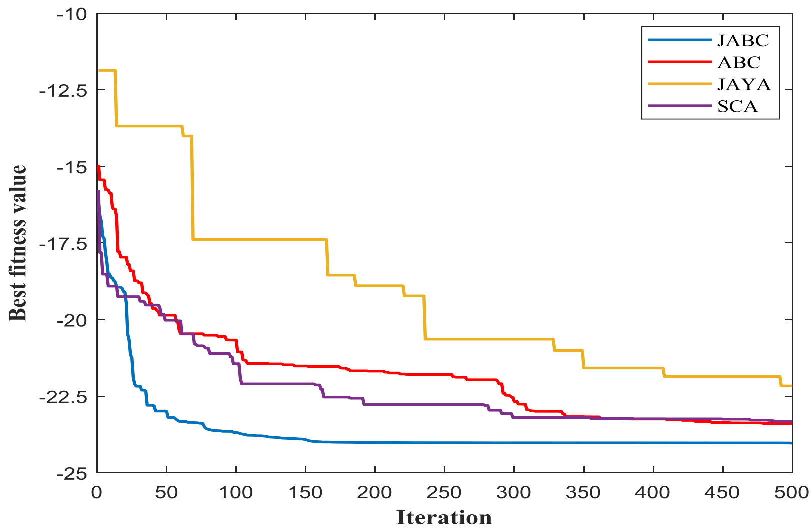

Figure 4.28: Convergence curve obtained by suggested algorithms for 28-element LAA over 500 iterations with FNBW constraint.

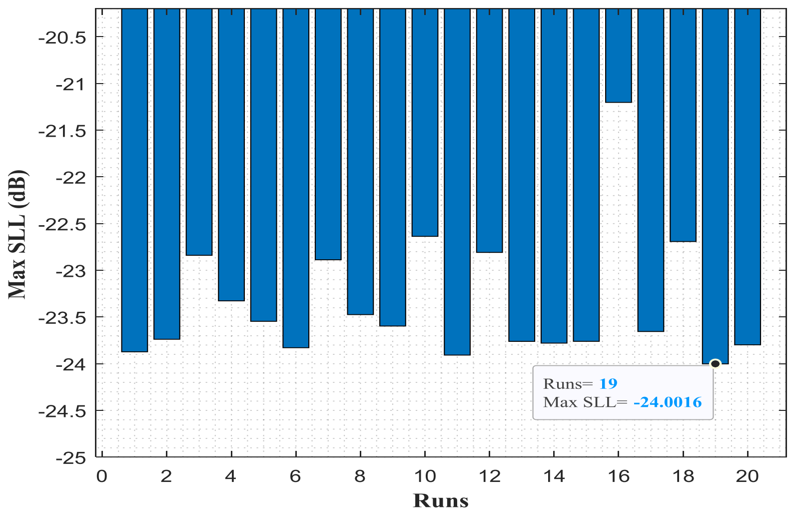

Figure 4.29: The maximum SLL obtained by JABC for 28 elements in 20 independent trials.

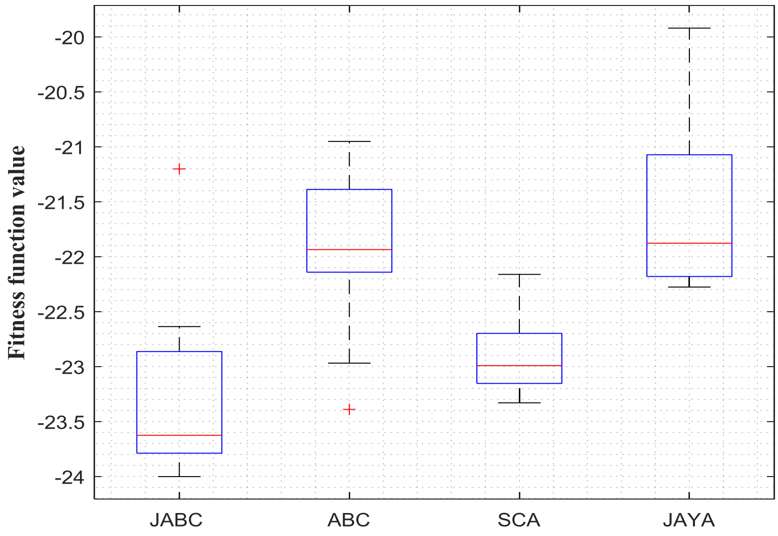

Figure 4.30: Box and whisker plot of 28-element LAA in 20 runs with FNBW constraint.

4.1.5 Optimization of Elements Phases (α_n)

As a uniform array, we set $I_n = 1$ and the spaces between elements (d= λ/2). Initial phase values are uniformly distributed in (0, 180). Elements Phases are considered to be symmetric as ($\alpha_n = \alpha_{-n}$ $n = 1, 2, \ldots, N$)**.** Where (α_n) phase of the n^{th} element. So, the AF becomes as the following:

$$\text{AF}(\theta)=2\sum_{n=1}^{N} exp(j\alpha_n)cos[(n-0.5)\,\pi cos(\theta)] \tag{4.8}$$

4.1.5.1 Case 11: 40 Elements LAA

To optimize the phases of a 40-element LAA, an amplitude of unity is set, and element spacing is kept at 0.5λ. The phase of excitations is defined as [0, π], and from Table 4.20, the maximum SLL using JABC is -18.18dB which is comparatively close when compared to SCA having an SLL of -18.05 dB. However, the radiation patterns and convergence curves in Figures 4.31 & 4.32 prove the significance of JABC over other algorithms under comparison.

Table 4.20: Optimized phases for a 40-element LAA obtained with proposed algorithms compared to other techniques.

Evolutionary algorithm	**Optimized element phases** $\alpha_1, \alpha_2, \ldots, \alpha_{20}$**(deg.)**	**Peak SLL in dB**	**FNBW**
JABC	73.485, 71.919, 74.404, 69.146, 66.585, 69.589, 71.782, 61.804, 64.706, 55.43, 65.562, 86.243, 6.4126, 45.037, 96.356, 177.48, 49.21, 109.66, 68.351, 73.276	-18.18	6.2°
ABC	122.51, 106.14, 106.93, 103.23, 117.01, 109.05, 106.66, 122.18, 113.8, 126.04, 129.77, 128.18, 154.54, 65.856, 140.34, 35.447, 50.567, 128.97, 130.64, 83.731	-17.25	6.5°
JAYA	90.549, 92.692, 87.634, 89.066, 92.932, 74.032, 87.602, 75.37, 64.679, 106.8, 72.997, 77.751, 112.9, 0.6161, 94.598, 115.31, 176.24, 66.407, 88.068, 89.272	-17.74	6.3°
SCA	76.89, 72.858, 75.491, 78.844, 78.536, 65.37, 67.364, 68.626, 74.999, 47.087, 58.069, 91.883, 80.885, 4.4867, 73.523, 179.55, 80.937, 118.79, 67.769, 63.385	-18.05	6.4°
SOS [17]	28.3636, 25.0046, 22.2290, 31.1901, 23.7626, 17.3337,15.5147, 39.0199, 18.1678, 7.8822, 1.8298, 60.0022, 0, 0.0146, 0.0161, 148.3908, 45.0096, 56.1693, 61.9867, 2.1350	-18.02	6.6°
CS [21]	45.9692, 39.7155, 39.6464, 36.8069, 41.0828, 42.4519, 50.2623, 32.5464,36.8147, 34.4894, 30.8162, 16.1212, 81.8888, 20.4923, 41.963, 177.6511,30.7085, 53.6503, 35.3756, 87.3351	-17.59	6.4°
GA [68]	69.7175, 68.4570, 72.3187, 63.5582, 53.3699, 51.9283, 66.1537, 36.5971, 50.4650, 38.3526, 75.1950, 15.6011, 91.3810, 39.8412, 83.9670, 171.8873, 32.3028, 28.6863, 57.2958, 73.1724	-17.39	6.2°
BBO [68]	90.4185, 90.5331, 97.2825, 90.2466, 88.3840, 97.1507, 90.0002, 90.3497, 97.2596, 85.9950, 75.0002, 115.5026, 71.8604, 0.3610, 122.9166, 97.0247, 178.8087, 83.3081, 83.9670, 79.2057	-17.96	6.2°
Uniform	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	-13.24	6.2°

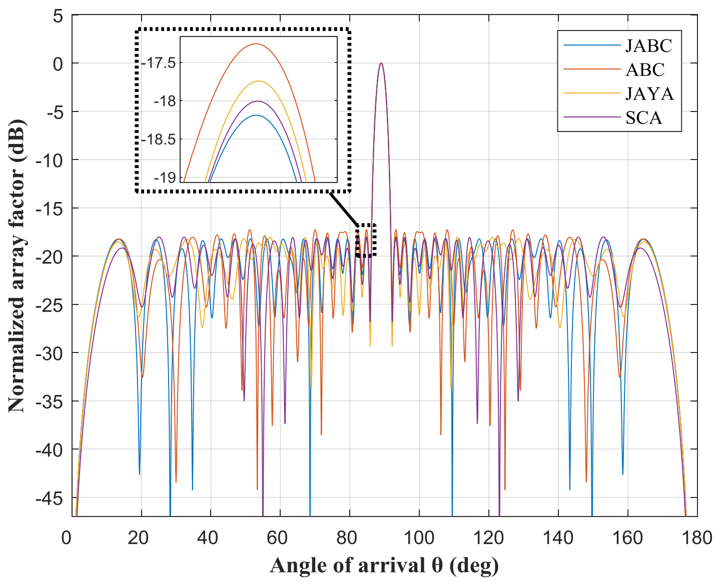

Figure 4.31: Radiation pattern obtained by JABC, ABC, JAYA, and SCA for 40-elements linear array synthesis

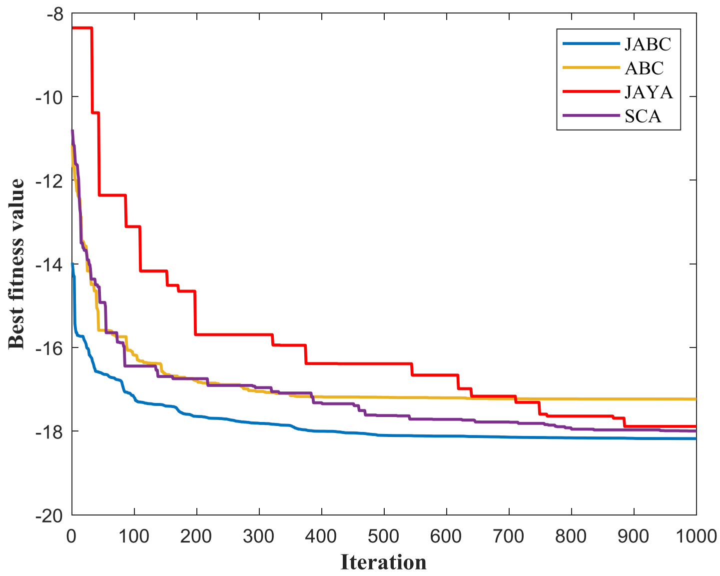

Figure 4.32: Convergence curve for 40-element LAA optimized over 1000 iterations.

4.1.6 Scanned Linear Antenna Array (LAA)

The AF equation for an N-element scanning antenna array that has a spacing of (λ/2) between consecutive elements can be expressed as follows:

$$AF\ (\theta)= \sum_{n=1}^{N} I_n exp(j\pi(n-1)\ [cos\ (\Theta)\ - cos(\emptyset_d)] \qquad (4.9)$$

Where θ is the azimuth angle, $\emptyset_d$ and I_n are the steering angle of the main lobe and the excitation amplitude of n^{th} element, respectively. In this section, the chosen values for the area of the side lobe that is dependent on the number of elements are [0,15°] ∪ [41°,180°],[0,38°] ∪[52°,180°], and [0,58°] ∪ [65°,180°] for the corresponding numbers 20, 26, and 30 elements and the steering angle of the main lobe 30°, 45°, and 60°**,** respectively.

4.1.6.1 Case 12: 20 Elements LAA with $\emptyset_d$ = 30°

In this scanned array, we consider a 20-element LAA where $\emptyset_d$ = 30°. Here 20 independent runs are performed to check the simulation results. Table 4.21 shows the results of proposed JABC with respect to Hybrid, SOS, ALO, JAYA, ABC, SCA, FA and conventional antenna array. Figure 4.33 shows the array pattern produced using the proposed algorithms. The convergence curve over 1000 iterations is shown in Figure 4.34. It can be observed from Table 4.22 that the Standard deviation (STD) for this algorithm is lower than the STD for Hybrid, ALO, SOS, and FA, which demonstrates the accuracy and robustness of the suggested techniques. Also, from the optimized array amplitudes, and the SLL, we can say that JABC provides better results, and this is because of the dynamic adaptive properties of the proposed algorithm.

Table 4.21: Optimal element amplitudes obtained with (N=20 and $\emptyset_d$ = 30°) LAAs using the proposed algorithms and compared to other methods.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \ldots, I_{20}$	Peak SLL (dB)	FNBW
JABC	0.9081, 0.2717, 0.4300, 0.3276, 0.4991, 0.5560, 0.4272, 0.5507, 0.7179, 0.5368, 0.6486, 0.3827, 0.6217, 0.6328, 0.4150, 0.4458, 0.4856, 0.3253, 0.4020, 0.7827	-17.07	25°
ABC	0.7847, 0.2210, 0.5451, 0.5848, 0.5660, 0.5486, 0.5948, 0.5678, 0.8434, 0.5089, 0.8887, 0.4118, 0.6468, 0.8322, 0.5027, 0.4360, 0.3533, 0.3994, 0.6337, 1.0000	-16.72	25°
JAYA	0.9945, 0.9926, 0.6226, 0.8314, 0.3553, 0.9770, 0.7587, 0.9999, 0.9997, 0.8326, 0.9930, 0.8967, 0.9919, 0.7246, 0.7356, 1.0000, 0.7683, 0.1878, 0.9999, 1.0000	-16.57	25°
SCA	1.0000, 0.2760, 0.4123, 0.2976, 0.5385, 0.4721, 0.5947, 0.6386, 0.5094, 0.6917, 0.5002, 0.7120, 0.4426, 0.5985, 0.5217, 0.4476, 0.5832, 0.3241, 0.4439, 0.8145	-16.70	25°
Hybrid [54]	0.8605, 0.3907, 0.3162, 0.4714, 0.3610, 0.4231, 0.5925, 0.5250, 0.4333, 0.3947, 0.6472, 0.6108, 0.4942, 0.4035, 0.4594, 0.4787, 0.3173, 0.4131, 0.2273, 1.0000	-15.66	25°
SOS [17]	1.0000, 0.2762, 0.4499, 0.3040, 0.3787, 0.6113, 0.5305, 0.5042, 0.5554, 0.6113, 0.4950, 0.4909, 0.5940, 0.4393, 0.3429, 0.5587, 0.4266, 0.3142, 0.4099, 0.9092	-15.64	25°
ALO [54]	1.0000, 0.4893, 0.5830, 0.4354, 0.2566, 0.8691, 0.4855, 0.8047, 0.3192, 0.6924, 0.5965, 0.8191, 0.7401, 0.4672, 0.2588, 0.7958, 0.5334, 0.2349, 0.5780, 0.9991	-15.45	25°
FA [53]	0.9804, 0.7662, 0.3690, 0.5529, 0.9071, 0.2019, 0.5196, 0.8449, 0.5094, 0.9805, 0.5142, 0.5387, 0.8027, 0.5540, 0.8808, 0.4037, 0.3321, 0.4655, 0.5034, 0.9460	-15.59	25°
Conv.	1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000, 1.0000	-13.18	25°

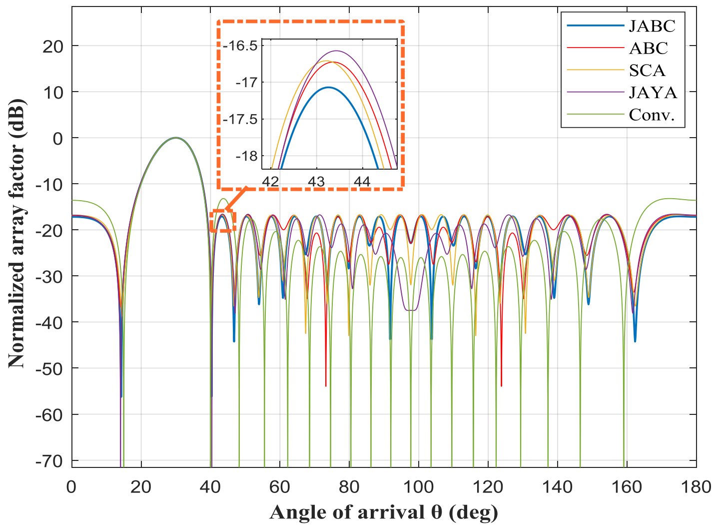

Figure 4.33: The radiation pattern obtained with $\varnothing_d$ = 30°, N=20.

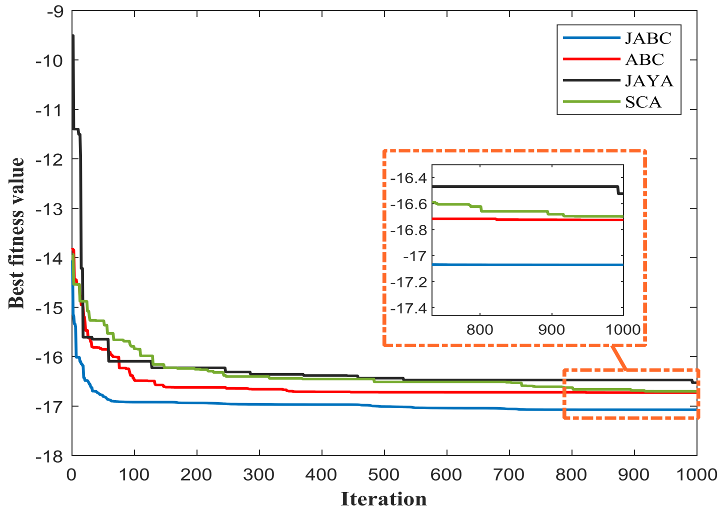

Figure 4.34: Convergence curve for 20-element LAA optimized over 1000 iterations.

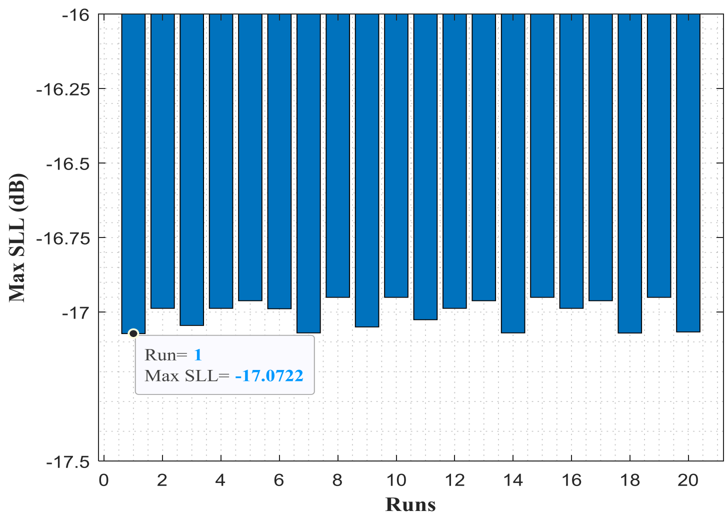

Figure 4.35: The maximum SLL obtained by JABC for 20 elements with $\varnothing_d$ = 30° in 20 independent trials.

Table 4.22: JABC algorithm effectiveness for (N=20 and $\emptyset_d$ = 30°) over 20 runs compared to other algorithms.

Evolutionary algorithm	Best SLL (dB)	Worst SLL (dB)	Mean (dB)	STD (dB)
JABC	-17.07	-16.95	-17.00	0.0422
ABC	-16.72	-16.25	-16.42	0.1150
JAYA	-16.57	-16.26	-16.41	0.0741
SCA	-16.70	-16.50	-16.63	0.0471
ALO [54]	-15.45	-15.14	-15.30	0.0859
Hybrid [54]	-15.66	-15.40	-15.56	0.0674
SOS [17]	-15.64	-	-16.04	0.1038
FA [53]	-15.59	-	-15.43	0.1332

4.1.6.2 Case 13: 26 Elements LAA with $\emptyset_d$ = 45°

The second case uses proposed algorithms to reduce the peak SLL of 26-element LAA with $\emptyset_d$ = 45°. Table 4.23 shows the peak SLL, FNBW, and optimum element amplitudes determined using the suggested algorithms, while Table 4.24 shows the effectiveness of the JABC over 20 runs. According to the results in Table 4.23, the peak SLL achieved by JABC has been minimized from -13.18 dB to -19.06 dB as compared to uniform array, which is less than FA, ALO, SOS, JAYA, SCA, ABC, and Hybrid algorithms. The azimuth radiation pattern along with the JABC algorithm is shown in Figure 4.36 compared to ABC, JAYA algorithms. The convergence curve over 1000 iterations is shown in Figure 4.37. It is also observed from Table 4.23 that the FNBW for all algorithms maintained constant.

Table 4.23: Optimal element amplitudes obtained with (N=26 and $\emptyset_d$ = 45°) LAAs using the proposed algorithms and compared to other methods.

Evolutionary algorithm	Optimized element amplitudes $I_1, I_2, \ldots, I_{26}$	Peak SLL (dB)	FNBW
JABC	0.9986, 0.5555, 0.2873, 0.3781, 0.6172, 0.6057, 0.7005, 0.4748, 0.8258, 0.5377, 0.8628, 0.8511, 0.6084, 0.9164, 0.6456, 0.7185, 0.8180, 0.6703, 0.7158, 0.6091, 0.4835, 0.4334, 0.6216, 0.4306, 0.3971, 1.0000	-19.06	13°
ABC	0.7795, 0.6937, 0.6097, 0.2405, 0.6110, 0.7186, 0.3999, 0.8060, 0.6769, 0.8737, 0.7362, 0.9632, 0.7535, 0.9318, 0.6730, 0.7943, 0.5323, 1.0000, 0.5222, 1.0000, 0.3787, 0.4538, 0.8284, 0.3290, 0.5060, 1.0000	-18.72	13°
JAYA	0.9891, 1.0000, 0.4235, 0.5412, 0.9807, 0.2657, 0.9223, 0.8166, 0.8942, 0.8613, 0.9600, 0.9804, 0.9858, 0.9808, 0.9980, 0.6952, 0.9903, 0.9250, 0.6557, 0.9407, 0.6533, 0.7935, 0.3677, 0.6364, 0.6018, 1.0000	-18.66	13°
SCA	1.0000, 0.4677, 0.2577, 0.5632, 0.3470, 0.3525, 0.5107, 0.6772, 0.5295, 0.6340, 0.4619, 0.6728, 0.6227, 0.5677,	-18.05	13°

	0.5690, 0.8123, 0.5362, 0.4292, 0.4769, 0.5183, 0.6707, 0.2039, 0.4677, 0.3104, 0.4615, 0.6696		
Hybrid [54]	0.9777, 0.3424, 0.5421, 0.0849, 0.2830, 0.5290, 0.5246, 0.4015, 0.4598, 0.5848, 0.3557, 0.5388, 0.5393, 0.5852, 0.3864, 0.6349, 0.4756, 0.5046, 0.3672, 0.4508, 0.3547, 0.3458, 0.5698, 0.2367, 0.2848, 1.0000	-16.32	13°
SOS [17]	1.0000, 0.2314, 0.4243, 0.4349, 0.3933, 0.4423, 0.4890, 0.3892, 0.5260, 0.5470, 0.3889, 0.8891, 0.4148, 0.5557, 0.4317, 0.7241, 0.4748, 0.3302, 0.7278, 0.5896, 0.2174, 0.5061, 0.1908, 0.4341, 0.6199, 0.9584	-16.18	13°
ALO [54]	0.9845, 0.9246, 0.0105, 0.6751, 0.6270, 0.0630, 0.6520, 0.5410, 0.5882, 0.7589, 0.4931, 0.6906, 0.8221, 0.4432, 0.5654, 0.6296, 0.6533, 0.5303, 0.4111, 0.7564, 0.3702, 0.7454, 0.0055, 0.7433, 0.3655, 1.0000	-16.05	13°
FA [53]	1.000, 0.7242, 0.5590, 0.4483, 0.7197, 0.3194, 0.7075, 0.6203, 0.5399, 0.8630, 0.6732, 0.7158, 0.8349, 0.7795, 0.4271, 0.7953, 0.7136, 0.6301, 0.6267, 0.6301, 0.7473, 0.0601, 0.7387, 0.5984, 0.7782, 0.9975	-15.61	13°
Conv.	1.0000, 1.0000	-13.22	13°

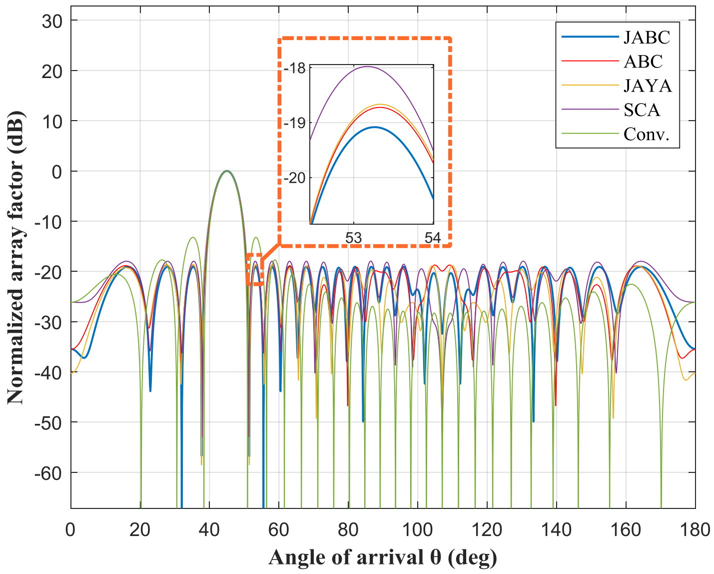

Figure 4.36: The radiation pattern obtained with $\O_d$ = 45°, N=26.

Table 4.24: JABC algorithm effectiveness for (N=26 and $\O_d$ = 45°) over 20 runs compared to other algorithms.

Evolutionary algorithm	Best SLL (dB)	Worst SLL (dB)	Mean (dB)	STD (dB)
JABC	-19.07	-18.76	-18.96	0.0920
ABC	-18.71	-17.95	-18.29	0.1937
JAYA	-18.66	-18.38	-18.55	0.0847
SCA	-18.05	-17.93	-17.99	0.0573
ALO [54]	-16.05	-15.76	-15.90	0.0943
Hybrid [54]	-16.32	-16.11	-16.21	0.0911
SOS [17]	-16.18	-	-15.84	0.0930
FA [53]	-15.61	-	-15.82	0.1184

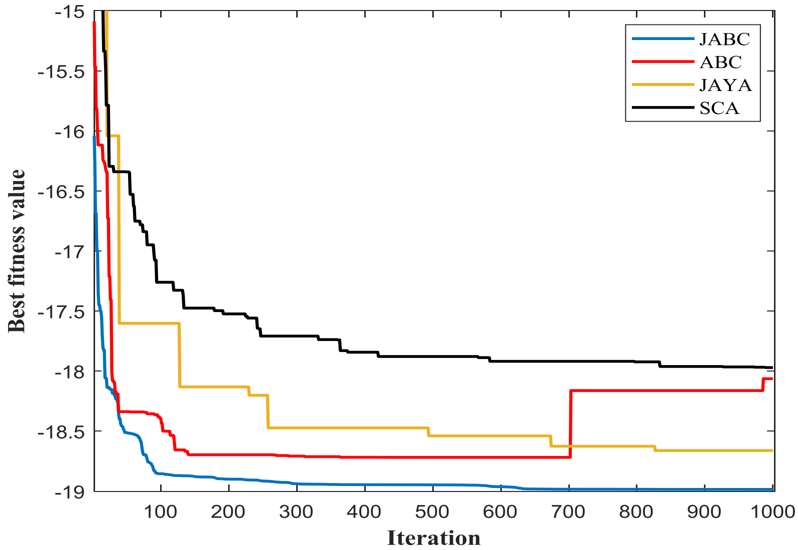

Figure 4.37: Convergence curve for 26-element LAA optimized over 1000 iterations.

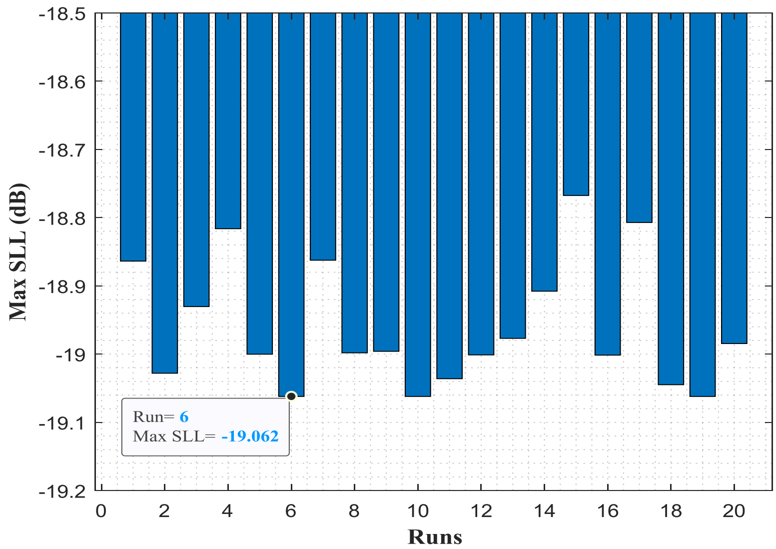

Figure 4.38: The maximum SLL obtained by JABC for 26 elements with $\emptyset_d$ = 45° in 20 independent trials.

4.1.6.3 Case 14: 30 Elements LAA with $\emptyset_d$ = 60°

In this case, the optimum excitation amplitude for 30 elements and 60° scanned LAA obtained and tabulated in Table 4.25, with Peak SLL and FNBW for the proposed algorithms. This table demonstrates that the JABC algorithm outperforms other methods considerably. The azimuth radiation pattern and the convergence curve along with the JABC algorithm is shown in Figure 4.39 and Figure 4.40, respectively. While Table 4.26 shows the effectiveness of the JABC algorithm over 20 runs.

Table 4.25: : Optimal element amplitudes obtained with (N=30 and $\emptyset_d$ = 60°) LAAs using the proposed algorithms and compared to other methods.

Evolutionary algorithm	**Optimized element amplitudes** $I_1, I_2, \ldots, I_{30}$	**Peak SLL (dB)**	**FNBW**
JABC	0.7255, 0.5136, 0.3879, 0.2863, 0.3306, 0.5426, 0.4878, 0.6280, 0.5099, 0.6187, 0.5997, 0.6344, 0.5382, 0.6164, 0.7717, 0.6675, 0.5843, 0.7055, 0.7910, 0.5832, 0.4895, 0.6713, 0.4827, 0.3558, 0.5498, 0.4017, 0.3594, 0.4210, 0.1964, 0.9298	-20.07	9°
ABC	0.4028, 0.6435, 0.8249, 0.2615, 0.4171, 0.7219, 0.3403, 0.4392, 0.6600, 0.5597, 0.6757, 0.7707, 0.9473, 0.6510, 0.6424, 0.8632, 0.7336, 0.9645, 0.5653, 0.6046, 0.5515, 0.9695, 0.6940, 0.6890, 0.5173, 0.2871, 0.6137, 0.4476, 0.2375, 1.0000	-19.32	9°
JAYA	0.9607, 0.6498, 0.5086, 0.4688, 0.3694, 0.7643, 0.5061, 0.8620, 0.8074, 0.7488, 0.7894, 0.8300, 0.8842, 0.7899, 0.9196, 0.8876, 0.9413, 0.9018, 0.8699, 0.7766, 0.8742, 0.5615, 0.8370, 0.5591, 0.5071, 0.7948, 0.4867, 0.3045, 0.6565, 1.0000	-19.90	9.3°
SCA	0.9897, 0.4624, 0.1580, 0.3478, 0.3486, 0.2591, 0.5198, 0.5018, 0.4885, 0.5101, 0.6641, 0.5252, 0.5595, 0.5786, 0.6234, 0.5115, 0.5406, 0.6200, 0.5150, 0.3568, 0.6000, 0.5028, 0.2902, 0.5674, 0.4192, 0.5451, 0.1918, 0.4242, 0.3971, 0.7614	-18.14	9°
Hybrid [54]	0.7668, 0.3184, 0.2907, 0.3373, 0.2029, 0.3096, 0.2510, 0.4666, 0.3332, 0.2843, 0.2275, 0.5557, 0.4165, 0.4146, 0.3572, 0.4841, 0.4154, 0.2713, 0.2931, 0.4726, 0.5228, 0.2981, 0.2830, 0.3537, 0.3229, 0.2353, 0.3448, 0.1407, 0.2381, 1.0000	-16.20	8.96°
SOS [17]	1.0000, 0.9219, 0.4011, 0.1512, 0.6258, 0.0149, 0.7433, 0.5357, 0.4412, 0.8182, 0.3055, 0.5388, 0.8813, 0.5962, 0.4734, 0.8110, 0.3965, 0.6665, 0.3149, 0.7865, 0.6591, 0.4047, 0.3755, 0.5224, 0.5257, 0.5935, 0.2734, 0.3698, 0.6766, 0.9982	-15.93	9°
ALO [54]	0.9380, 0.5472, 0.4980, 0.4647, 0.4777, 0.0382, 0.4824, 0.7979, 0.3462, 0.4455, 0.6926, 0.3170, 0.6597, 0.6022, 0.7500, 0.1379, 0.8532, 0.5132, 0.7009, 0.1911, 0.7336, 0.7231, 0.0303, 0.6409, 0.5290, 0.3575, 0.3012, 0.2129, 0.7843, 1.0000	-15.94	9°
FA [53]	0.9957, 0.6844, 0.6299, 0.0499, 0.1793, 0.7345, 0.4852, 0.6181, 0.3336, 0.6318, 0.6364, 0.3934, 0.4918, 0.7724, 0.6454, 0.4840, 0.7396, 0.7441, 0.5279, 0.4501, 0.8221, 0.5290, 0.4582, 0.4190, 0.4868, 0.2416, 0.8668, 0.6361, 0.2969, 0.9993	-15.97	9.08°
Conv.	1.0000, 1.0000	-13.21	8.52°

Table 4.26: JABC algorithm effectiveness for (N=30 and $\emptyset_d$ = 60°) over 20 runs compared to other algorithms.

Evolutionary algorithm	**Best SLL (dB)**	**Worst SLL (dB)**	**Mean (dB)**	**STD (dB)**
JABC	-20.07	-19.87	-19.98	0.0517
ABC	-19.32	-18.69	-18.93	0.1826
JAYA	-19.87	-19.38	-19.64	0.1297
SCA	-18.14	-17.93	-18.01	0.0341
ALO [54]	-15.94	-15.66	-15.80	0.0865
Hybrid [54]	-16.19	-15.96	-16.06	0.0501
SOS [17]	-15.45	-	-15.93	0.0470
FA [53]	-15.38	-	-15.97	0.0686

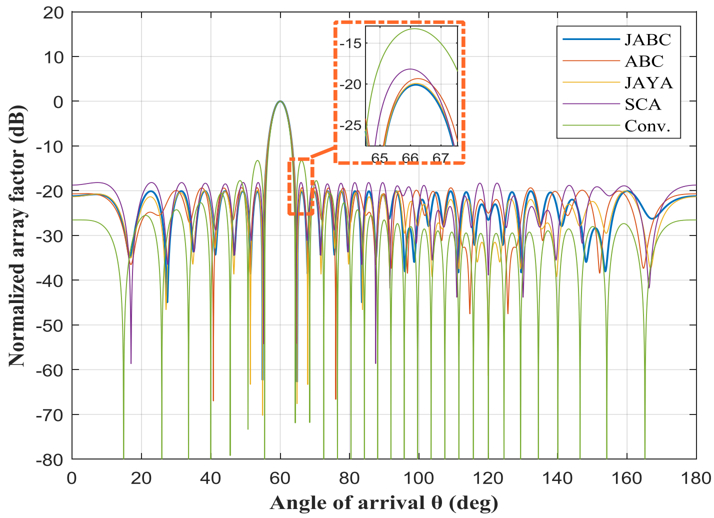

Figure 4.39: The radiation pattern obtained with $\emptyset_d$ = 60°, N=30.

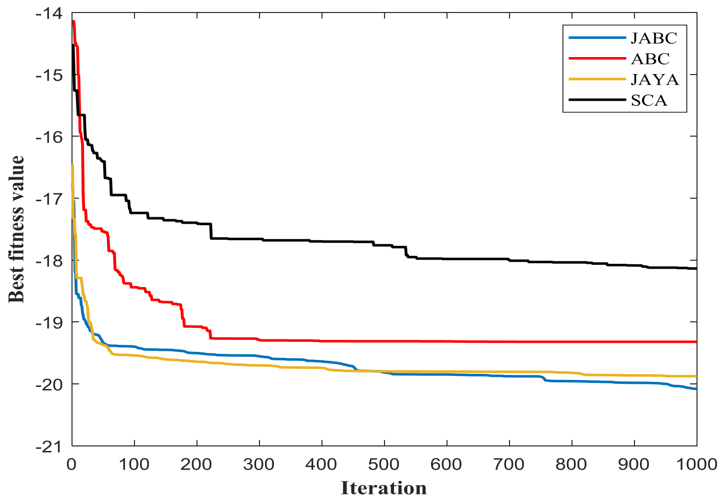

Figure 4.40: Convergence curve for 30-element LAA optimized over 1000 iterations.

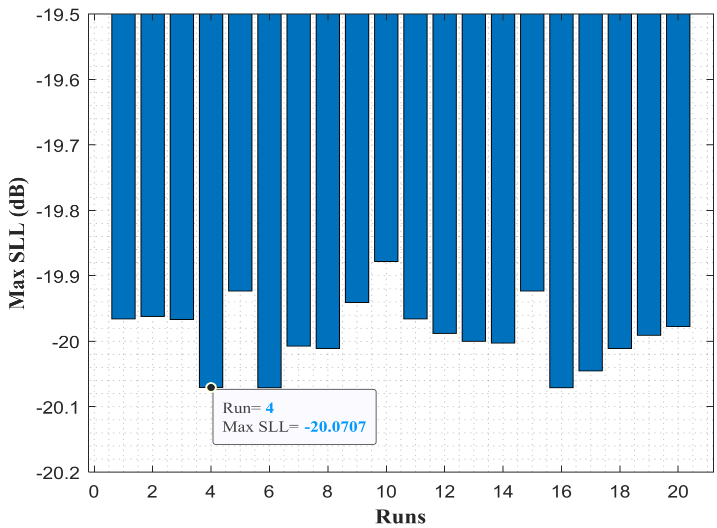

Figure 4.41: The maximum SLL obtained by JABC for 30 elements with $\emptyset_d$ = 60° in 20 independent trials.

4.2 Summary

In this study, the synthesis of a linear antenna array (LAA) was successfully accomplished using different optimization algorithms, namely JABC, ABC, JAYA, and SCA. The optimization process focused on determining the optimal values for the amplitude, position, and phase of the antenna elements, both with and without the constraint of maintaining a fixed First Null Beam Width (FNBW).

The results showed that the proposed algorithm, JABC, outperformed the other algorithms in terms of synthesizing the LAA. It achieved superior performance in terms of optimizing the antenna parameters and meeting the desired FNBW constraint. This suggests that JABC algorithm is more effective and efficient in finding the optimal solutions for LAA synthesis compared to ABC, JAYA, SCA, and other algorithms.

The successful synthesis of the LAA using JABC algorithm highlights the importance of using advanced optimization techniques to achieve desirable antenna performance. The findings of this study contribute to the field of antenna array design and provide valuable insights for future research in this area.

Chapter 5

CONCLUSIONS AND FUTURE SCOPE

5.1 Conclusions

Various optimization problems have gained the attention of researchers in recent years due to their complex structure. Finding a suitable optimization method to solve real-world problems is challenging. It needs powerful optimization search methods to be solved effectively. A novel hybrid variant of ABC clubbed with JAYA, namely JABC algorithm, is proposed. The proposed algorithm has the added advantages of both ABC and JAYA, and is done by incorporating JAYA into the scout phase of ABC. The enhanced scout phase helps the algorithm in providing better convergence speed and hence avoids local optima stagnation. Apart from that, the addition of simulated annealing based mutation weight provides better exploration and keeps a balanced exploration as well as exploitation operation.

The simulation results are evaluated using CEC 2005 benchmark problems and synthesis of LAA. A total of 14 cases have been used for SLL reduction, for element, amplitude and phase optimization. Two statistical tests namely wilxocon's test and Freidman test have been done to check the significance of the algorithm statistically. For optimizing the element amplitudes, 10-element, 16-elemnt, and 24-element LAA's are used. We find that a maximum SLL without FNBW constraint of -28.94 dB at FNBW 34.42^o for 10-element, -37.30 dB at FNBW 25.17 ^{o} for 16-element, and -49.09 dB at FNBW 20.55^o for 24-element is achieved respectively, and is significantly better than MSOA, CSA, PSO, and others. To optimize positions, 10-element, and 28-element LAA's are used. Here we find that a maximum SLL without FNBW constraints of -23.36 dB at FNBW 38.6^o for 10-element, and -25.80 dB at FNBW 10.8^o for 28-element LAA is achieved. The third case uses phase optimization of a 40-element LAA where the peak SLL achieved is -18.18 dB at FNBW 6.2^o. The final case three different LAA configurations are optimized with different main lobe's steering angles (N = 20 with $\emptyset_d$ = 30°, N = 26 with $\emptyset_d$ =45, N = 30 with $\emptyset_d$ =60°), and we find that a maximum SLL equals -17.07 dB,-19.06 dB, and -20.07 dB for 20, 26, and 30 elements, respectively. Overall, in all the cases, the proposed JABC performs significantly better as compared to other algorithms such as MA, CS, CSO, PSO, ABC, and JAYA, among others. From these results, it has been concluded that the proposed algorithm outperformed all the metaheuristic optimization algorithms in most of the test functions, confirming the high exploitation, exploration, and convergence rate of the proposed hybrid algorithm and the capability to deal with high dimensions problems. Furthermore, the results prove the ability of our Hybrid algorithm to successfully overcome many drawbacks and combine the main features of ABC and JAYA algorithms. Consequently, both algorithms are utilized in proper hybridization.

5.2 Scope for future work

In order to further advance the research in this field, several potential avenues for future work have been identified and are suggested below:

1. JABC algorithm can be further enhanced by analyzing the impact of all of its parameters.
2. Instead of random initialization, using a local memory of past results or other techniques to select the initial values.
3. New mutation operators and inertia weight operators can be highlighted the proposed method can be investigated deeply to see the effect of each component and enhance it by other search operators.
4. The JABC methods can be used to solve other optimization problems such as feature selection, text clustering, thresholding of image segmentation, task scheduling-based cloud computing; text classification, photovoltaic parameter estimation, constrained optimization

problems, text summarization, big data application, image edge detection, networks applications, smart home management, and other industrial engineering problems.

5. In this thesis, the radiation pattern synthesis of antenna arrays was achieved for isotropic radiators. The choice of the array elements to be isotropic can be extended to other antenna types such as dipoles, horn, patch, ring …etc.
6. Synthesis of cylindrical antenna arrays, elliptical antenna array (EAA), circular antenna arrays (CAA), and concentric circular antenna arrays (CCAA) using JABC algorithm.

By addressing these suggested future works, researchers can advance the current findings, explore new possibilities, and contribute to the overall body of knowledge in the field of antenna array synthesis and optimization. These endeavors will pave the way for more efficient and effective solutions to real-world problems in antenna engineering.

REFERENCES

[1] C. A. Balanis, Balanis C.A., editor, “Modern antenna handbook”, Sep, 2011.

[2] D. E. Goldberg, and J. H. Holland, “Genetic algorithms and machine learning”, Machine learning, vol. 3, pp. 95–99, 1988.

[3] G. Sun, Y. Liu, H. Li, S. Liang, A. Wang, and B. Li, “An antenna array sidelobe level reduction approach through invasive weed optimization”, International Journal of Antennas and Propagation, vol. 2018, pp. 1–16, 2018.

[4] R. Storn, and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces”, Journal of global optimization, vol. 11, no. 4, pp.341–359, 1997

[5] R. Salgotra, and U. Singh, “The naked mole-rat algorithm”, Neural Computing and Applications, vol. 31, no. 12, pp. 8837–8857, 2019.

[6] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer”, Advances in engineering software, vol. 69, pp. 46–61, 2014.

[7] L. Abualigah, and A. Diabat, “Advances in sine cosine algorithm: a comprehensive survey”, Artificial Intelligence Review, vol. 54, no. 4, pp. 1–42, 2021.

[8] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, “Harris hawks optimization: Algorithm and applications”, Future generation computer systems, vol. 97, pp. 849–872, 2019.

[9] L. Abualigah, D. Yousri, M. Abd Elaziz, A. A. Ewees, M. A. Al-qaness, and A. H. Gandomi, “Aquila optimizer: A novel meta-heuristic optimization algorithm”, Computers & Industrial Engineering, vol.157, pp. 107250, 2021.

[10] C. Science, V. Vihar, R. Road, L. Abualigah, Abd Elaziz, M. Sumari, P. Geem, Z.W. and A.H. Gandomi, “Reptile Search Algorithm (RSA): A nature-inspired meta-heuristic optimizer”. Expert Systems with Applications, vol. 191, pp. 116158, no. 2, 2022.

[11] D. M. Burchell, T. A. Franz-odendaal, P. Joy, Agushaka, J.O., Ezugwu, A.E. and L. Abualigah, “Dwarf mongoose optimization algorithm”, Computer methods in applied mechanics and engineering, vol. 391, pp. 114570, 2022.

[12] B. J. A. Nelder, R. Meadf, Nelder, J.A. and R. Mead, “A simplex method for function minimization”. The computer journal, vol. 7, no. 4, pp.308-313, 1964.

[13] R. Salgotra, and U. Singh, “Application of mutation operators to flower pollination algorithm”, Expert Systems with Applications, vol. 79, pp. 112–129, 2017.

[14] H. Wang, C. Liu, H. Wu, B. Li, and X. Xie, “Optimal pattern synthesis of linear array and broadband design of whip antenna using grasshopper optimization algorithm”, International Journal of Antennas and Propagation, vol. 2020, jan 2020.

[15] P. Saxena, and A. Kothari, “Ant lion optimization algorithm to control side lobe level and null depths in linear antenna arrays”, AEU-International Journal of Electronics and Communications, vol. 70, No. 9, pp. 1339–1349, 2016.

[16] U. Singh, and R. Salgotra, “Synthesis of linear antenna array using flower pollination algorithm”, Neural Computing and Applications, vol. 29, pp. 435–445, 2018.

[17] N. I. Dib, “Design of linear antenna arrays with low side lobes level using symbiotic organisms search”, Progress In Electromagnetics Research B, Vol. 68, pp. 55–71, 2016.

[18] A. Sharaqa, and N. Dib, “Design of linear and elliptical antenna arrays using biogeography based optimization”, Arabian Journal for Science and Engineering, Vol. 39, pp. 2929–2939, 2014.

[19] M.M. Khodier, M. Al-Aqeel, “Linear and circular array optimization: A study using particle swarm intelligence”, Progress In Electromagnetics Research B, vol. 15, pp. 347–373, jan 2009.

[20] N. I. Dib, S. K. Goudos, and H. Muhsen, “Application of taguchi’s optimization method and self adaptive differential evolution to the synthesis of linear antenna arrays”, Progress In Electromagnetics Research, vol. 102, pp. 159–180, 2010.

[21] M. Khodier, “Comprehensive study of linear antenna array optimization using the cuckoo search algorithm”, IET Microwaves, Antennas & Propagation, vol. 13, No. 9, pp. 1325–1333, 2019.

[22] U. Singh, and R. Salgotra, “Synthesis of linear antenna arrays using enhanced firefly algorithm”, Arabian Journal for Science and Engineering, vol. 44, pp. 1961–1976, 2019.

[23] R. Salgotra, and U. Singh, “A novel bat flower pollination algorithm for synthesis of linear antenna arrays”, Neural Computing and Applications, vol. 30, pp. 2269–2282

[24] L. Merad, F. Bendimerad, and S. Meriah, “Design of linear antenna arrays for side lobe reduction using the tabu search method”, International Arab Journal of Information Technology (IAJIT), vol. 5, No. 3, 2008.

[25] A. S. Al-zoubi, A. A. Amaireh, and N. I. Dib, “Comparative and comprehensive study of linear antenna arrays’ synthesis”, International Journal of Electrical and Computer Engineering (IJECE), vol. 12, no. 3, pp. 2645–2654, Jun 2022

[26] Z. Luo, F. Liu, Z. Zou, S. Guo, and T. Shen, “Optimum design of both linear and planar

sparse arrays with sidelobe level reduction using salp swarm algorithm", Journal of Electromagnetic Waves and Applications, vol. 35, no. 5, pp. 690-704, 2021

[27] E. Kurt, S. Basbug, and K. Guney, "Linear antenna array synthesis by modified seagull optimization algorithm", The Applied Computational Electromagnetics Society Journal (ACES), vol. 36, no. 12, pp. 1552–1562, 2021.

[28] S. Liang, T. Feng, and G. Sun, "Sidelobe-level suppression for linear and circular antenna arrays via the cuckoo search–chicken swarm optimisation algorithm", IET Microwaves, Antennas & Propagation, vol. 11, no. 2, pp. 209-218, 2016.

[29] A. B. Gabis, Y. Meraihi, S. Mirjalili, and A. Ramdane-cherif, "A comprehensive survey of sine cosine algorithm: variants and applications", Artificial Intelligence Review, 2021 Oct; vol. 54, no. 7, pp. 5469-5540, 2021

[30] S. Mirjalili, " SCA: a sine cosine algorithm for solving optimization problems", Knowledge-based systems, vol. 96, pp. 120-133, Mar 2016.

[31] D. Karaboga, "An idea based on honey bee swarm for numerical optimization", Tech. report-tr06, Erciyes university, engineering faculty, computer, vol. 200, pp. 1-10, 2005.

[32] Q. Qin, S. Cheng, Q. Zhang, L. Li, and Y. Shi, "Artificial bee colony algorithm with time-varying strategy", Discrete Dynamics in Nature and Society, 2015.

[33] H. Narasimhan, "Parallel artificial bee colony (pabc) algorithm", 2009 World congress on nature & biologically inspired computing (NaBIC), IEEE, pp. 306–311, 2009.

[34] Y. Huo, Y. Zhuang, J. Gu, S. Ni, and Y. Xue, "Discrete gbest-guided artificial bee colony algorithm for cloud service composition", Applied Intelligence, vol. 42, pp. 661–678, 2015.

[35] S. S. Jadon, R. Tiwari, H. Sharma, and J. C. Bansal, "Hybrid artificial bee colony algorithm with differential evolution", Applied Soft Computing, vol. 58, pp. 11-24, 2017.

[36] B. Akay, D. Karaboga, B. Gorkemli, and E. Kaya, "A survey on the artificial bee colony algorithm variants for binary, integer and mixed integer programming problems", Applied Soft Computing, vol. 106, pp. 107351, 2021.

[37] A. Sharma, A. Sharma, S. Choudhary, R. K. Pachauri, A. Shrivastava, and D. Kumar, "A review on artificial bee colony and it's engineering applications", Journal of Critical Reviews, vol. 7, no. 11, pp. 4097–4107, 2020.

[38] E. Kaya, B. Gorkemli, B. Akay, and D. Karaboga, "A review on the studies employing artificial bee colony algorithm to solve combinatorial optimization problems", Engineering Applications of Artificial Intelligence, vol. 115, pp. 105311, 2022.

[39] R. Ahmad, N. Akhtar, R. Ahmad, and N. Akhtar, "Variants of Artificial Bee Colony algorithm and its applications in medical image processing", Applied soft computing, vol. 97, pp. 106799, 2020.

[40] D. Karaboga, and B. Basturk, "On the performance of artificial bee colony (abc) algorithm", Applied soft computing, vol. 8, no. 1, pp. 687–697, 2008.

[41] R. Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems", International Journal of Industrial Engineering Computations, vol. 7, No. 1, pp. 19–34, 2016.

[42] P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y.-P. Chen, A. Auger, and S. Tiwari, "Problem definitions and evaluation criteria for the cec 2005 special session on real-parameter optimization", KanGAL report, pp. 2005005, 2005.

[43] K. Yan and Y. Lu, "Sidelobe reduction in array-pattern synthesis using genetic algorithm", IEEE Trans. Antennas Propag., vol. 45, no. 7, pp. 1117–1122, Jul. 1997.

[44] Y. Tian, S. Member, and J. Qian, "Improve the performance of a linear array by changing the spaces among array elements in terms of genetic algorithm", IEEE Trans. Antennas Propag., vol. 53, no. 7, pp. 2226–2230, Jul. 2005.

[45] R. K. Jeffers and S. G. Boucher, "Optimization of linear antenna array using firefly algorithm", Int. J. Sci. Eng. Res., vol. 4, no. 8, pp. 601–606, 2013.

[46] M. A. Zaman and A. Matin, "Nonuniformly spaced linear antenna array design using firefly algorithm", Int. J. Microw. Sci. Technol., vol. 2012, pp. 1–8, Apr. 2012.

[47] S. Jayaprakasam, S. Kamal, A. Rahim, and C. Y. Leow, "Sidelobe reduction and capacity improvement of open-loop collaborative beamforming in wireless sensor networks", PLoS ONE, vol. 12, no. 5, May 2017.

[48] H. Wang, C. Liu, H. Wu, B. Li, and X. Xie, "Optimal pattern synthesis of linear array and broadband design of whip antenna using grasshopper optimization algorithm", International Journal of Antennas and Propagation, 2020.

[49] A. Das, D. Mandal, S. Ghoshal, and R. Kar, "Moth flame optimization based design of linear and circular antenna array for side lobe reduction", *International Journal of Numerical Modelling: Electronic Networks Devices and Fields*, Vol. 32, No. 1, pp. e2486, 2019.

[50] L. Liang, J. Sun, H. Li, J. Liu, Y. Jiang, and J. Zhou, "Research on side lobe suppression of time-modulated sparse linear array based on particle swarm optimization", Int. J. Antennas Propag., vol. 2019, pp. 1–14, Aug. 2019.

[51] S. Liang, Z. Fang, G. Sun, Y. Liu, G. Qu, and Y. Zhang, "Sidelobe reductions of antenna

arrays via an improved chicken swarm optimization approach", IEEE Access, vol. 8, pp. 37664–37683, 2020.

[52] Owoola, E.O., Xia, K., Wang, T., Umar, A. and Akindele, R.G., 2021. Pattern synthesis of uniform and sparse linear antenna array using mayfly algorithm", IEEE Access, vol. 9, pp. 77954-77975, 2021.

[53] B. Basu, and G. Mahanti, "Fire Fly and Artificial Bees Colony Algorithm for Synthesis of Scanned and Broadside Linear Array Antenna", Progress In Electromagnetics Research B, vol. 32, pp. 169-190, 2011.

[54] A. A. Amaireh, A. S. Al-zoubi, and N. I. Dib, "Amaireh AA, Al-Zoubi AS, Dib NI. The optimal synthesis of scanned linear antenna arrays", International Journal of Electrical and Computer Engineering (IJECE), vol. 10, no. 2, pp. 1477–1484, Apr. 2020.

[55] P. Lakshman, and D. Ghosh, "Linear antenna array synthesis using cat swarm optimization", AEU International Journal of Electronics and Communications, vol. 68, No. 6, pp. 540-549, 2014.

[56] A.A., Al-Azza, Al-Jodah, A.A., and Harackiewicz F.J., "Spider monkey optimization: A novel technique for antenna optimization". IEEE Antennas and Wireless Propagation Letters, vol. 15, pp.1016–1019, 2015.

[57] U. Singh and R. Salgotra, "Optimal synthesis of linear antenna arrays using modified spider monkey optimization", Arabian Journal for Science and Engineering, vol. 41, no. 8, pp. 2957– 2973, 2016.

[58] N. Vegesna, Y. Govindarajan, and S. K. Terlapu, "Linear Antenna Array synthesis using Rao and Jaya algorithms", vol. 26, no.1, June 2022.

[59] A. Durmus, "Novel metaheuristic optimization algorithms for sidelobe suppression of linear antenna array". In 2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), pp. 291-294, Oct. 2022.

[60] A. Amaireh and N. Dib, "Design of Linear Antenna Arrays Using Antlion and Grasshopper Optimization Algorithms", Oct. 2017.

[61] R. Salgotra, U. Singh, S. Singh, and N. Mittal, "A hybridized multi-algorithm strategy for engineering optimization problems", Knowledge-Based Systems, vol. 217, pp. 106790, 2021.

[62] R. Salgotra, U. Singh, and S. Saha, "New cuckoo search algorithms with enhanced exploration and exploitation properties", Expert Systems with Applications, vol. 95, pp. 384–420, 2018.

[63] X. Yang, “Firefly algorithms for multimodal optimization”, in: International symposium on stochastic algorithms, Springer, pp. 169–178, 2009.

[64] D. Simon and S. Member, “Biogeography-based optimization”, IEEE transactions on evolutionary computation, vol. 12, no. 6, pp. 702–713, 2008.

[65] J. Derrac, S. Garc´ıa, D. Molina, and F. Herrera, “A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms”, Swarm and Evolutionary Computation, vol. 1, No. 1, pp. 3–18,2011.

[66] G. G. Tejani, V. J. Savsani, V. K. Patel, and S. Mirjalili, “Truss optimization with natural frequency bounds using improved symbiotic organisms search”, Knowledge-Based Systems vol. 143, pp. 162– 178, 2018.

[67] B. Li, C. Liu, H. Wu, Y. Zhao, and Y. Dong, “Chaotic adaptive butterfly mating optimization and its applications in synthesis and structure optimization of antenna arrays”, International Journal of Antennas and Propagation, vol. 2019, 2019.

[68] A. Sharaqa, “Biogeography-based optimization method and its application in electromagnetics”, master thesis, Jorden University for Science and Technology, 2012.

[69] R. A. Formato, “Al-Kubati G. Central force optimization method and its application to the design of antennas. Master Thesis, Jordan University of Science and Technology. July 2009.