

A Comparison of Swarm-Based Optimization Algorithms in Linear Antenna Array Synthesis

Ali Durmus (✉ alidurmus@kayseri.edu.tr)

University of Kayseri <https://orcid.org/0000-0001-8283-8496>

Rifat KURBAN

Kayseri Universitesi

Ercan KARAKOSE

Kayseri Universitesi

Research Article

Keywords: Linear antenna array synthesis, side lobe level, particle swarm optimization, artificial bee colony algorithm, mayfly algorithm, jellyfish search algorithm

Posted Date: February 17th, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-233415/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

A Comparison of Swarm-Based Optimization Algorithms in Linear Antenna Array Synthesis

Ali DURMUS^{1*}, Rifat KURBAN², Ercan KARAKOSE³

¹ *Department of Electricity and Energy, Vocational College, Kayseri University, Turkey.

² Department of Computer Technologies, Vocational College, Kayseri University, Turkey.

³ Department of Natural Sciences, Engineering & Architecture and Design Faculty, Kayseri University, Turkey

Abstract

Today, the design of antenna arrays is very important in providing effective and efficient wireless communication. The purpose of antenna array synthesis is to obtain a radiation pattern with low side lobe level (SLL) at a desired half power beam width (HPBW) in far-field. The amplitude and position values of the array elements can be optimized to obtain a radiation pattern with suppressed SLLs. In this paper swarm-based meta-heuristic algorithms such as Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC), Mayfly algorithm (MA) and Jellyfish Search (JS) algorithms are compared to realize optimal design of linear antenna arrays. Extensive experiments are conducted on designing 10, 16, 24 and 32-element linear arrays by determining the amplitude and positions. Experiments are repeated 30 times due to the random nature of swarm-based optimizers and statistical results show that performance of the novel algorithms, MA and JS, are better than well-known methods PSO and ABC.

Keywords: Linear antenna array synthesis, side lobe level, particle swarm optimization, artificial bee colony algorithm, mayfly algorithm, jellyfish search algorithm

Corresponding author: A. Durmus; email: alidurmus@kayseri.edu.tr

1. Introduction

With the introduction of Industry 4.0 in daily life, the Internet of Things (IoT) has gained importance and with this, the need for wireless communication has increased in every field. In order to meet this need effectively and efficiently, advanced antenna designs are used. It is very difficult to obtain the desired radiation pattern with a single antenna. Therefore, antenna arrays are formed by bringing together more than one antenna. In this way, both routing and communication demand over long distances are provided effectively. Antennas with different geometries are designed to obtain the desired radiation pattern. Antennas take names such as linear, circular, elliptical, and rectangular according to their geometric structures [1].

In the design of an effective and efficient antenna array, it is aimed to suppress the side lobe levels (SLL) in the radiation pattern and to make the half power beam width (HPBW) as narrow as possible. By preventing electromagnetic interference in the environment, the side lobes in the radiation pattern are suppressed in order to receive only the data in the desired direction and the HPBW is narrowed in order to communicate with less loss with long distances [2].

There are several studies in the literature for linear antenna array (LAA) synthesis, which is one of the most studied topics in electromagnetic problems. In antenna array synthesis, generally three parameters are adjusted by optimization algorithms to obtain the desired SLL and HPBW. These are the amplitudes, positions and phases of the array elements. Different methods are used to determine these parameter values in antenna arrays. Classical methods are generally derivative based and these methods have been replaced by meta-heuristic algorithms, which are faster and more flexible methods, due to their difficulties in computation. There are several meta-heuristic optimization techniques for LAA synthesis in the literature. A plant growth simulation algorithm is used to pattern nulling of LAAs by amplitude only control in [3]. In [4] Dib et.al. proposed a symbiotic organisms search (SOS) technique to synthesis of LAAs. A biogeography-based optimization (BBO) is utilized for design of linear and elliptical antenna arrays [5]. In [6], Khodier et.al. proposed particle swarm optimization (PSO) algorithm for linear and circular array design. Taguchi's optimization method and self-adaptive differential evolution are used for designing of LAAs in [7]. Das et.al implemented moth flame optimization (MFO) to the synthesis of linear and circular antenna array for side lobe reduction [8]. In [9], a runner-root algorithm is applied to control side lobe level and null depths in LAAs. Artificial Bee Colony (ABC) algorithm is utilized to design the LAA [10, 11]. Cuckoo search (CS) algorithm is used to design LAA by minimizing the maximum side lobe level with and without null steering [12]. Pattern synthesis of LAA is realized by using improved differential evolution algorithm in [13]. Three kinds of antenna arrays such as LAA, circular antenna array (CAA) and random antenna array (RAA) are studied in [14] with Chicken Swarm Optimization (CSO). The strawberry algorithm (SBA) as an optimization tool has been applied to antenna array synthesis and faster convergence characteristic is obtained in addition to improvement in side-lobe suppression and prescribed null placements [15].

In this study, LAAs with different number of elements are considered. The synthesis of sequences with 10, 16, 24 and 32-elements, which are widely used in the literature, is carried out by four different swarm-based optimization methods. Two of these meta-heuristic methods are PSO and

ABC, which are used in the literature in several studies, while the other two novel optimization methods, Mayfly algorithm (MA) and Jellyfish search algorithm (JS), have been used in antenna array synthesis for the first time. The common side of these algorithms is that they are all inspired from the social behavior of different such as animal, fish, insect and bee swarms. To test the performance of these meta-heuristic methods in LAA synthesis; 10, 16, 24 and 32-elements is examined. By using these four algorithms, the amplitude and position values of LAA elements with different numbers of elements have been found optimally. In addition, the values obtained by the algorithms such as SLL and CPU timing are statistically compared.

The rest of this paper is organized as follows: the problem formulation is examined in Section 2. In Section 3, the swarm based meta-heuristic optimization methods are briefly described. Numerical results and comparative statistical data obtained with PSO, ABC, MA and JS are given in Section 4. Finally, Section 5 concludes the paper.

2. Linear Array Model and Problem Formulation

The configuration of the linear array with $2M$ elements placed symmetrically along the x-axis is shown in Figure 1. With reference to the origin point, a total of $2M$ elements of the array are placed equally in both regions of the x-axis as M elements. The array factor (AF) expression of the LAA with symmetrical M elements positioned along the x axis is given as follows.

$$AF(\theta) = 2 \sum_{n=1}^M I_n \cos(kd_n \sin \theta + \alpha_n) \quad (1)$$

where α_n , I_n and d_n are phase, excitation amplitude and position weight of n th element in array, respectively. The scanning angle is θ . k represents the wave number and is formulated as $k = 2\pi/\lambda$. The total number of elements in the antenna array is $2M$, but since these antennas are symmetrical, the parameter to be optimized is equal to half of the total number of antennas, M . In this study, α_n phase values are taken as zero.

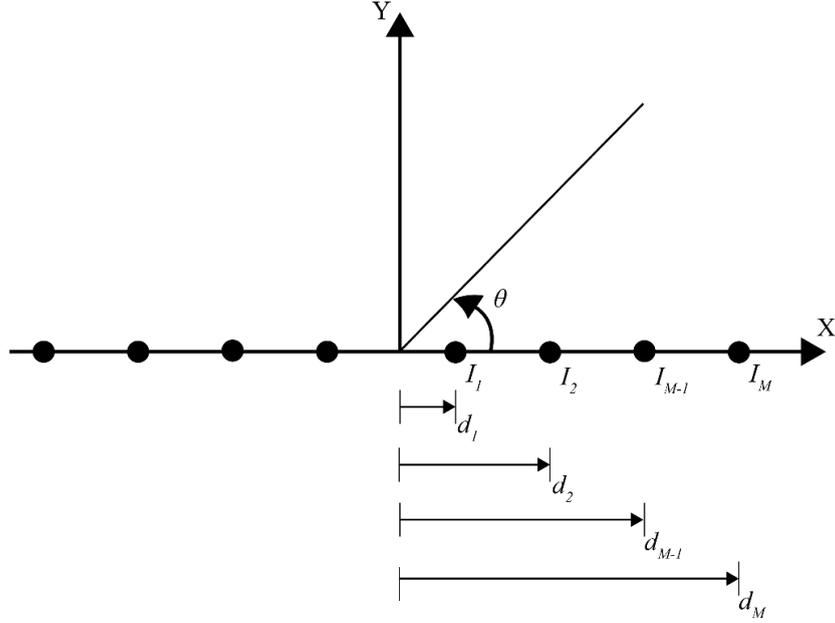


Fig. 1. Geometry of M -element LAA.

The position and excitation amplitude values of the LAAs with the desired radiation patterns are optimally determined by four swarm-based algorithms which namely PSO, ABC, MA, JA. The purpose of antenna design is to transfer data to long distances without being affected by the electromagnetic pollution in the environment. Therefore, it is very important to suppress unwanted interference in the environment. In order to prevent these attempts, the side lobe levels of the radiation patterns should be suppressed. The aim here is to obtain a radiation model with a minimum SLL value. HPBW value is also included in the fitness function and this value has been determined as constant in all simulation studies. To achieve the desired goal, the fitness function can be defined as follows.

$$Fitness_{function} = \begin{cases} Inf & \text{if } HPBW_{current} > HPBW_{desired} \\ f_{SLL} & \text{else} \end{cases} \quad (2)$$

where $HPBW_{current}$ is the HPBW of the potential solution produced by the optimization algorithms, $HPBW_{desired}$ is the desired HPBW for the specified array, Inf is a penalty value and f_{SLL} is the function formulated as follows,

$$f_{SLL} = \max[AF(\theta)] \quad (3)$$

3. Swarm Based Optimization Techniques

3.1. Particle Swarm Optimization (PSO)

PSO is a swarm intelligence algorithm based on the population-based movement of birds or fish and used to solve optimization problems developed in 1995. In PSO, a search field is defined and the position of a moving particle in that field represents the solution to the problem. In problem solving, each particle changes its position according to its choice and the choice of the community with which it moves together. The position of any particle is changed by adding a

speed to its position. Thus, the velocity of each particle depends on its previous most optimum position and the previous most optimal position of the flock to which it is attached. The advantages of PSO can be briefly summarized as; the derivative is not calculated so that the solution is not stuck at a local optimum where the derivative is zero, the members in the group share information constantly between each other, and the knowledge of the best solutions is used by all members. More information about the PSO algorithm can be found [16].

3.2. Artificial Bee Colony (ABC) Algorithm

The artificial bee colony (ABC) algorithm is a flock of intelligence methods that have been successfully applied to many optimization problems and proposed in 2007. In the ABC algorithm, bees are divided into three classes called employed, onlooker, and scout. The ABC algorithm is inspired by the foraging behavior of bees. Employing bees take advantage of specific food sources and are on the move to find better food sources. While the onlooker bees watch the movements of the working bees to select their food sources, the scout bees seek out random food sources. In this algorithm, bees using the food source represent an existing solution and lead to the solution of new problems by simulating how the food source will be consumed in the audience bee community. The working logic of the ABC algorithm has four stages. At each renewal, these steps are performed sequentially by running and the iteration repeats until they reach the goal. You can find details about ABC method in Reference [17].

3.1. Mayfly Algorithm (MA)

This algorithm is a novel metaheuristic method which inspired by the mating processes it uses to maintain mayflies generations, proposed in 2020. Mayflies' lifespan is only a few days, and according to this algorithm, mayflies become adult after hatching and the strongest mayflies are considered to survive. A potential solution for the any problem corresponds to the position of any mayfly in the search space. In the algorithm, two different clusters corresponding to male and female populations are formed. That is, each mayfly male or female is randomly placed into the problem space in an area represented by ad-dimensional vector $x = (1, \dots, xd)$. The x vector performance is evaluated on the predetermined target function $f(x)$. Simply, the velocity of a mayfly can be defined as the change in position $v = (v1, \dots, vd)$. The flight orientation of a mayfly is expressed as a complex that includes both the collective and individual flight preferences of the flies. That is, each mayfly orientation is taken and used as the individual best position (*best*) and the e good position (*gbest*) in swarm movement. Readers can find detailed information about MA in Ref. [18].

3.2. Jellyfish Search (JS) Algorithm

The JS algorithm, inspired by the behaviour of jellyfish in the ocean, is one of the newest swarm intelligences meta-heuristics methods for optimization problems, developed in 2020. The behaviour simulation of the jellyfish in the swarm includes the movement of the jellyfish in the flock, the instant tracking of the sea current and the time control mechanism for movements. JS provides an optimum convergence by balancing the exploration and utilization of the search area. The jellyfish's movement in the sea is based on searching for food. For this reason, jellyfish constantly move towards areas with more food. Ref. [19] contains detailed information about the JS optimization method.

4. Numerical Results

In this study, the amplitude and position of LAAs with 10, 16, 24 and 32 elements are optimally determined by four different swarm-based optimization methods and the values found are compared statistically. The common side of the selected PSO, ABC, MA and JS algorithms is that they are all swarm-based. The reason why PSO and ABC are preferred in simulation studies is that these algorithms are the most used optimization technique in the literature. MA and JS are optimization techniques recently presented to the literature and in this study these algorithms have been applied to antenna array synthesis for the first time.

The purpose of all experiments is to reach radiation patterns with lower SLL values at a desired HPBW. Experimental simulations are realized by using MATLAB software and a personal computer with 16 GB of RAM and a 2.6 GHz i7 processor.

For all algorithms the population size is set as 50 and the maximum iteration number is set as 500. The remaining internal parameters of the optimization algorithms used in the experiments are given in Table 1. These parameters are the suggested default values from the creators of the original algorithms.

Table 1. Parameter settings of the optimization algorithms.

Algorithm	Parameters
<i>PSO</i>	$w=0.5$, $w_{damp}=1$, $c_1=1$, $c_2=1$
<i>ABC</i>	onlooker_count=25, max_acc=1
<i>MA</i>	$g=0.5$, $g_{damp}=1$, $a_1=1$, $a_2=1.5$, $a_3=1.5$, $\beta=2$, dance=1, $f_1=1$, $dance_{damp}=0.8$, $f_{1damp}=0.99$, $\mu=0.01$
<i>JS</i>	No parameters

Since the following HPBW values are commonly chosen in the majority of the studies in literature, the desired HPBW are determined as 12.5°, 9°, 6 ° and 3.5° for 10, 16, 24 and 32-elements of arrays, respectively.

Due to the random nature of swarm-based optimization algorithms, all experiments are repeated 30 times and the average, best, worst conditions and the standard deviation of all runs are reported.

4.1. Amplitude only Design of Optimal LAA

In the first example, the amplitudes of the 10, 16, 24 and 32-element symmetric LAA array are determined by using PSO, ABC, MA and JS methods to perform the optimum design. Figure 2 shows the radiation patterns obtained with PSO, ABC, MA and JS. The suppressed SLL and CPU times obtained by using PSO, ABC, MA and JS are listed in Table 2. As can be clearly seen from Table 2 and Figure 2, for 10-element LAA all algorithms performed same in terms of mean SLL values except ABC. JS is ranked 1st by means of CPU time consumption solving the problem

in 3.1 seconds. PSO, MA, ABC is ranked 2nd, 3rd and 4th by solving the problem in 9.7, 13.3 and 14.2 seconds, respectively. For 16-element LAA, algorithms are ranked from best to worst as MA, JS, PSO and ABC in terms of mean SLL and MA, JS, ABC, PSO in terms of standard deviation of SLL. JS, PSO, ABC and MA solved the problem in 4.4, 10.2, 15.6 and 17.7 seconds. For 24-element LAA, algorithms are ranked from best to worst as MA, JS, PSO and ABC in terms of mean SLL and MA, ABC, JS, PSO in terms of standard deviation of SLL. JS, PSO, ABC and MA solved the problem in 6.1, 11.9, 18.4 and 23.7 seconds. For 32-element LAA, algorithms are ranked from best to worst as JS, MA, PSO and ABC in terms of mean SLL and JS, MA, ABC, PSO in terms of standard deviation of SLL. JS, PSO, ABC and MA solved the problem in 8.1, 13.7, 21.1 and 29.3 seconds. As a result, for amplitude only design of optimal 10, 16 and 24-element LAA, MA performed the best among the other algorithms in terms of mean SLL and standard deviation of SLL for 30 independent runs. However, MA is the slowest algorithm among the other algorithms in terms of CPU time consumption. JS performed second for 16 and 24-element LAA in terms of mean SLL and standard deviation of SLL for 30 independent runs however JS performed the best for 32-element LAA design both for mean and standard deviation of SLL. Furthermore, in all simulations JS is the fastest algorithm compared to PSO, ABC and MA. Best amplitude values obtained by PSO, ABC, MA and JS methods for 10, 16, 24 and 32-elements are tabulated in Table 3.

Table 2. Statistical SLL and CPU time comparison of PSO, ABC, MA and JS for amplitude only design of optimal LAA for 10, 16, 24 and 32-elements.

	10-element				16-element			
	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>
Best SLL (dB)	-26.9772	-26.9504	-26.9772	-26.9772	-40.2334	-39.7988	-40.2472	-40.2471
Mean SLL (dB)	-26.9772	-26.9058	-26.9772	-26.9772	-39.7041	-38.8234	-40.2472	-40.1986
Worst SLL (dB)	-26.9767	-26.8063	-26.9772	-26.9764	-35.9071	-37.8194	-40.2472	-39.1132
Std. Dev. SLL (dB)	0.0001	0.0341	0.0000	0.0002	0.8996	0.4736	0.0000	0.2065
Best CPU Time (s)	7.7581	13.8050	12.6659	3.1369	9.5540	14.7093	16.7023	4.1206
Mean CPU Time (s)	9.7063	14.2248	13.3780	3.3126	10.1806	15.6192	17.7317	4.3870
Worst CPU Time (s)	10.8955	14.8900	14.0229	3.6738	10.9683	16.4124	18.6868	4.7096
Std. Dev. CPU Time (s)	0.5090	0.2599	0.3097	0.1286	0.4174	0.5498	0.6040	0.1951
	24-element				32-element			
	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>
Best SLL (dB)	-40.5498	-39.7922	-41.2720	-40.9730	-22.7572	-22.4181	-23.2519	-23.1101
Mean SLL (dB)	-38.6388	-38.6031	-41.0523	-38.6710	-21.9665	-21.9189	-22.8378	-22.8525
Worst SLL (dB)	-33.0826	-37.3883	-40.0423	-32.1808	-21.0074	-21.5110	-22.2075	-22.3647
Std. Dev. SLL (dB)	1.7683	0.6543	0.2913	1.7260	0.4946	0.2199	0.2187	0.1526
Best CPU Time (s)	11.5868	17.9760	23.0663	5.9064	12.8456	19.9022	27.2495	7.5898
Mean CPU Time (s)	11.9383	18.3983	23.7065	6.0734	13.6918	21.0755	29.3229	8.1332
Worst CPU Time (s)	14.0326	20.0984	26.6039	6.6417	15.0637	23.0044	31.6852	8.8115
Std. Dev. CPU Time (s)	0.4680	0.4938	0.9999	0.1594	0.5511	0.8156	1.1763	0.3115

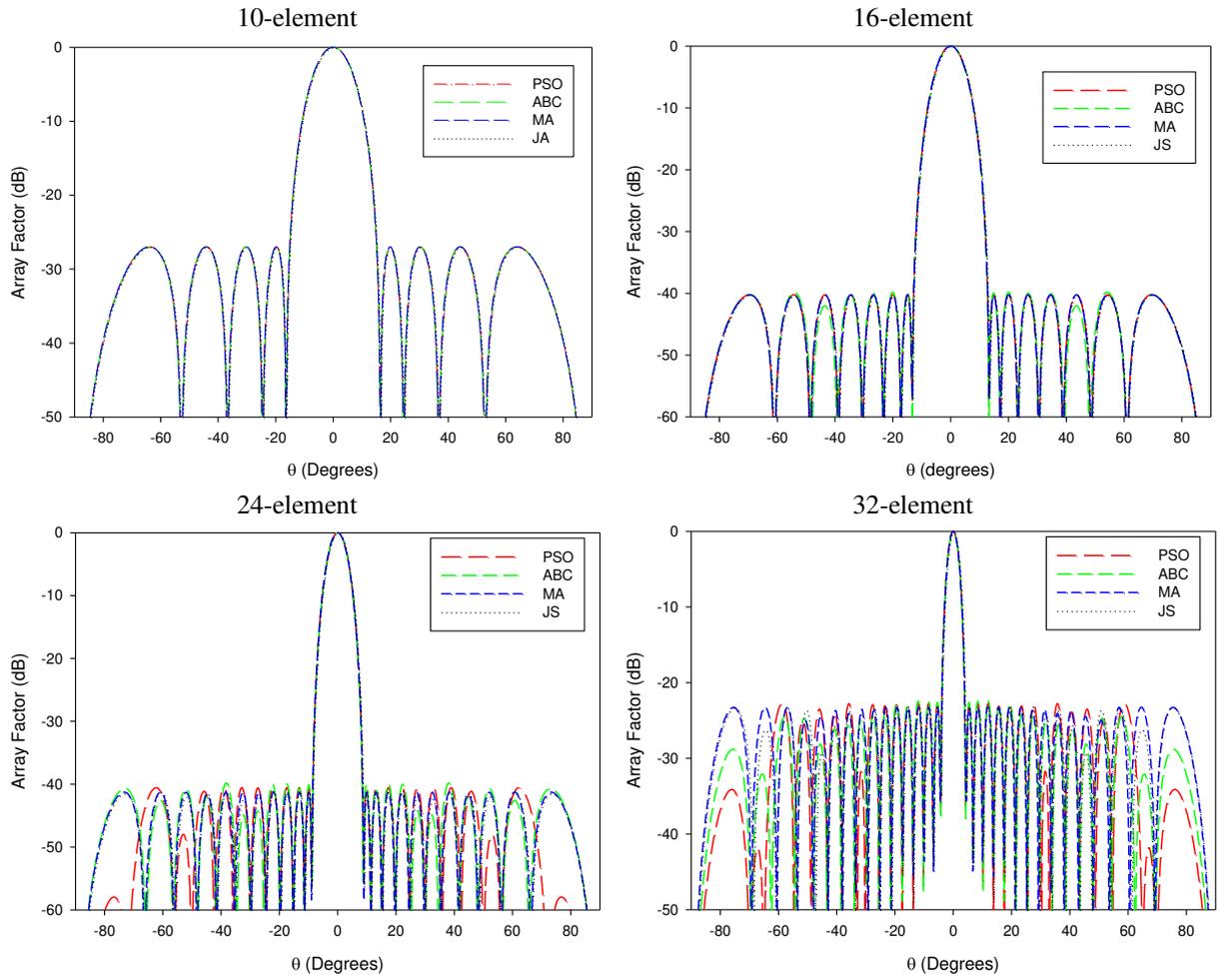


Fig. 2. Best radiation patterns for 10, 16, 24 and 32-element for amplitude only optimal LAA design.

Table 3. The best optimized amplitude values of LAA using PSO, ABC, MA and JS methods for 10, 16, 24 and 32-elements.

10-element	PSO	[0.8900 0.7928 0.6231 0.4220 0.2950]
	ABC	[0.8859 0.7895 0.6204 0.4204 0.2939]
	MA	[0.9837 0.8763 0.6888 0.4664 0.3261]
	JS	[0.8886 0.7916 0.6222 0.4213 0.2945]
16-element	PSO	[0.9992 0.9342 0.8147 0.6591 0.4900 0.3294 0.1941 0.1114]
	ABC	[1.0000 0.9316 0.8166 0.6597 0.4891 0.3322 0.1953 0.1108]
	MA	[0.8406 0.7859 0.6853 0.5544 0.4122 0.2770 0.1632 0.0936]
	JS	[0.9515 0.8896 0.7757 0.6276 0.4665 0.3135 0.1847 0.1060]
	PSO	[0.8934 0.8600 0.8219 0.7357 0.6627 0.5556 0.4607 0.3594 0.2726 0.1818 0.1362 0.0889]

24-element	ABC	[1.0000 0.9776 0.9121 0.8359 0.7436 0.6368 0.5135 0.4047 0.3097 0.2208 0.1396 0.1095]
	MA	[0.9003 0.8741 0.8224 0.7503 0.6625 0.5640 0.4626 0.3633 0.2708 0.1906 0.1238 0.0993]
	JS	[0.8931 0.8667 0.8163 0.7426 0.6579 0.5610 0.4581 0.3599 0.2697 0.1905 0.1230 0.0973]
32-element	PSO	[0.6700 0.7960 0.6850 0.6937 0.6008 0.7223 0.5845 0.5345 0.5220 0.5177 0.4727 0.3034 0.4385 0.2449 0.4295 0.5994]
	ABC	[1.0000 1.0000 0.9863 1.0000 0.8747 0.9413 0.7950 0.8109 0.7527 0.6417 0.6608 0.5093 0.5174 0.4444 0.5696 0.8410]
	MA	[0.7611 0.7519 0.7299 0.7333 0.6923 0.6630 0.6431 0.5801 0.5626 0.5083 0.4555 0.4302 0.3451 0.3391 0.2942 0.7365]
	JS	[0.5929 0.5858 0.5860 0.5705 0.5142 0.5672 0.4624 0.4725 0.4499 0.3727 0.3748 0.3433 0.2509 0.2700 0.2631 0.5525]

Averaged converge curves of PSO, ABC, MA and JS of 30 independent runs are illustrated in Figure 3 for 10, 16, 24 and 32-element amplitude only optimization of LAA. As can be seen from Figure 3, for 10-element LAA all algorithms are converged the best result in 200 iterations except ABC. For 16-element LAA, PSO and MA converged early however JS caught after 350 iterations. For 24-element LAA, MA converged smoothly compared to other algorithms. For 32-element MA converged early however JS took the lead at the end.

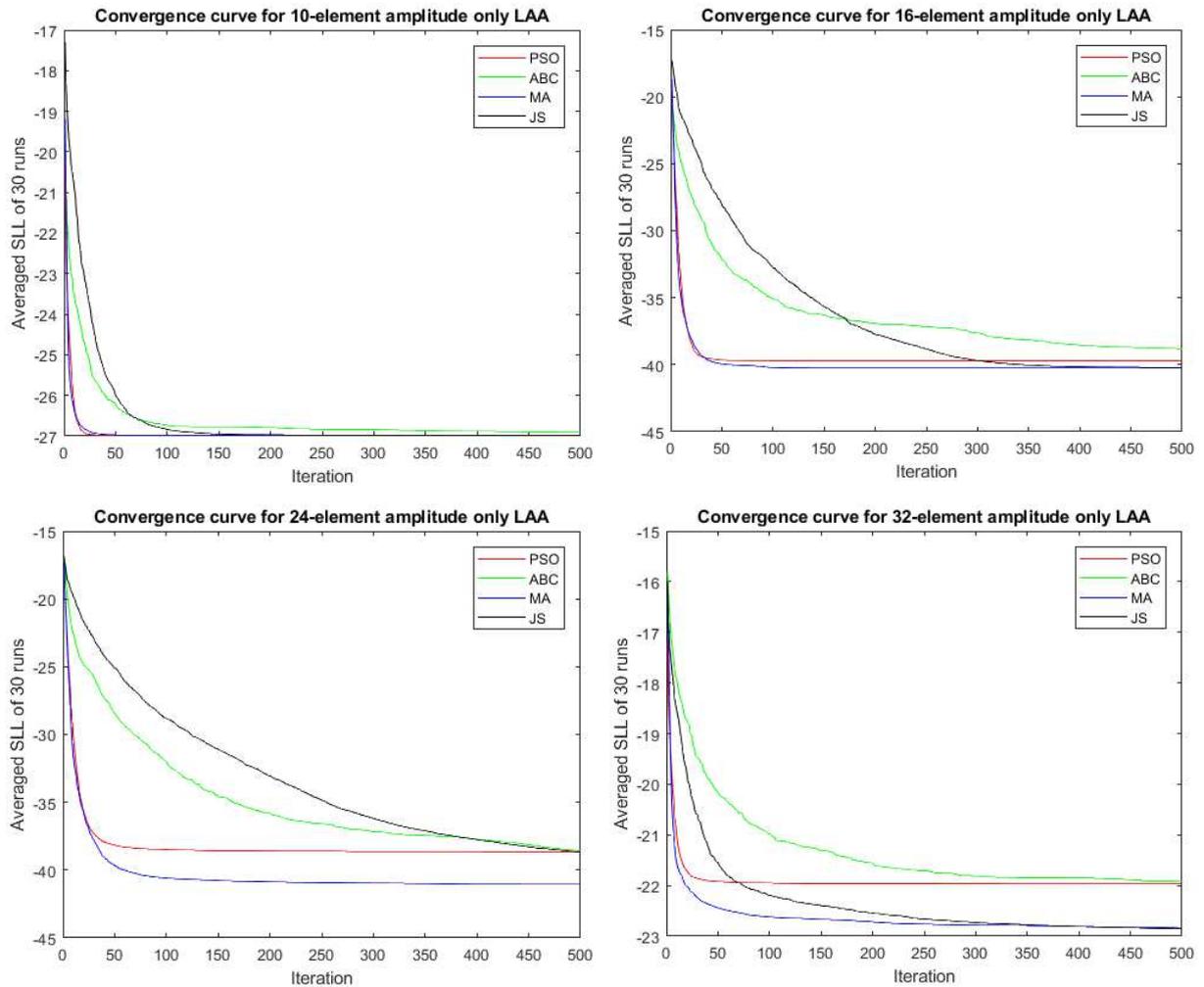


Fig. 3. Averaged converge curves of PSO, ABC, MA and JS for amplitude only design of LAA with 10, 16, 24 and 32-elements.

4.2. Position only Design of Optimal LAA

In the second example, the positions of the 10, 16, 24 and 32-element symmetric LAA are determined by using PSO, ABC, MA and JS methods to perform the optimum design. Figure 4 shows the radiation patterns obtained with PSO, ABC, MA and JS. The suppressed SLL and CPU times obtained by using PSO, ABC, MA and JS are listed in Table 4. As can be clearly seen from Table 4 and Figure 4, for element sizes algorithms are ranked as MA, JS, PSO and ABC from best to worst in terms of mean SLL values for 30 runs. By means of CPU times, algorithms are ranked as JS, PSO, ABC and MA from best to worst. For position only design of optimal LAA design, MA performed the best among the other algorithms in terms of mean SLL. However, MA is the slowest algorithm among the other algorithms in terms of CPU time consumption. JS performed second in terms of mean SLL however in all simulations JS is the fastest algorithm compared to PSO, ABC and MA. Best position values obtained by PSO, ABC, MA and JS methods for 10, 16, 24 and 32-elements are tabulated in Table 5.

Table 4. Statistical SLL and CPU time comparison of PSO, ABC, MA and JS for position only design of optimal LAA for 10, 16, 24 and 32-elements.

	10-element				16-element			
	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>
Best SLL (dB)	-21.3920	-21.1796	-21.3958	-21.4305	-24.5285	-22.7773	-24.3570	-23.5265
Mean SLL (dB)	-20.2687	-19.7332	-20.7130	-20.6417	-22.5863	-22.3006	-22.9624	-22.7405
Worst SLL (dB)	-19.7709	-17.6890	-19.0663	-19.0669	-19.9370	-21.3002	-20.1059	-21.2464
Std. Dev. SLL (dB)	1.0769	0.3094	0.9393	1.1319	1.0299	0.3189	0.8128	0.6167
Best CPU Time (s)	8.7570	12.8468	13.8618	2.9977	9.6604	14.6433	16.2801	4.2945
Mean CPU Time (s)	9.0856	13.6510	14.4751	3.1230	10.2636	15.5047	17.3993	4.5102
Worst CPU Time (s)	9.4691	14.6641	15.2326	3.3974	11.0657	16.3744	18.3763	4.7998
Std. Dev. CPU Time (s)	0.2058	0.3509	0.4181	0.0936	0.3686	0.5543	0.6421	0.1525
	24-element				32-element			
	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>	<i>PSO</i>	<i>ABC</i>	<i>MA</i>	<i>JS</i>
Best SLL (dB)	-26.2968	-23.8109	-26.4776	-25.1024	-24.6166	-23.7019	-25.6247	-25.2416
Mean SLL (dB)	-23.9238	-23.2231	-25.0056	-24.2014	-23.0955	-22.8880	-24.8860	-24.0254
Worst SLL (dB)	-19.8481	-22.5523	-23.4525	-21.8441	-20.2336	-22.1449	-22.6132	-22.6249
Std. Dev. SLL (dB)	1.4283	0.3488	0.8761	0.8088	1.0998	0.3990	0.7279	0.6871
Best CPU Time (s)	11.9245	17.9540	22.5680	6.2005	12.8993	19.9043	27.0965	7.6804
Mean CPU Time (s)	12.3666	18.2710	23.1640	6.3915	13.9272	21.0188	29.1125	8.2533
Worst CPU Time (s)	13.8323	20.3462	26.4553	7.7653	14.9782	22.6796	31.4356	9.3972
Std. Dev. CPU Time (s)	0.4098	0.4826	0.8481	0.3168	0.5327	0.7309	1.0820	0.3611

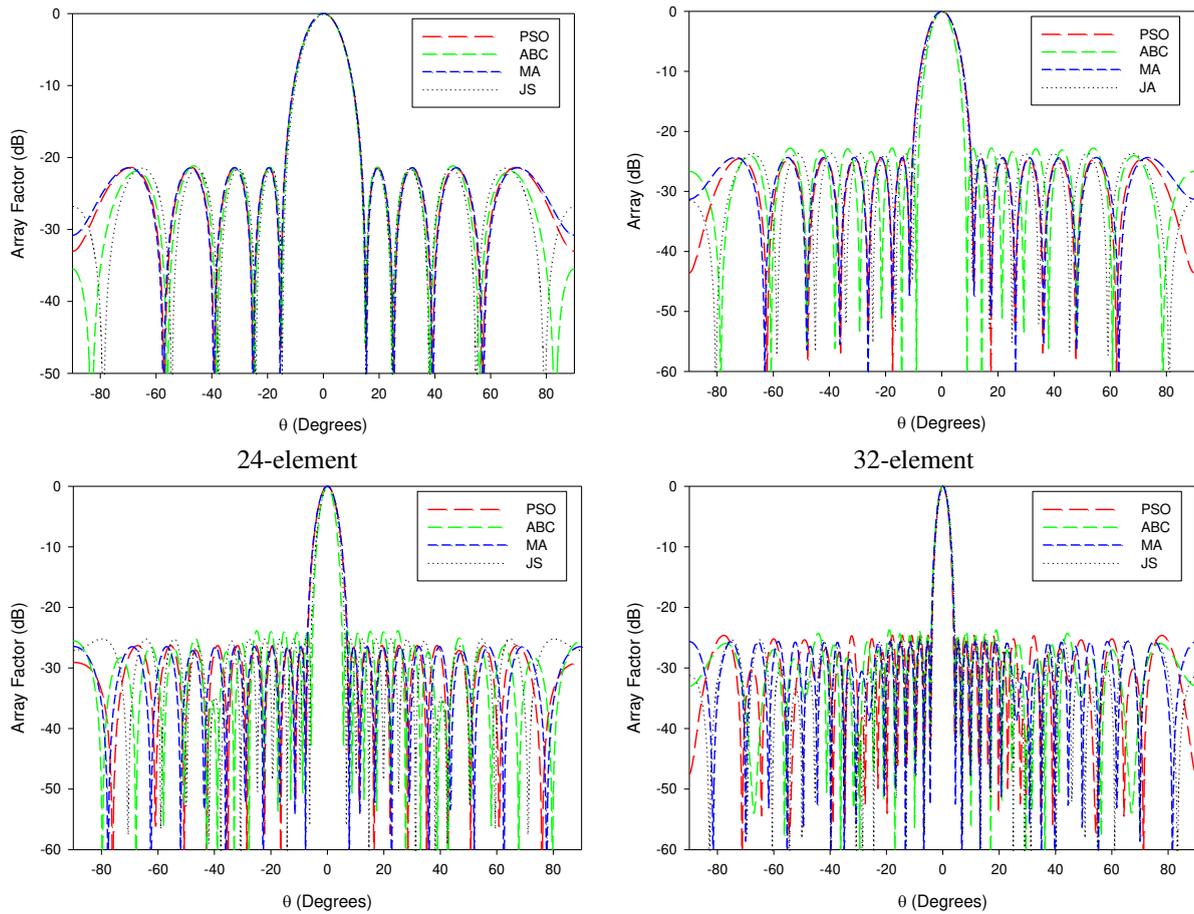


Fig. 4. Best radiation patterns for 10, 16, 24 and 32-element for position only optimal LAA design.

Table 5. The best optimized position values of LAA using PSO, ABC, MA and JS methods for 10, 16, 24 and 32-elements.

10-element	PSO	[0.3345 0.3888 0.9681 1.3514 2.0090]
	ABC	[0.3683 0.3683 0.9831 1.3721 2.0396]
	MA	[0.3600 0.3600 0.9646 1.3439 1.9999]
	JS	[0.3658 0.3783 0.9967 1.3876 2.0654]
16-element	PSO	[0.1823 0.4752 0.6727 1.1865 1.3424 1.8701 2.3206 3.0066]
	ABC	[0.1851 0.5481 0.9513 1.3352 1.8037 2.2825 2.9197 3.6633]
	MA	[0.1112 0.5421 0.6335 1.1725 1.3534 1.8597 2.3176 2.9917]
	JS	[0.2002 0.4486 0.8059 1.1563 1.5154 1.9532 2.4831 3.1557]
24-element	PSO	[0.0597 0.5555 0.6413 1.1163 1.3584 1.7007 2.1020 2.4342 2.9100 3.3799 4.0396 4.7953]
	ABC	[0.1733 0.6391 0.9411 1.4267 1.7968 2.2816 2.7032 3.2638 3.7707 4.4611 5.2854 6.0878]
	MA	[0.1309 0.4822 0.6646 1.1426 1.2523 1.7255 2.0052 2.4121 2.8350 3.3138 3.9515 4.7043]
	JS	[0.1802 0.4880 0.8975 1.2023 1.6218 1.9480 2.4049 2.8536 3.3207 3.9171 4.6549 5.4438]

32-element	PSO	[0.2119 0.6917 0.9307 1.4255 1.8137 2.2378 2.7201 3.1631 3.6071 4.1347 4.7124 5.2376 5.9530 6.7527 7.6710 8.4690]
	ABC	[0.2120 0.5839 0.9696 1.2748 1.7577 2.0859 2.5501 3.0111 3.4238 3.9094 4.4924 4.9714 5.6404 6.4386 7.2759 8.0211]
	MA	[0.2029 0.5257 0.9317 1.3505 1.7267 2.1467 2.5052 2.9446 3.4058 3.9146 4.4395 4.9265 5.5611 6.4199 7.2074 8.1295]
	JS	[0.1835 0.5751 0.9457 1.3239 1.7507 2.1134 2.5725 2.9558 3.4466 3.8855 4.4722 4.9815 5.5849 6.4261 7.2671 8.1108]

Averaged converge curves of PSO, ABC, MA and JS of 30 independent runs are illustrated in Figure 5 for 10, 16, 24 and 32-element position only optimization of LAA. As can be seen from Figure 5, for all element sizes of LAA, MA and JS converged better than PSO and ABC.

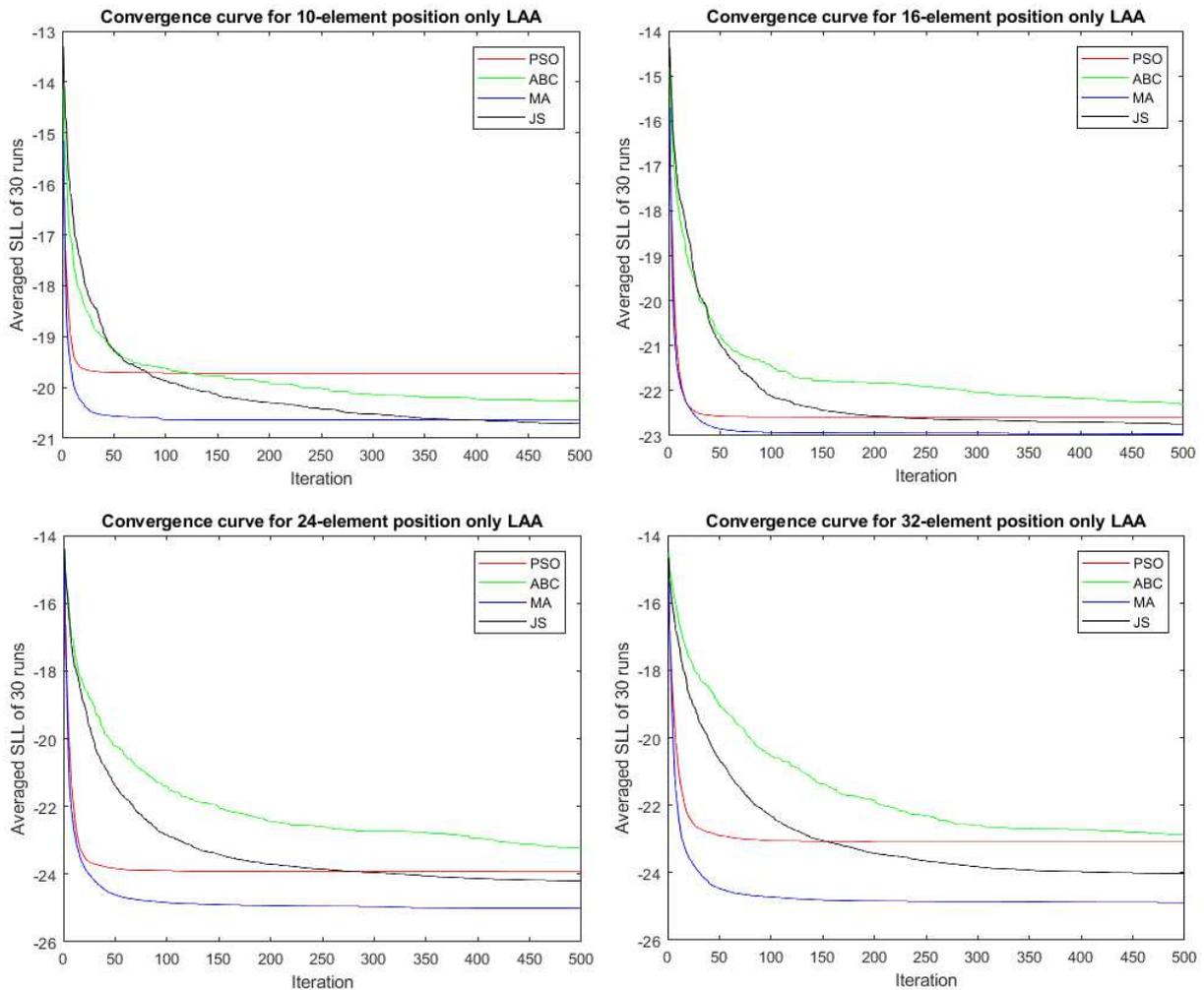


Fig. 5. Averaged converge curves of PSO, ABC, MA and JS for position only design of LAA with 10, 16, 24 and 32-elements.

5. CONCLUSIONS

In this paper, the optimum design of LAAs with a different number of elements is realized by using swarm-based optimization methods known as PSO, ABC, MA and JS. The position and amplitude values of the antenna array elements are determined for the first time by novel MA

and JS techniques to obtain radiation patterns with the lowest SLL at a desired HPBW. Experiments are conducted for 10, 16, 24 and 32-element LAA design both for amplitude only and position only cases. Simulations are repeated 30 times due to the random nature of the swarm-based meta-heuristic algorithms. Mean of the maximum SLL values and the standard deviations of independent runs are compared. Furthermore, CPU consumption times of the algorithms are also compared. Results show that MA performed the best and JS performed the second best in terms of mean SLL in general for all cases. However, MA is the slowest and JS is the fastest algorithm by means of time consumption. Overall average CPU time of MA is 20.9 seconds and JS is 5.5 seconds. In conclusion, results of novel MA and JS algorithms are better than well-known optimization techniques namely PSO and ABC. With MA and JS algorithms, antenna arrays with different geometric structures such as microstrip patch, elliptical, circular, and concentric circular can be synthesized in future studies.

References

- [1] Balanis, C. A. (2016). *Antenna theory: analysis and design*. John Wiley & Sons.
- [2] Mailloux, R. J. (2017). *Phased array antenna handbook*. Artech house.
- [3] Guney, K., Durmus, A., & Basbug, S. (2009). A plant growth simulation algorithm for pattern nulling of linear antenna arrays by amplitude control. *Progress In Electromagnetics Research*, 17, 69-84.
- [4] Dib, N. I. (2016). Design of linear antenna arrays with low side lobes level using symbiotic organisms search. *Progress In Electromagnetics Research*, 68, 55-71.
- [5] Sharaqa, A., & Dib, N. (2014). Design of linear and elliptical antenna arrays using biogeography based optimization. *Arabian Journal for Science and Engineering*, 39(4), 2929-2939.
- [6] Khodier, M. M., & Al-Aqeel, M. (2009). Linear and circular array optimization: A study using particle swarm intelligence. *Progress In Electromagnetics Research*, 15, 347-373.
- [7] Dib, N. I., Goudos, S. K., & Muhsen, H. (2010). Application of Taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays. *Progress In Electromagnetics Research*, 102, 159-180.
- [8] Das, A., Mandal, D., Ghoshal, S. P., & Kar, R. (2019). 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*, 32(1), e2486.
- [9] Subhashini, K. R. (2020). Runner-Root Algorithm to Control Sidelobe Level and Null Depths in Linear Antenna Arrays. *Arabian Journal for Science and Engineering*, 45(3), 1513-1529.

- [10] Zaman, M. A., Gaffar, M., Alam, M. M., Mamun, S. A., & Matin, M. A. (2011). Synthesis of antenna arrays using artificial bee colony optimization algorithm. *International journal of microwave and optical technology*, 6(4), 234-241.
- [11] Basu, B., & Mahanti, G. K. (2011). Fire fly and artificial bees colony algorithm for synthesis of scanned and broadside linear array antenna. *Progress In Electromagnetics Research*, 32, 169-190.
- [12] Khodier, M. (2019). Comprehensive study of linear antenna array optimisation using the cuckoo search algorithm. *IET Microwaves, Antennas & Propagation*, 13(9), 1325-1333.
- [13] Zhang, R., Zhang, Y., Sun, J., & Li, Q. (2020). Pattern Synthesis of Linear Antenna Array Using Improved Differential Evolution Algorithm with SPS Framework. *Sensors*, 20(18), 5158.
- [14] Liang, S., Fang, Z., Sun, G., Liu, Y., Qu, G., & Zhang, Y. (2020). Sidelobe Reductions of Antenna Arrays via an Improved Chicken Swarm Optimization Approach. *IEEE Access*, 8, 37664-37683.
- [15] Subhashini, K. R. (2019). Antenna array synthesis using a newly evolved optimization approach: Strawberry algorithm. *Journal of Electrical Engineering*, 70(4), 317-322.
- [16] Kennedy, J., & Eberhart, R. (1995, November). Particle swarm optimization. In *Proceedings of ICNN'95-International Conference on Neural Networks (Vol. 4, pp. 1942-1948)*. IEEE.
- [17] Karaboga, D., & Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *Journal of global optimization*, 39(3), 459-471.
- [18] Zervoudakis, K., & Tsafarakis, S. (2020). A mayfly optimization algorithm. *Computers & Industrial Engineering*, 106559.
- [19] Chou, J.-S., & Truong, D.-N. (2021). A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean. *Applied Mathematics and Computation*, 389, 125535.

Figure Captions

Fig. 1. Geometry of M-element LAA.

Fig. 2. Best radiation patterns for 10, 16, 24 and 32-element for amplitude only optimal LAA design.

Fig. 3. Averaged converge curves of PSO, ABC, MA and JS for amplitude only design of LAA with 10, 16, 24 and 32-elements.

Fig. 4. Best radiation patterns for 10, 16, 24 and 32-element for position only optimal LAA design.

Fig. 5. Averaged converge curves of PSO, ABC, MA and JS for position only design of LAA with 10, 16, 24 and 32-elements.

Table Captions

Table 1. Parameter settings of the optimization algorithms.

Table 2. Statistical SLL and CPU time comparison of PSO, ABC, MA and JS for amplitude only design of optimal LAA for 10, 16, 24 and 32-elements.

Table 3. The best optimized amplitude values of LAA using PSO, ABC, MA and JS methods for 10, 16, 24 and 32-elements.

Table 4. Statistical SLL and CPU time comparison of PSO, ABC, MA and JS for position only design of optimal LAA for 10, 16, 24 and 32-elements.

Table 5. The best optimized position values of LAA using PSO, ABC, MA and JS methods for 10, 16, 24 and 32-elements.

Figures

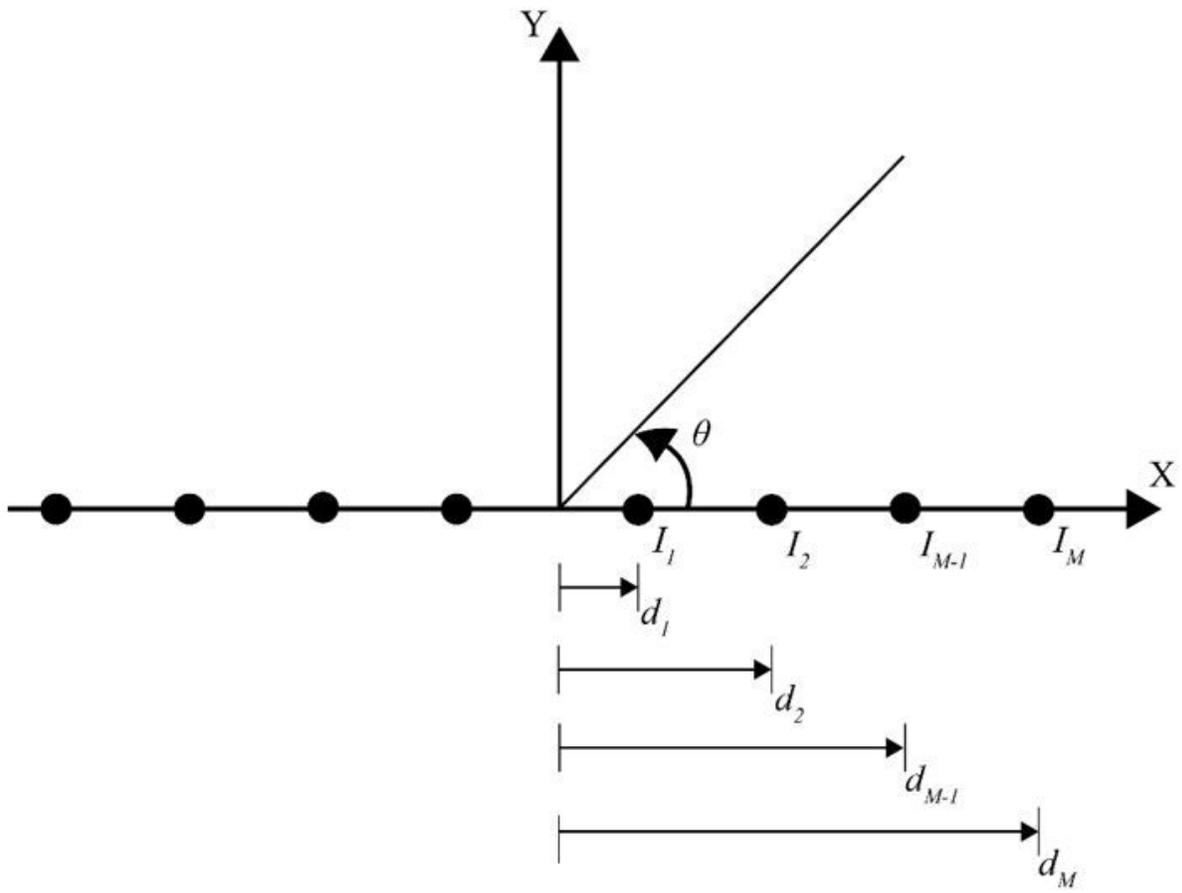


Figure 1

Geometry of M-element LAA.

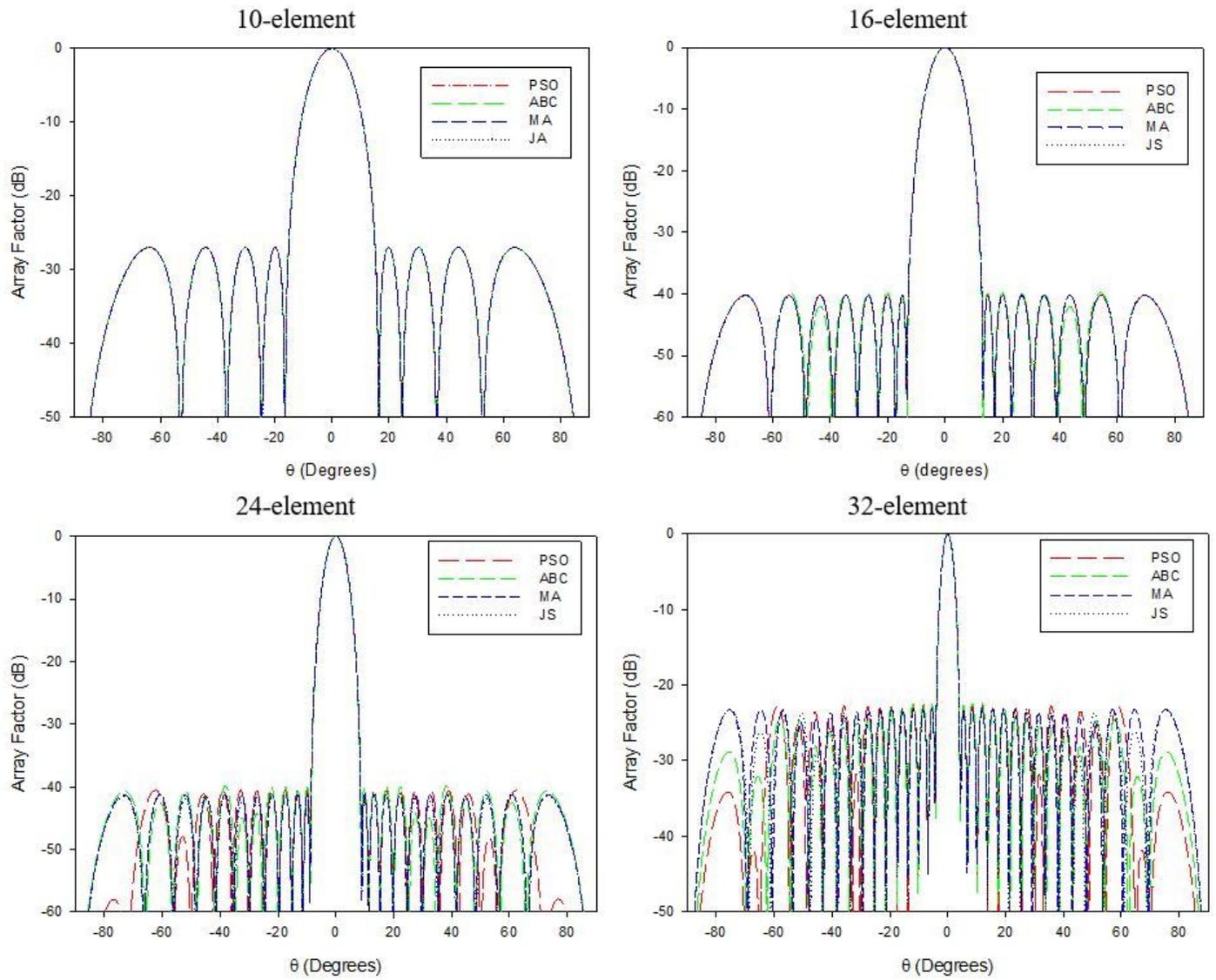


Figure 2

Best radiation patterns for 10, 16, 24 and 32-element for amplitude only optimal LAA design.

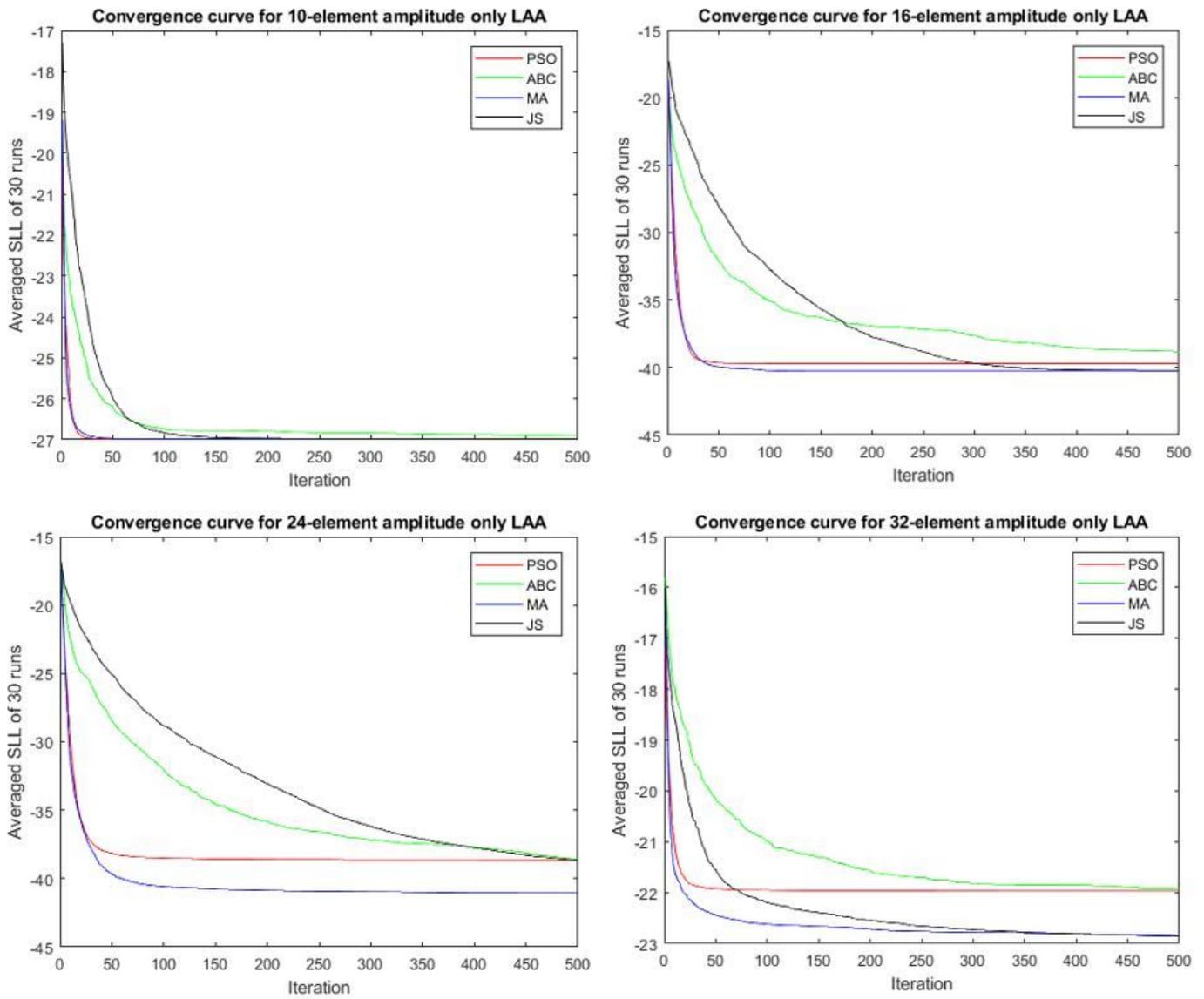


Figure 3

Averaged converge curves of PSO, ABC, MA and JS for amplitude only design of LAA with 10, 16, 24 and 32-elements.

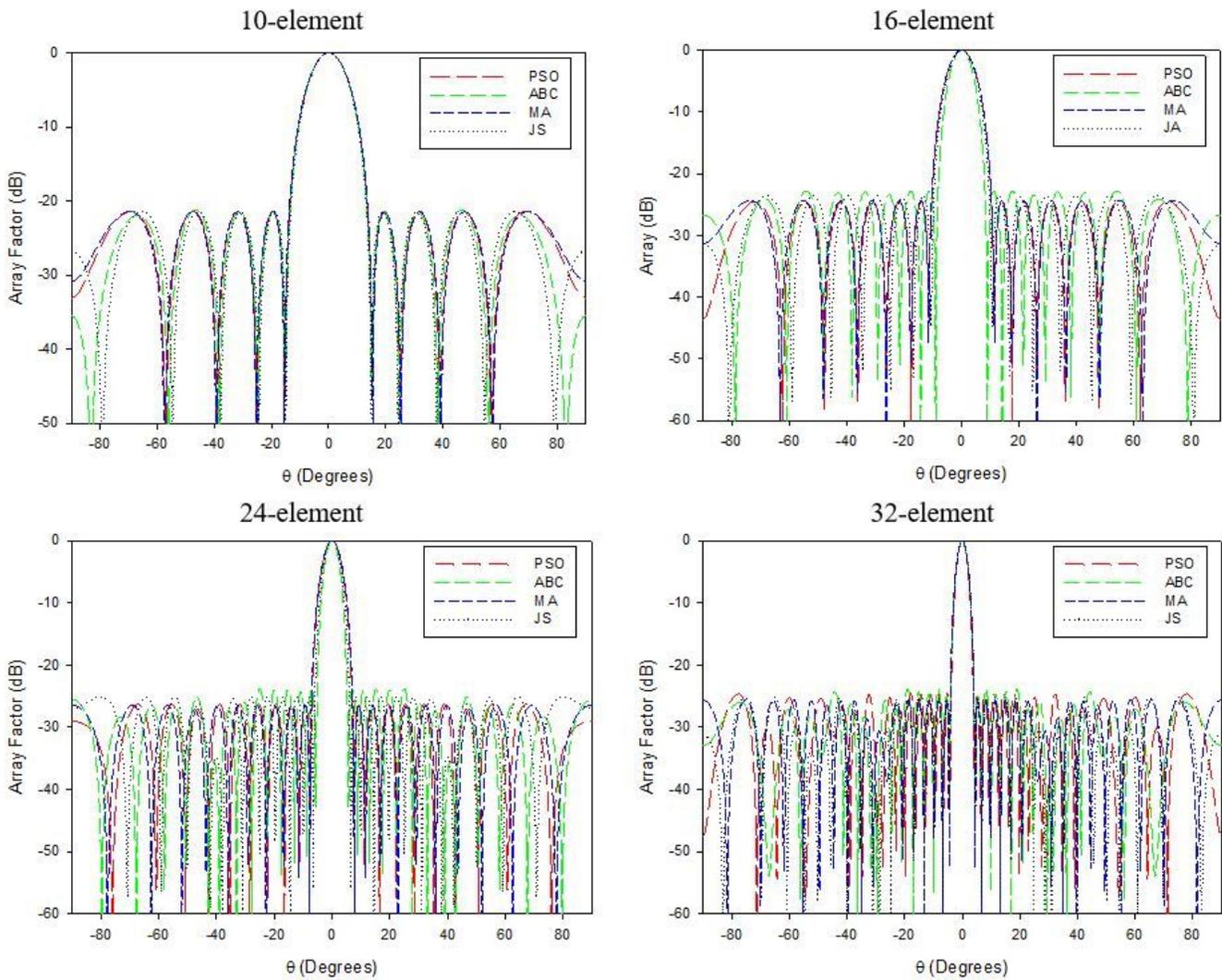


Figure 4

Best radiation patterns for 10, 16, 24 and 32-element for position only optimal LAA design.

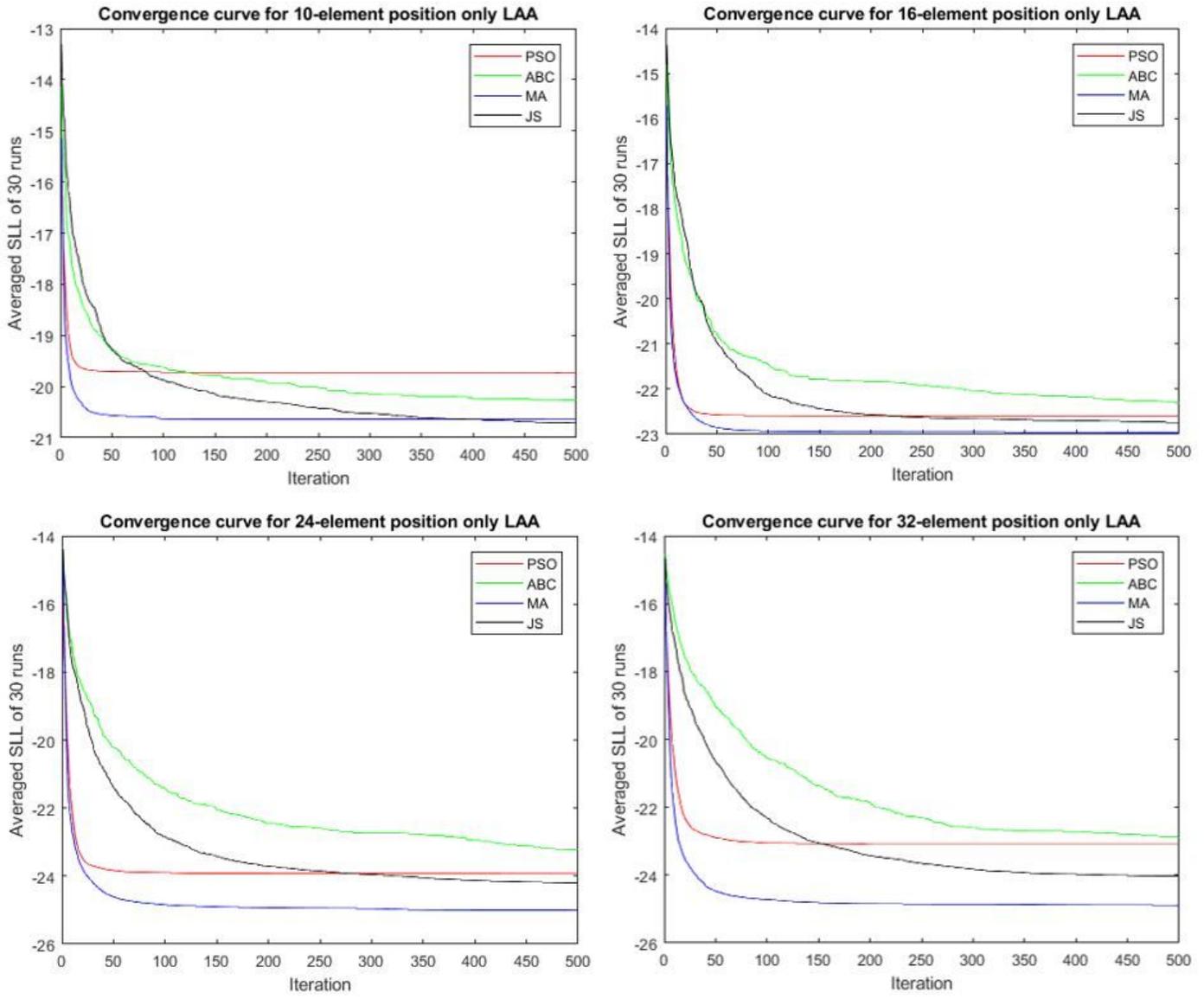


Figure 5

Averaged converge curves of PSO, ABC, MA and JS for position only design of LAA with 10, 16, 24 and 32-elements.