Arab J Sci Eng
DOI 10.1007/s13369-016-2053-2

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Optimal Synthesis of Linear Antenna Arrays Using Modified Spider Monkey Optimization

Urvinder Singh[1] · Rohit Salgotra[2]

Received: 25 May 2015 / Accepted: 20 January 2016

Abstract This paper presents a novel optimization technique named as modified spider monkey optimization (MSMO) for the synthesis of linear antenna array (LAA). The proposed method is inspired from a recently developed spider monkey optimization (SMO) swarm intelligent technique. The competitiveness of SMO has been already proved using numerical optimization functions. To improve the performance of SMO, a MSMO algorithm based on dual-search strategy is proposed in this paper. This approach generates a new solution using a search equation selected randomly from a candidate pool consisting of two search strategies. The performance of the proposed method is tested by applying it to find the optimal solutions for standard benchmark functions. Further, the capability and effectiveness is also proved by using it for practical optimization problem, i.e., synthesis of LAA for three different cases. Experimental results show that MSMO outperforms other popular algorithms like particle swarm optimization, cuckoo search, firefly algorithm, biogeography based optimization, differential evolution, tabu search and Taguchi method in terms of reduced side lobe level and faster convergence speed.

Keywords Swarm intelligence · Spider monkey optimization · MSMO · Side lobe level · Antennas · Linear antenna arrays

✉ Urvinder Singh
urvinders@gmail.com

Rohit Salgotra
cassonova.00@gmail.com

1 Department of ECE, Thapar University, Patiala, India

2 Department of ECE, Chandigarh University, Mohali, India

1 Introduction

Swarm intelligence (SI) is a nature-inspired meta-heuristic approach, and various optimization algorithms have been proposed in this field. The major advancements are ant colony optimization (ACO) [1,2], particle swarm intelligence (PSO) [3], bacterial foraging optimization (BFO) [4], artificial bee colony (ABC) [5], and others. Spider monkey optimization (SMO) algorithm is the newest addition to the field of SI. SMO algorithm was given by Bansal et al. [6] for numerical optimization problems and found it to be a novel optimization algorithm among the already present SI algorithms. It has been designed to overcome the inherent drawbacks like premature convergence and stagnations of various population-based stochastic algorithms such as DE and PSO [7–10]. It is also found to eradicate the poor performance of ABC algorithm by efficiently exploring the search space [11,12]. As SMO can be treated as the best of all already known algorithms, so proposing an enhanced version of the same will provide better results. In this article, a modified spider monkey optimization (MSMO) algorithm based on dual-search strategy is proposed. This approach uses two position updating equations and randomly selects one at a time in order to achieve faster convergence and better accuracy. The proposed algorithm is applied to benchmark problems and compared with five general algorithms, ABC [5], biogeography-based optimization (BBO) [13], covariance matrix adaptation-evolutionary search (CMA-ES) [14], adaptive DE algorithms JADE [15] and SADE [16] to prove its superiority. It is also applied for classical antenna design problems.

Antenna is the core of wireless communication, and poorly designed antennas will result in electromagnetic pollution. A single antenna does not have the enough gain for long distance communication; hence, antenna arrays having

Published online: 13 February 2016

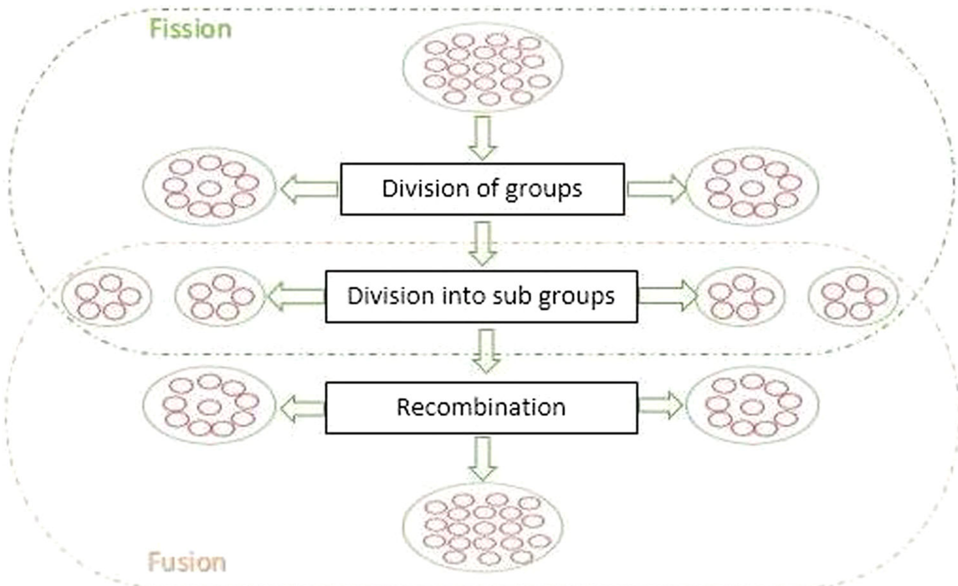


Fig. 1 Fission–fusion-based social structure (FFSS) in spider monkeys

number of antenna elements arranged in a suitable manner are employed. Among antenna arrays, linear antenna array (LAA) is very popular due to its simple geometry and wide range of applications. Synthesis of LAA is a classical problem in electromagnetism and has found interest of various researchers in the past. Several popular global optimization algorithms like genetic algorithm (GA) [17], differential algorithm (DE) [18,19], cuckoo search (CS) [20], firefly algorithm (FA) [21], tabu search (TS) [22], PSO [23–28], (BBO) [29–31] and cuckoo optimization algorithm (COA) [32] have been used for the optimization of LAA. But these methods suffer from certain shortcomings like the premature convergence to local minima while searching for global solution. In this work, LAA is designed using the proposed MSMO algorithm. The novel method is used to optimize three different antenna arrays that have already been designed using popular algorithms. This is done to make a comparison between the performance of MSMO and other optimization techniques reported in the literature for antenna design.

The rest of the paper is organized as follows: Sect. 2 presents basic SI and SMO algorithm, while Sect. 3 gives the details of the proposed MSMO algorithm. In Sect. 4, results of benchmark functions and LAA are presented. Finally, in Sect. 5, the paper is concluded.

2 SMO Algorithm and SI

SMO algorithm is based on the foraging behavior of spider monkeys. Spider monkeys are fission–fusion-based social structured animals [33], living in a group of 40–50 individuals. In fission–fusion-based social structure, a large group is firstly divided into (fission) smaller groups until a certain criteria is fulfilled and then combined (fusion) to form a single group. These groups are formed during day time with sun rise and recombine for sleeping at night [34]. These animals follow such structure in search of food and depending upon its availability divide or combine the group, hence reducing the direct competition among group members when there is scarcity of food. The group is headed by a global leader (preferably female); when this leader is not able to find sufficient food for the group, she divides the group into subgroups headed by local leaders. The subgroups are temporary and vary in formation throughout the day with a minimum of three members in a subgroup. The advantage of division of groups and being a member of larger group increase mating chances and provides security from predators. When the group is too small in size, animals are at risk of being preyed upon and are at disadvantage of suffering from lower feeding rates as more time is spent in scanning for predators. When group is very large, though the competition for food increases, but the potential risk of being preyed is reduced [34]. Figure 1 shows the various stages of fission–fusion-based social structure (FFSS) in a group of 20 spider monkeys.

With the sunrise, all the members of the larger group start moving out in smaller groups 10–12 members each resulting in fission. Depending upon the availability of food in a particular region, these smaller groups further divide to form subgroups of 5–6 members. The group members communicate with each other by using postures and positions of sexual receptivity and attack. Each individual communicate with other over long distances using visual or vocal communication, and this permits spider monkeys to get together, forage food and stay away from enemies. The members of different subgroups are not always close to each other but are mutually tolerant toward each other and hence show as if they belong to a larger group [6]. As the sun gets down, all these spider start gathering toward their local leader and further to the global leader resulting in fusion of smaller subgroups into a larger group. The SMO algorithm is divided into seven major parts as explained in [6].

In [5], the authors found that any algorithm which follows *self-organization and division of labor* is said to be swarm intelligent. Self-organization is important feature in swarms and spider monkeys depict this property by means of interactions among its subgroups without any central authority. All the group members build up a single, organized and distributed structure. It has four important characteristics given by Bonabeau et al. [35]

(i) *Positive Feedback* It is the information from output reapplied to the input. In SMO local learning, global learning and self-experience utilized in different steps is considered as positive feedback.
(ii) *Negative Feedback* It is meant for compensation of positive feedback. Here, local leader and global leader limit provides negative feedback to help local and global leader in decision making.
(iii) *Fluctuations* It is meant for randomness of the swarm. Here, randomness means finding food in different directions. Fluctuations help the system to get rid of the stagnation problem.
(iv) *Multiple Interactions* It helps the swarm to interact with each other and learning from each other for providing better intelligence. Here, global and local leader phases are meant to communicate with each other and give optimum results based on the fitness of the system.

Division of labor deals with performing various tasks simultaneously by cooperating with each other. Simultaneous tasks provide better results as compared to sequential tasks. In SMO, when global leader gets trapped in the local minima, it divides the group into subgroups and hence justifying the definition of division of labor. Hence, SMO algorithm is considered as a SI technique.

3 Modified Spider Monkey Optimization

Modified spider monkey optimization algorithm is an improved version of basic SMO algorithm in terms of convergence speed and solution quality. The algorithm is motivated from M-ABC algorithm proposed in [36]. In their paper, the authors firstly modified ABC algorithm using random key-based encoding for solving combinatorial optimization problems and then used a multi-search strategy employed in a random manner to generate a new neighbor solution randomly from four selected search strategies and found it effective in terms of solution quality and CPU time. In this work, a dual-search strategy inspired from multi-search strategy is used to update the position updating equations of the basic SMO algorithm. This approach generates a new solution using a search equation selected randomly from a candidate pool consisting of two search strategies. Dual-search strategy is followed in local leader phase, global leader phase and local leader decision phase. The steps of MSMO algorithm are as follows:

Initialization In MSMO algorithm, there is an initial population of N spider monkeys (SM). Here, each SM is initialized and acts as the potential solution of problem under consideration.

$$\mathrm{SM}_{i,j} = \mathrm{SM}_{\min,j} + b * \left(\mathrm{SM}_{\min,j} - \mathrm{SM}_{\max,j}\right) \tag{1}$$

where $i \in \{1,\ldots,\mathrm{SM}\}$, $j \in \{1,\ldots,D\}$, $\mathrm{SM}_{i,j}$ is the ith spider monkey in the jth dimension; b is randomly generated number between [0, 1]; $\mathrm{SM}_{\min,j}$, $\mathrm{SM}_{\max,j}$ are lower and upper bounds, respectively. D is the problem dimension or number of variables in the optimization problem. The fitness (fit_i) for objective function (f_x) is calculated using the equation given below:

$$\mathrm{fit}_i = \begin{cases} 1 + f_x, & f_i \geq 0, \\ \frac{1}{1+|f_x|}, & f_i \leq 0 \end{cases} \tag{2}$$

Local Leader Phase This is the second phase in MSMO. Here, new solutions or position is updated using the experience of local leader and local group members. Unless a new best fit solution is produced, the position of previous best solution is not updated. The position of spider monkeys is updated by randomly selecting one (among the two) search equation:

(i) Original SMO equation:

$$\mathrm{SMnew}_{i,j} = \begin{cases} \mathrm{SM}_{i,j} + \left(\left(b * \left(\mathrm{LL}_{k,j} - \mathrm{SM}_{i,j}\right)\right) + \left(d * \left(\mathrm{SM}_{r,j} - \mathrm{SM}_{i,j}\right)\right)\right), & b \geq \mathrm{pr} \\ \mathrm{SM}_{i,j}, & \text{otherwise} \end{cases} \tag{3}$$

(ii) ABC/best/1 inspired by DE and proposed by Gao and Liu [12]:

$$\mathrm{SMnew}_{i,j} = \mathrm{LL}_{k,j} + b * \left(\mathrm{SM}_{z,j} - \mathrm{SM}_{a,j}\right) \tag{4}$$

Here z and a are random indexes chosen from the spider monkey population. $\mathrm{SMnew}_{i,j}$ is the new solution or updated position of spider monkeys, $\mathrm{SM}_{i,j}$ is the previous solution, $\mathrm{SM}_{r,j}$ is the jth dimension of rth spider monkey, $\mathrm{LL}_{k,j}$ is the local leader belonging to the jth dimension of kth group, d

is a random number between $[-1, 1]$ and pr is perturbation rate.

Global Leader Phase In this phase, position of spider monkeys is updated by using global leader (GL) and group member experience. All the positions are updated by calculating probability for each member. The probability is calculated as:

$$P_i = 0.9 * \frac{(\text{fit}_i)}{\text{max\_fit}} + 0.1 \tag{5}$$

The position update equations for this phase are given as:

(i) Original SMO equation:

$$\begin{aligned}\text{SMnew}_{i,j} = \text{SM}_{i,j} + \left(b * \left(\text{GL}_j - \text{SM}_{i,j}\right)\right) \\ + d * \left(\text{SM}_{r,j} - \text{SM}_{i,j}\right)\end{aligned} \tag{6}$$

(ii) ABC/best/1 inspired by DE [12]:

$$\text{SMnew}_{i,j} = \text{GL}_j + b * \left(\text{SM}_{z,j} - \text{SM}_{a,j}\right) \tag{7}$$

where b is random number in range of [0, 1], d is another random number in the range of $[-1, 1]$ and GL_j is the updated global leader.

Local Leader Learning Phase In this phase, the position of local leader is updated based upon greedy search in a particular group. The SM having the best fitness in a particular group is considered as local leader and if the local leader is not getting updated, the local leader count is incremented by 1.

Global Leader Learning Phase Here, global leader's position is updated by applying greedy search. The spider monkey having the best fitness is considered as the global leader and if the global leader is not updating, the global leader count is incremented by 1.

Local Leader Decision Phase If any LL is not updating its position up to some specified number of times, all the group member positions are updated and local leader count is made zero. The update equations are given below:

(i) Original SMO equation:

$$\text{SM}_{i,j} = \begin{cases} \text{SM}_{i,j} + \left(b * \left(\text{SM}_{i,j} - \text{LL}_{k,j}\right)\right) \\ \quad + b * \left(\text{GL}_j - \text{SM}_{i,j}\right), & b < \text{pr} \\ \text{SM}_{\min,j} + b * \left(\text{SM}_{\min,j} - \text{SM}_{\max,j}\right), & \text{otherwise} \end{cases} \tag{8}$$

(ii) ABC/best/1 inspired by DE:

$$\text{SMnew}_{i,j} = \text{LL}_{k,j} + b * \left(\text{GL}_j - \text{SM}_{a,j}\right) \tag{9}$$

Global Leader Decision Phase If any GL is not updating after a specified number of times, divide the group into subgroups. When maximum number of groups have been formed, recombine all the groups to form a single group and update the local leader position. Here, the global leader and local leader are made the same.

There are six major phases in this algorithm, the first phase is local leader phase where exploration (also called deviation process) takes place, second phase is global leader phase that is meant for exploitation (also called selection process) and together these form the main part of the algorithm. The local leader phase is made to run every time but global leader phase runs only when the value of perturbation rate exceeds some particular random value. In local learning and global learning phase, there are two control parameters, one is global leader limit and another is local leader limit. Local leader limit and global leader limit is used to avoid stagnation for local leader and global leader, respectively. In the local learning phase, local leader is updated and the local leader count is incremented if and only if the value of local leader is not updating. For global learning phase, global leader is updated and when it is not updating global leader count is incremented. Next phase is the local leader decision phase in which re-initialization takes place and the local leader counter is made zero. This phase also helps in better exploration of search space. The last phase is the global leader decision phase in which the group is divided into maximum number of parts and this happens only if the global leader is not updating after a specified number of times. Finally the best fit individual gives the solution of problem under consideration. The flow code for MSMO algorithm is given in Fig. 2.

ABC/best/1 [12] equation inspired by DE is added due to the high robustness and faster convergence of DE algorithm. Hence, hybridizing DE with SMO is expected to provide a lot better results than the basic SMO algorithm. The inclusion of ABC/best/1 equation from DE adds up a modification in the basic SMO algorithm. In comparison with other algorithms, the main focus here is that the system may not get stuck at any of the local minima or converge prematurely and provide faster convergence speed.

4 Results and Discussion

4.1 Benchmark Results

In order to test the performance of MSMO algorithm and compare it with other popular optimization algorithms, it is applied to standard benchmark functions and LAA synthesis. Firstly, the MSMO is used to find optimum solutions for classical benchmark functions and its performance is compared with other popular algorithms. Secondly, MSMO is employed to find optimum solutions for LAA and again its

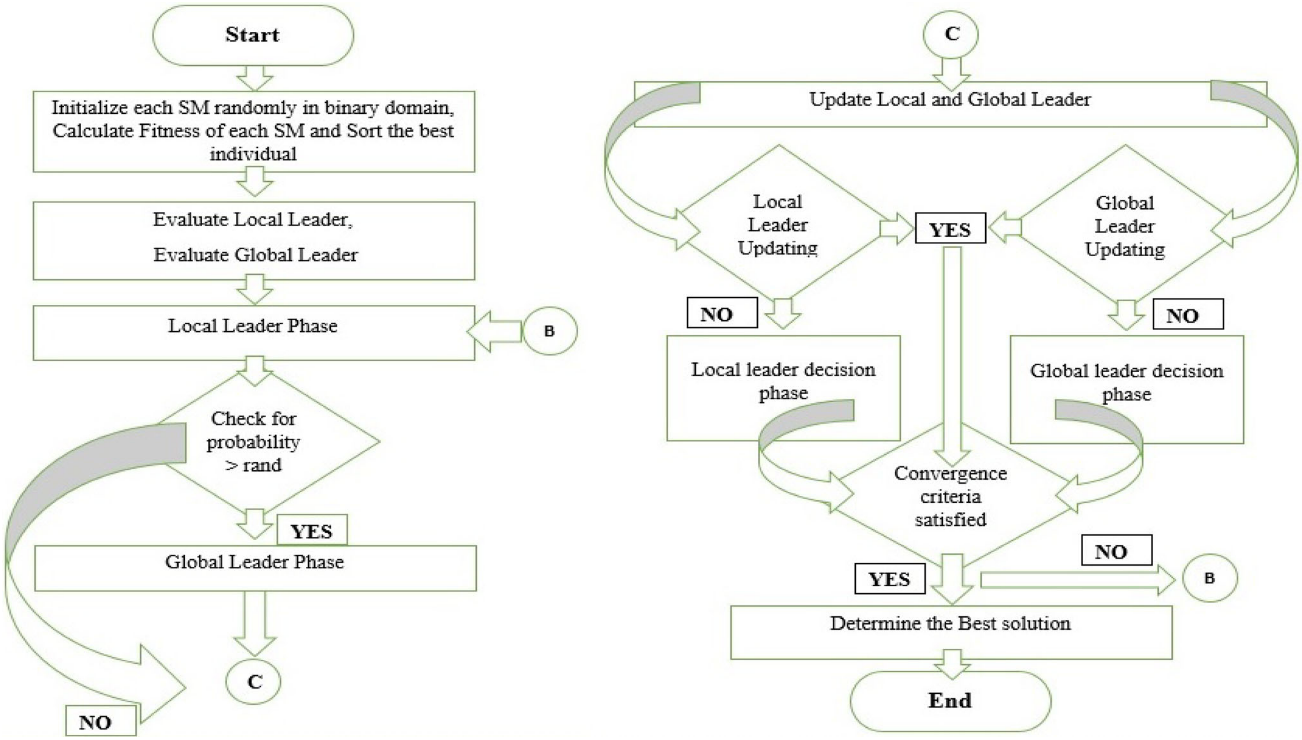


Fig. 2 Flow code for MSMO algorithm

performance is compared with well-known algorithms that have been used in the past for the synthesis of LAA. The simulations are performed on Windows 7, Intel core i3 processor with MATLAB version 7.10.0 (R2010a).

MSMO is used to optimize unimodal and multimodal benchmark functions [37,38]. Table 1 shows test functions of variable dimension with their search ranges and optimal goals. For the purpose of comparison, SMO, ABC, BBO, CMA-ES, JADE and SADE are also applied to the same benchmark functions. The maximum number of iterations are set to 150, while the algorithm is run 20 times and the best results are reported. Convergence curves are also drawn for test functions to show the better performance of MSMO algorithm over others.

The algorithms are stochastic in nature, so statistical tests should be conducted. The Wilcoxon rank-sum test [39] has been conducted in order to prove their significance statistically. This test is a nonparametric test used to detect static significance of any algorithm. Here, two pairs of populations are compared and their differences are analyzed. The test returns a p value determining the significance level of two algorithms. For an algorithm to be statistically significant, the result of p test should be <0.05.

The parameters for MSMO are taken as:

- Number of spider monkeys, SM = 20 (for test functions).
- Maximum number of groups = 5.
- Perturbation rate = [0.1, 0.9], linearly increasing as:

$$\mathrm{pr}_{G+1} = \mathrm{pr}_G + (0.4 - 0.1)/\mathrm{maxiter}$$

where G is the iteration counter, pr is the perturbation rate and maxiter is the maximum number of iterations.

- Global leader limit = SM/2.
- Local leader limit = $[D/6] \times \mathrm{SM}$; where D is the problem dimension.
- Maximum iterations = 150.

The parameter settings for other algorithms used for benchmark functions are shown in Table 2. In the present work, emphasis has been laid to check the performance using minimum number of function evaluations. The experimental results show that MSMO algorithm performs better than SMO algorithm for most of the test functions. For functions f_3, f_6, f_8, f_{15}, f_{16}, f_{17}, f_{18} and f_{19} the SD of MSMO algorithm is much better than all other algorithms (except for SADE over f_1, f_2, SMO over f_4, CMA-ES over f_5, f_{13}, BBO over f_7 and ABC over f_{14}), showing that MSMO is consistent in finding optimum solution. It can simply be seen from Table 3 that MSMO outperform other algorithms for most of the test function in achieving the global minimum while giving comparable results for rest of test functions. For f_5, ABC has got the best SD but is not able to achieve global optimum value and gets struck at local minima. BBO and JADE also get stuck in local minima for f_5. In case of f_9, f_{10}, f_{11}, and f_{12}, JADE, SADE and CMA-ES get stuck in

Table 1 Description of Test Functions

Test problems	Objective function	Search range	Optimum value	D
Schwefel function	$f_1(x) = \sum_{i=1}^{D} \left[x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right]$	$[-500, 500]$	$-418.9829\times$ D	30
Schwefel function	$f_2(x) = \sum_{i=1}^{D} \left[x_i \sin\left(\sqrt{\lvert x_i \rvert}\right)\right]$	$[-500, 500]$	$-418.9829\times$ D	100
Rastrigin function	$f_3(x) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	0	30
Rastrigin function	$f_4(x) = 10D + \sum_{i=1}^{D} \left[x_i^2 - 10\cos(2\pi x_i)\right]$	$[-5.12, 5.12]$	0	100
Branin RCOS function	$f_5(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$x_1 \in [-5, 10], x_2 \in [0, 15]$	0.397887	2
Six hump camel function	$f_6(x) = \left(4 - 2.1x_1^2 + \frac{x_1^4}{3}\right)x_1^2 + x_1x_2 + \left(-4 + 4x_2^2\right)x_2^2$	$[-5, 5]$	-1.0316	2
Goldstein and price function	$f_7(x) = (1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14$ $x_2 + 6x_1x_2 + 3x_2^2))(30 + (2x_1 - 3x_2)^2 (18 - 32x_1$ $+12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2))$	$[-2, 2]$	3	2
Hartmann function 3	$f_8(x) = -\sum_{i=1}^{4}\alpha_i \exp\left[-\sum_{j=1}^{3} A_{ij}\left(x_j - P_{ij}\right)^2\right]$	$[0, 1]$	-3.86278	3
Hartmann function 6	$f_9(x) = -\sum_{i=1}^{4}\alpha_i \exp\left[-\sum_{j=1}^{6} A_{ij}\left(x_j - P_{ij}\right)^2\right]$	$[0, 1]$	-3.32237	6
Shekel function 5	$f_{10}(x) = -\sum_{j=1}^{5}\left[\sum_{i=1}^{4}\left(\left(x_i - C_{ij}\right)^2 + \beta_j\right)^{-1}\right]$	$[0, 10]$	-10.1532	4
Shekel function 7	$f_{11}(x) = -\sum_{j=1}^{7}\left[\sum_{i=1}^{4}\left(\left(x_i - C_{ij}\right)^2 + \beta_j\right)^{-1}\right]$	$[0, 10]$	-10.4029	4
Shekel function 10	$f_{12}(x) = -\sum_{j=1}^{10}\left[\sum_{i=1}^{4}\left(\left(x_i - C_{ij}\right)^2 + \beta_j\right)^{-1}\right]$	$[0, 10]$	-10.5364	4
Sphere function	$f_{13}(x) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]$	0	10
Beale function	$f_{14}(x) = [1.5 - x_1(1 - x_2)]^2 + \left[2.25 - x_1\left(1 - x_2^2\right)\right]^2$ $+\left[2.625 - x_1\left(1 - x_2^3\right)\right]^2$	$[-4.5, 4.5]$	0	2
Easom function	$f_{15}(x) = -\cos x_1 \cos x_2 e^{\left(-(x_1-\pi)^2-(x_2-\pi)^2\right)}$	$[-10, 10]$	-1	2
Rotated high conditional elliptic function	$f_{16}(x) = \sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]$	0	30
Rotated high conditional elliptic function	$f_{17}(x) = \sum_{i=1}^{D}\left(10^6\right)^{\frac{i-1}{D-1}} x_i^2$	$[-100, 100]$	0	50
Shifted and rotated Weierstrass function	$f_{18}(x) = \sum_{i=1}^{D}\sum_{k=0}^{kmax}[a^k \cos\left(2\pi b^k (x_i + 0.5)\right)]$ $-D\sum_{k=0}^{\mathrm{kmax}}\left[a^k \cos\left(2\pi b^k .0.5\right)\right]$; where $a = 0.5, b = 3$, kmax $= 20$	$[-0.5, 0.5]$	0	30
Shifted and rotated Weierstrass function	$f_{19}(x) = \sum_{i=1}^{D}\sum_{k=0}^{\mathrm{kmax}}[a^k \cos\left(2\pi b^k (x_i + 0.5)\right)]$ $-D\sum_{k=0}^{\mathrm{kmax}}[a^k \cos\left(2\pi b^k .0.5\right)]$; where $a = 0.5, b = 3$, kmax $= 20$	$[-0.5, 0.5]$	0	50

Table 2 Parameter settings for MSMO, SMO, ABC, BBO, CMA-ES, JADE and SADE algorithms

Algorithm	Parameters	Values
SMO	Number of spider monkeys	20
	Maximum number of groups	5
	Global leader limit	[D/2]
	Local leader limit	[D/6] × number of spider monkeys
	Perturbation rate	[0.1, 0.9]
	Maximum number of iterations	150
ABC	Colony size	20
	Number of food sources	Colony size/2
	Limit	100
	Maximum cycles	150
	Stopping criteria	Max iteration
BBO	Population size	20
	Habitat modification probability	1
	Mutation probability	0
	Maximum iterations	150
	Stopping criteria	Max iteration
CMA-ES	Maximum iterations	150
	Stopping criteria	Max iteration
JADE	Population size	20
	Maximum iterations	150
	Stopping criteria	Max iteration
SADE	Population size	20
	Maximum iterations	150
	Stopping criteria	Max iteration

local minima but MSMO achieves global optimum for almost all the cases. So even for f_9, f_{10}, f_{11}, and f_{12}, MSMO can be considered as the best. The p values in Table 4 also show that MSMO algorithm is statistically superior since p values are much less than 0.05. Also from the convergence curves in Fig. 3, it can be simply seen that MSMO algorithm performs better than other algorithms for most of the test functions in terms of convergence speed while comparable results have been obtained for others. This happens due to the benefits of higher exploitation assisting MSMO to converge rapidly toward the optimum solution. Thus from the above comparison, it has been found that MSMO algorithm performs better than ABC, BBO, SMO, JADE, SADE and CMA-ES algorithms for benchmark functions.

4.2 Linear Antenna Array Results

The strength and competence of MSMO is evaluated by applying it to synthesis of linear arrays. In this paper, three different classical linear antenna problems are taken from the literature which have been investigated in the past using number of optimization techniques [18,20,22,24–29,31,32].

A LAA consists of antenna elements placed along a straight line. A symmetric LAA along the x-axis with 2N elements is shown in Fig. 4. The array factor (AF) in x–y plane (azimuth plane) is given by [40]:

$$\mathrm{AF}\,(\emptyset) = \sum_{\substack{n=-N \\ n\neq 0}}^{N} I_n \exp\left(j\left[kx_n \cos(\emptyset) + \varphi_n\right]\right) \tag{10}$$

where I_n, φ_n, x_n are amplitude excitation, phase and position of nth element, respectively, $\emptyset$ is the polar angle of the far-field point from the end fire direction and $k = 2\pi/\lambda$ is the wave number.

For a linear array placed symmetrically along the x-axis, taking $I_{-n} = I_n$ and $\varphi_{-n} = -\varphi_n$, the AF can be simplified as:

$$\mathrm{AF}\,(\emptyset) = 2\sum_{n=1}^{N} I_n \cos\left[kx_n \cos(\emptyset) + \varphi_n\right] \tag{11}$$

The MSMO is investigated for two different types of linear arrays: an equally spaced linear array and an unequally spaced linear array.

Table 3 Results of MSMO algorithm compared with other algorithms

Objective function	Algorithm	Best	Worst	Mean	SD
$f_1(x)$	SMO	−9.13E+03	−7.02E+03	−7.82E+03	5.06E+02
	ABC	−8.75E+03	−7.08E+03	−7.98E+03	4.71E+02
	BBO	−1.85E+03	−1.10E+03	−1.48E+03	2.31E+02
	CMA-ES	−1.21E+03	−1.12E+03	−1.17E+03	2.21E+01
	JADE	−7.95E+03	−7.80E+03	−7.88E+03	4.16E+01
	SADE	−1.53E+03	−1.51E+03	−1.52E+03	**7.84−00**
	MSMO	**−1.11E+04**	**−7.90E+03**	**−9.09E+03**	7.84E+02
$f_2(x)$	SMO	−1.51E+04	−1.36E+04	−1.51E+04	8.47E+02
	ABC	−1.79E+04	−1.39E+04	−1.63E+04	1.14E+03
	BBO	−1.32E+04	−1.57E+04	−1.43E+04	7.53E+02
	CMA-ES	−5.04E+03	−4.44E+03	−4.76E+03	1.56E+02
	JADE	−1.47E+04	−1.43E+04	−1.45E+04	9.46E+01
	SADE	−1.25E+04	−1.22E+04	−1.24E+04	**7.15E+01**
	MSMO	**−2.52E+04**	**−2.24E+04**	**−2.52E+04**	1.20E+03
$f_3(x)$	SMO	0.000000	0.995000	0.0497000	2.22E−01
	ABC	0.222700	6.894900	2.3416000	1.765200
	BBO	4.20E+01	7.24E+01	5.533E+01	6.931700
	CMA-ES	2.31E+02	2.42E+02	2.37E+02	3.40E−00
	JADE	1.34E+02	1.38E+02	1.38E+02	1.29E−00
	SADE	1.35E+02	1.40E+02	1.38E+02	1.42E−00
	MSMO	**0.0000000**	**1.50E−02**	**7.495E−04**	**3.40E−03**
$f_4(x)$	SMO	**0.000000**	**0.000000**	**0.000000**	**0.000000**
	ABC	1.13E−00	9.06E−00	5.07E−00	2.15E−00
	BBO	4.59E+02	6.32E+02	5.33E+02	4.20E+01
	CMA-ES	8.21E+02	8.49E+02	8.34E+02	6.72E−00
	JADE	7.95E+02	8.09E+02	8.03E+02	3.73E−00
	SADE	7.46E+02	7.60E+02	7.55E+02	3.60E−00
	MSMO	0.000000	3.97E−00	9.45E−01	1.18E−00
$f_5(x)$	SMO	0.3979000	0.3979000	0.3979000	5.58E−07
	ABC	0.0000000	0.0000000	0.0000000	0.000000
	BBO	0.5513000	0.6445000	0.4532000	8.23E−02
	CMA-ES	0.3979000	0.3979000	0.3979000	1.70E−16
	JADE	0.5320000	0.5768000	0.5553000	1.35E−02
	SADE	0.3987000	0.3991000	0.3989000	8.96E−01
	MSMO	**0.3979000**	**0.3979000**	**0.3979000**	**1.83E−10**
$f_6(x)$	SMO	−1.031600	−1.031600	−1.031600	1.40E−07
	ABC	−1.031600	−1.024900	−1.030400	1.70E−03
	BBO	−1.023400	−0.047900	−0.731400	3.45E−01
	CMA-ES	−1.027500	−1.003000	−1.016200	6.40E−03
	JADE	−1.016600	−1.016000	−1.014100	1.50E−03
	SADE	−1.030700	−1.030400	−1.030600	8.66E−05
	MSMO	**−1.031600**	**−1.031600**	**−1.031600**	**8.96E−11**
$f_7(x)$	SMO	3.0000000	3.0000000	3.0000000	5.11E−08
	ABC	3.0002000	3.2993000	3.0369000	7.33E−02
	BBO	**3.0000000**	**3.0000000**	**3.0000000**	**0.000000**
	CMA-ES	3.0000000	84.000000	15.150000	2.96E+01
	JADE	9.877E+05	9.410E+05	9.598E+05	1.28E+05

Table 3 continued

Objective function	Algorithm	Best	Worst	Mean	SD
	SADE	3.0187000	3.0283000	3.0224000	2.50E−03
	MSMO	**3.0000000**	**3.0000000**	**3.0000000**	3.89E−11
$f_8(x)$	SMO	−3.862800	−3.862800	−3.862800	1.51E−07
	ABC	−3.861600	−3.856600	−3.861600	1.70E−03
	BBO	−3.266000	−0.001200	−0.918800	1.02E−00
	CMA-ES	−3.808100	−3.656000	−3.751200	3.58E−01
	JADE	−3.851900	−3.848200	−3.850300	8.85E−04
	SADE	−3.825800	−3.814700	−3.821000	3.00E−03
	MSMO	**−3.862800**	**−3.862800**	**−3.862800**	**2.50E−09**
$f_9(x)$	SMO	−3.322400	−3.203200	−3.278000	5.73E−02
	ABC	−3.321900	−3.188300	−3.247600	6.06E−02
	BBO	−3.037200	−1.733000	−2.621400	4.04E−01
	CMA-ES	−3.188200	−2.788000	−3.006700	1.10E−01
	JADE	−2.978800	−2.946000	−2.961900	8.70E−03
	SADE	−2.608300	−2.515200	−2.564300	2.43E−02
	MSMO	**−3.322400**	**−3.203200**	**−3.238900**	**5.60E−02**
$f_{10}(x)$	SMO	−10.15320	−2.682700	**−7.315800**	3.104400
	ABC	−10.12730	−2.247400	−6.023200	3.180200
	BBO	−10.15320	−2.630400	−5.163100	3.429700
	CMA-ES	−4.982900	−4.801400	−4.895200	4.00E−02
	JADE	−0.250500	−0.206600	−0.227700	**1.11E−02**
	SADE	−1.790600	−1.50610	−1.70580	7.14E−02
	MSMO	**−10.15320**	−2.63050	−5.64220	**3.198100**
$f_{11}(x)$	SMO	−10.40290	−2.75190	−6.31860	3.825600
	ABC	−10.38210	−1.82620	−5.77530	3.133200
	BBO	−10.40280	−2.75190	−5.72450	3.544400
	CMA-ES	−4.991800	−4.74720	−4.89850	5.81E−02
	JADE	−0.281800	−0.21860	−0.24380	1.42E−02
	SADE	−2.126400	−1.93150	−2.01210	6.06E−02
	MSMO	**−10.40290**	**−3.63970**	**−8.45850**	**2.802500**
$f_{12}(x)$	SMO	−10.53640	−2.87110	−8.45440	**2.778100**
	ABC	−10.50260	−1.97760	−6.19360	3.748800
	BBO	−10.53630	−2.42080	−5.37680	3.163100
	CMA-ES	−5.023000	−4.87470	−4.96250	3.72E−02
	JADE	−0.300500	−0.24670	−0.26990	1.35E−02
	SADE	−2.27090	−1.92160	−2.12670	8.89E−02
	MSMO	**−10.53640**	−2.87110	**−8.17620**	3.341500
$f_{13}(x)$	SMO	5.04E−10	8.31E−08	2.01E−08	2.12E−08
	ABC	5.694400	1.61E+02	4.27E+01	3.65E+01
	BBO	3.02E−03	3.93E−01	1.48E−01	8.31E−02
	CMA-ES	**1.95E−14**	**1.29E−10**	**9.26E−12**	**2.90E−11**
	JADE	1.19E−06	2.62E−06	1.58E−06	3.28E−07
	SADE	1.11E−03	1.40E−02	3.00E−03	2.90E−03
	MSMO	1.09E−06	1.03E−01	9.87E−03	3.24E−02
$f_{14}(x)$	SMO	7.94E−08	1.75E−05	3.37E−06	5.09E−06
	ABC	**0.000000**	**0.000000**	**0.000000**	**0.000000**
	BBO	3.71E−04	7.03E−01	3.57E−01	2.99E−01

Table 3 continued

Objective function	Algorithm	Best	Worst	Mean	SD
	CMA-ES	3.59E−13	4.56E−01	4.55E−02	1.40E−01
	JADE	4.90E−03	4.83E−02	2.11E−02	1.03E−02
	SADE	3.59E−04	5.40E−03	6.68E−04	1.10E−03
	MSMO	6.42E−12	8.47E−08	6.42E−09	1.88E−08
$f_{15}(x)$	SMO	−1.00000	−1.00000	−1.00000	4.17E−08
	ABC	−1.00000	−0.99350	−0.99880	1.60E−03
	BBO	−0.96760	−0.94160	−0.94510	7.90E−03
	CMA-ES	−0.96890	−7.78E−05	−0.51710	4.80E+01
	JADE	−0.68040	−0.58990	−0.63430	2.62E−02
	SADE	−0.99980	−0.99970	−0.99970	2.23E−05
	MSMO	**−1.00000**	**−1.00000**	**−1.00000**	**3.64E−10**
$f_{16}(x)$	SMO	1.23E+06	9.98E+07	3.70E+07	2.67E+07
	ABC	1.38E+08	9.27E+08	2.93E+08	2.64E+08
	BBO	2.08E+07	1.19E+08	7.03E+07	3.58E+07
	CMA-ES	3.64E+08	4.68E+08	4.10E+08	2.85E+07
	JADE	1.22E+07	1.48E+07	1.33E+07	6.58E+05
	SADE	4.53E+07	5.45E+07	5.12E+07	2.01E+06
	MSMO	**9.84E+02**	**7.79E+05**	**1.29E+05**	**2.40E+05**
$f_{17}(x)$	SMO	6.15E+06	2.68E+08	7.71E+07	7.03E+08
	ABC	6.33E+07	1.31E+09	5.78E+08	3.37E+08
	BBO	7.06E+07	8.49E+08	2.53E+08	2.25E+08
	CMA-ES	1.33E+09	1.49E+09	1.39E+09	3.79E+07
	JADE	7.28E+07	8.08E+07	7.67E+07	2.22E+07
	SADE	2.09E+08	2.30E+08	2.20E+08	5.47E+08
	MSMO	**6.77E+03**	**1.10E+06**	**1.93E+05**	**2.71E+05**
$f_{18}(x)$	SMO	6.97E−00	2.27E+01	1.46E+01	4.13E−00
	ABC	1.42E+01	2.49E+01	2.03E+01	3.11E−00
	BBO	**0.000000**	5.18E+01	1.99E+01	2.50E+01
	CMA-ES	3.42E+01	3.98E+01	3.42E+01	1.56E+00
	JADE	3.96E+01	4.04E+01	4.01E+01	2.22E−01
	SADE	4.17E+01	4.22E+01	4.19E+01	1.44E−01
	MSMO	3.98E−01	**6.08E−00**	**2.13E−00**	**1.43E−00**
$f_{19}(x)$	SMO	2.28E+01	4.50E+01	3.48E+01	5.66E−00
	ABC	3.77E+01	5.52E+01	4.69E+01	4.85E−00
	BBO	**0.000000**	9.25E+01	7.04E+01	3.62E+01
	CMA-ES	6.54E+01	7.07E+01	6.86E+01	1.55E−00
	JADE	7.38E+01	7.48E+01	7.43E+01	2.48E−01
	SADE	5.85E+02	5.98E+02	5.92E+02	3.73E−00
	MSMO	7.71E−00	**1.82E+01**	**1.19E+01**	**2.60E−00**

Best values are in bold

4.2.1 Equally Spaced Linear Array

In order to validate the efficacy of MSMO for real problems, in the first instance, it is used to optimize two equally spaced LAA of different sizes. The goal is to have a set of optimized element amplitude excitations of LAA so that its radiation pattern has low SLL. In the equally spaced array, the distance between the array elements is uniform, i.e., $x_n = \lambda/2$ and also has uniform phase excitation $\varphi_n = 0°$. The position of the first element is taken as $x_1 = \lambda/4$. The AF given in (11) of equally spaced array now becomes:

$$\mathrm{AF}(\emptyset) = 2\sum_{n=1}^{N} I_n \cos\left[(n - .5)\,\pi \cos(\emptyset)\right] \tag{12}$$

Table 4 *p* test comparison of various algorithms

Objective function	SMO	ABC	BBO	CMA-ES	JADE	SADE	MSMO
$f_1(x)$	1.08E−**06**	**6.67E−06**	**6.79E−08**	**6.79E−08**	0.0071	6.79E−**08**	NA
$f_2(x)$	6.79E−**08**	6.79E−**08**	6.79E−**08**	6.79E−**08**	6.79E−**08**	6.79E−**08**	NA
$f_3(x)$	0.5737	**1.12E−08**	**1.12E−08**	**1.12E−08**	**1.12E−08**	**1.12E−08**	**NA**
$f_4(x)$	NA	**8.00E−09**	**8.00E−09**	**8.00E−09**	**8.00E−09**	**8.00E−09**	**5.76E−05**
$f_5(x)$	5.22E−**07**	**8.00E−09**	**6.03E−08**	**2.99E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_6(x)$	9.17E−**08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_7(x)$	8.00E−**09**	**8.00E−09**	**NA**	**5.48E−05**	**8.00E−09**	**8.00E−09**	**0.0034**
$f_8(x)$	7.89E−**08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_9(x)$	0.9461	**6.79E−08**	**6.79E−08**	**5.22E−07**	**6.79E−08**	**6.79E−08**	**NA**
$f_{10}(x)$	0.4735	**0.0077**	0.0033	0.0294	6.79E−**08**	1.65E−**07**	NA
$f_{11}(x)$	0.1404	**5.09E−04**	0.0017	1.61E−**04**	6.79E−**08**	6.79E−**08**	NA
$f_{12}(x)$	0.5428	**4.60E−04**	0.0041	0.0071	6.79E−**08**	9.74E−**06**	NA
$f_{13}(x)$	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**	**6.79E−08**	**6.79E−08**	**6.79E−08**
$f_{14}(x)$	8.00E−**09**	**NA**	**7.50E−09**	**8.00E−09**	**8.00E−09**	**8.00E−09**	**8.00E−09**
$f_{15}(x)$	2.95E−**07**	**6.79E−08**	1.51E−**08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_{16}(x)$	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_{17}(x)$	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_{18}(x)$	**6.79E−08**	**6.79E−08**	**0.2788**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**
$f_{19}(x)$	**6.79E−08**	**6.79E−08**	**1.30E−05**	**6.79E−08**	**6.79E−08**	**6.79E−08**	**NA**

Best values are in bold

In order to achieve the desired goal, the following fitness function is used:

$$\begin{aligned}&\text{Minimize fit} = \max\{20\log|\text{AF}(\varnothing)|\}\\&\text{subject to } \varnothing \in [0, \varnothing_1]\,\&\,[\varnothing_2, 180°]\end{aligned} \tag{13}$$

where $[0, \varnothing_1]\,\&\,[\varnothing_2, 180°]$ is the region other than the main lobe, i.e., side lobe region.

The first example shows the synthesis of a 16-element array for suppressed SLL in the region $\varnothing \in [0, 80°]\,\&\,[100, 180°]$. The number of parameters in this optimization, i.e., number of amplitude excitations to be optimized are eight since it is a symmetric array. The search region for MSMO is [0, 1]. The population size for MSMO is taken as 50 with number of iterations equal to 100. The normalized results of MSMO optimization are given in Table 5. The convergence graph of MSMO and TM [18] are shown in the Fig. 5 which shows that convergence of MSMO is pretty faster than TM [18]. The maximum SLL obtained by MSMO is −33.24 dB, while maximum SLL of the uniform array (having equal element amplitude excitations) is −13.54 dB. The SLL obtained by PSO [24], TS [22], TM [18], SADE [18] and BBO [29] antenna is −30.7, −26.2, −31.21, −31.06 and −33.06 dB, respectively, as shown in Table 6. The 3-dB beamwidth (BW) of all the different arrays is also compared as it is well known fact that as the SLL decreases, the beamwidth increases [40]. The beamwidth of MSMO is slightly higher than that of PSO [24], TS [22] and SADE [18] arrays but lower than TM [18] and BBO [29]. The radiation pattern of PSO [24], TS [22] and MSMO in azimuth plane ($x-y$ plane) is shown in Fig. 6.

The second example illustrates the design of 24-element antenna using MSMO for the same objective as in the previous example. In this optimization problem, the number of parameters are 12 with $\varnothing \in [0°, 82°]\,\&\,[98°, 180°]$. The amplitudes obtained after optimization are given in Table 7. The comparison of results for SLL obtained using different algorithms is shown in Table 8. The maximum SLL obtained by MSMO is −37.52 dB which is lower than 3.02, 9.98, 3.02, 2.27, 2.31 dB than PSO [24], TS [22], TM [18], CS [20] and SADE [18] optimized arrays, respectively. The convergence graph shown is Fig. 7 shows that MSMO converges toward the final solution at faster rate than TM [18]. The radiation pattern of TS [22], PSO [24] and MSMO optimized antenna arrays is shown in Fig. 8.

4.2.2 Unequally Spaced Linear Array

The synthesis of unequally spaced array has attracted attention of number of researchers in the recent past [17,19,25–28,32]. As against equally spaced arrays, the array factor of these arrays lack periodicity due to nonuniform element spacings. This aperiodic nature helps in achieving lower SLL with lesser number of elements for a given aperture size. Moreover, unequally spaced arrays save cost and complexity of feed network as lower SLL is achieved by uniform amplitude

Fig. 3 Convergence profiles of 19 different benchmark functions

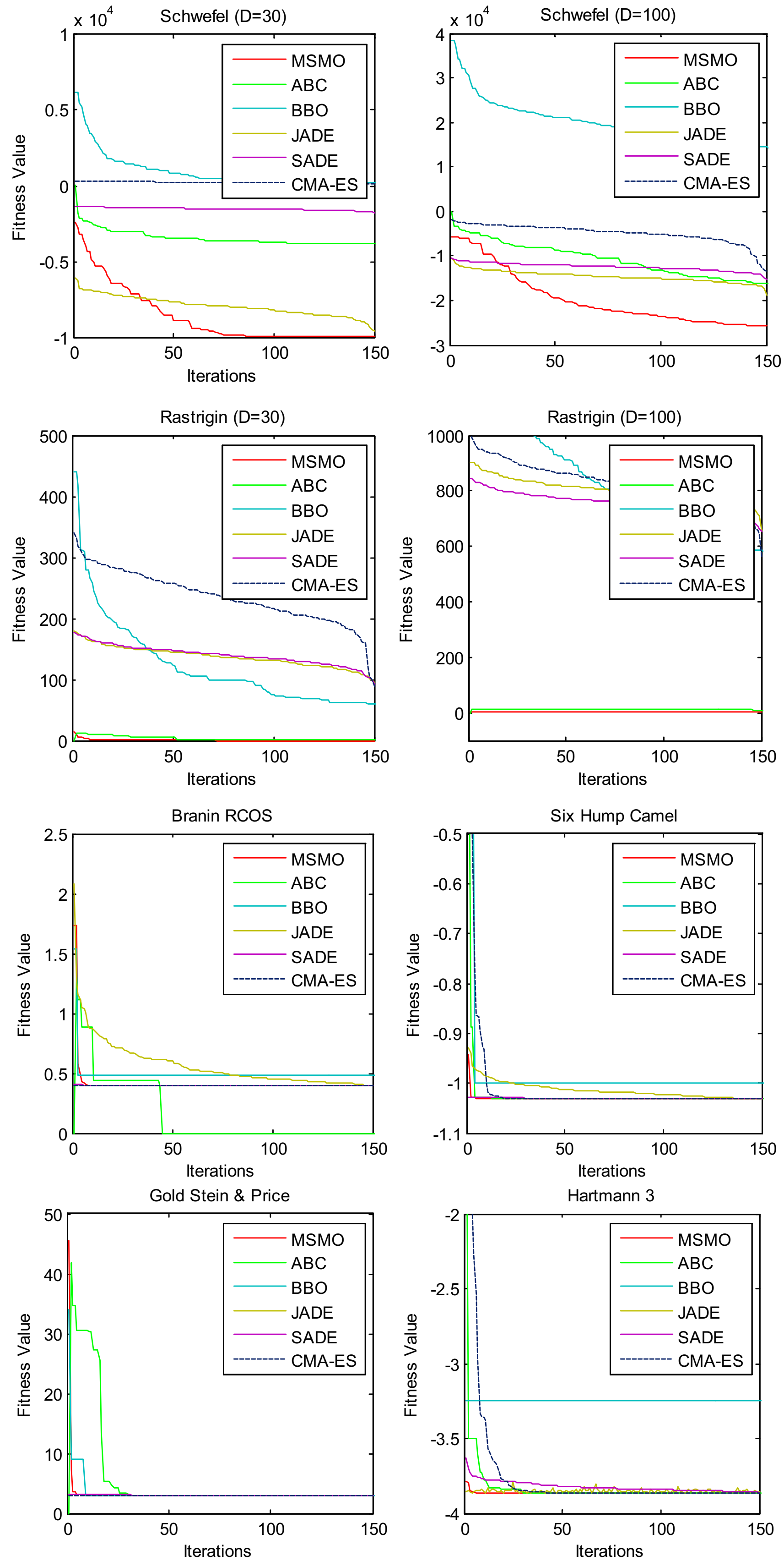


Fig. 3 continuous

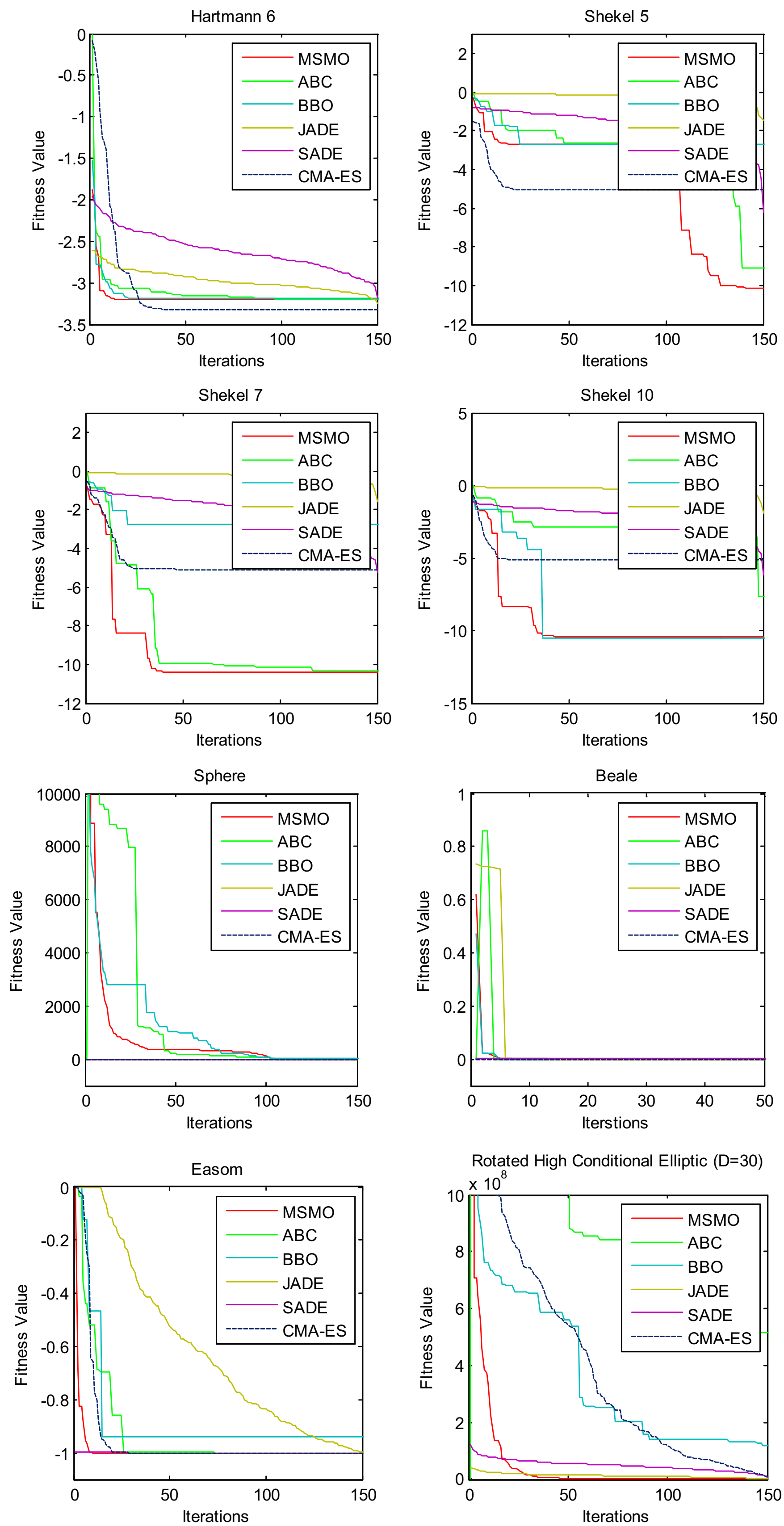
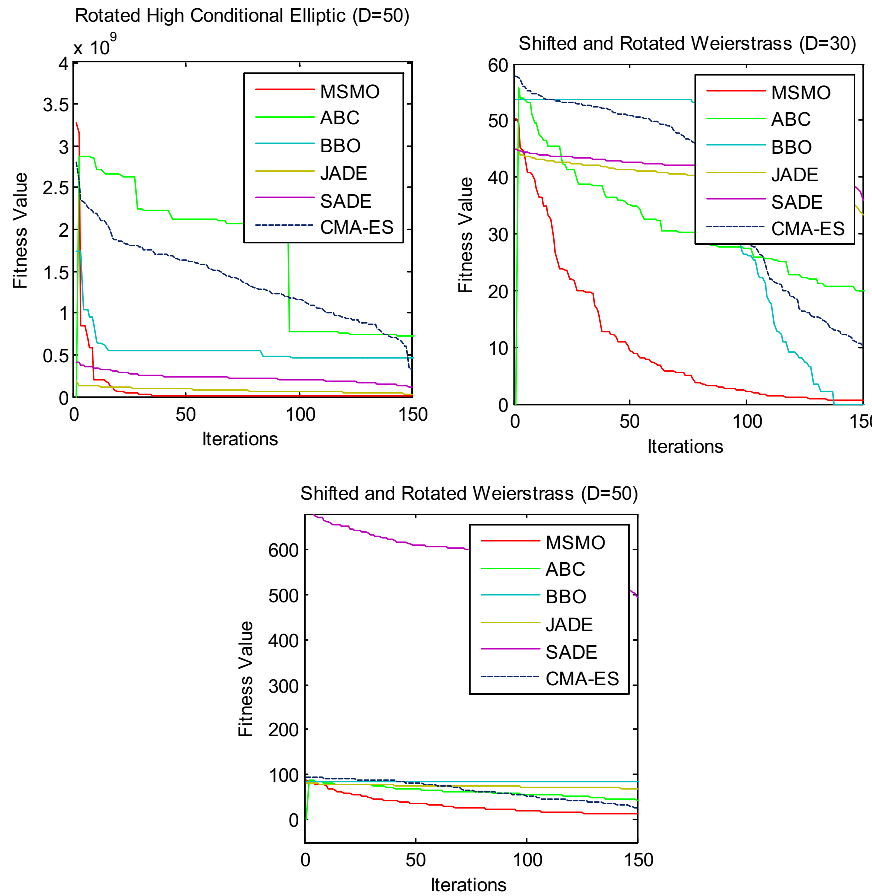


Fig. 3 continuous



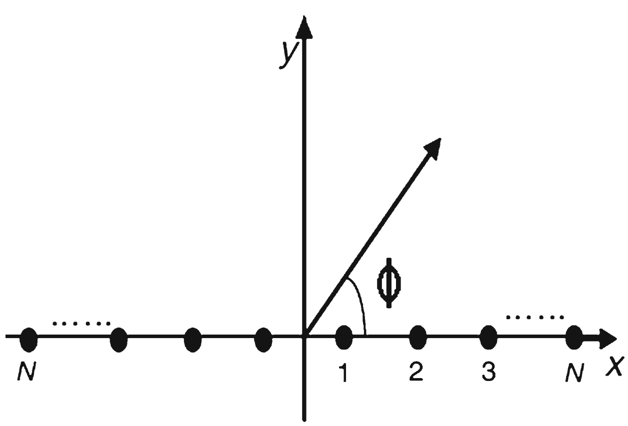


Fig. 4 A 2N-element linear antenna array

excitations. However, the design of unequally spaced array is hard due to nonlinear and non-convex relationship of its array function on element positions and excitation phases.

For a non-uniform array with uniform amplitude excitations, i.e., $I_n = 1$ and $\varphi_n = 0°$, the AF given in (11) becomes:

$$\mathrm{AF}\,(\phi) = 2\sum_{n=1}^{N} \cos\left[kx_n\, \cos\left(\phi\right)\right] \tag{14}$$

A 32-element non-uniform LAA is synthesized using MSMO. The aim is to have minimum SLL along with null placement in required directions with fixed beamwidth. The same problem has been investigated in the past using different optimization methods such as CS [20], PSO [25], inheritance learning particle swarm optimization (ILPSO) [26], comprehensive learning particle swarm optimization (CLPSO) [27], chaotic particle swarm optimization (CPSO) [28] and COA [32]. So, it makes sense to evaluate the performance of MSMO for this array in order to make comparison of performance of MSMO with the other optimization methods. The fitness function is also the same taken in the literature by CLPSO and is given by:

Table 5 Optimized element amplitudes of 16-element LAA obtained using MSMO

Element	1	2	3	4	5	6	7	8
Amplitude	1.0000	0.9613	0.7249	0.8346	0.5556	0.3977	0.2842	0.1844

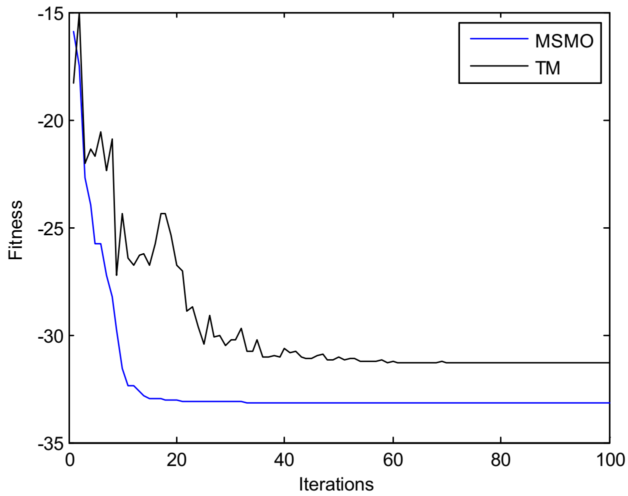


Fig. 5 Convergence graph of TM and MSMO for 16-element uniform LAA

Table 6 Comparison of results of different algorithms for 16-element LAA

Algorithm	PSO [24]	TS [22]	TM [18]	BBO [29]	SADE [18]	MSMO
SLL (dB)	−30.7	−26.2	−31.21	−33.06	−31.06	−33.24
BW (°)	8.2	8.0	8.4	8.4	8.2	8.3

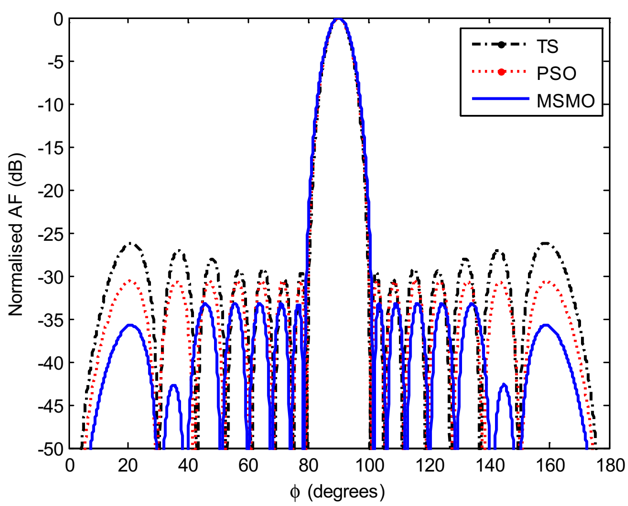


Fig. 6 Radiation pattern of 16-element uniform LAA

$$\text{fit} = \text{SLL} + \alpha * \max\left\{0, \left|\text{FNBW}-\text{FNBW}_d\right| - 1\right\} + \alpha * \left\{\sum_{k=1}^{K} \max\left\{0, \text{AF}_{\text{dB}}\left(\phi_k\right) - \text{Nu}_{\text{dB}}\right\}\right\} \quad (15)$$

where SLL is the peak side lobe level, FNBW, FNBW_d are the calculated and desired first null beamwidth, respectively, K is the number of the required null directions, Nu_{dB} is the desired null level in dB, ϕ_k is the direction of the kth null and α is a very large number. The fitness function given in

Table 7 Optimized element amplitudes of 24-element LAA obtained using MSMO

Element	1	2	3	4	5	6
Amplitude	1	0.9717	0.9195	0.8438	0.7555	0.6565
Element	7	8	9	10	11	12
Amplitude	0.5278	0.4534	0.3194	0.2430	0.1818	0.1296

Table 8 Comparison of results of different algorithms for 24-element LAA

Algorithm	PSO [24]	TS [22]	TM [18]	CS [20]	SADE [18]	MSMO
SLL (dB)	−34.50	−27.54	−35.25	−34.50	−35.21	−37.52
BW (°)	5.6	5.4	5.8	5.6	5.8	5.8

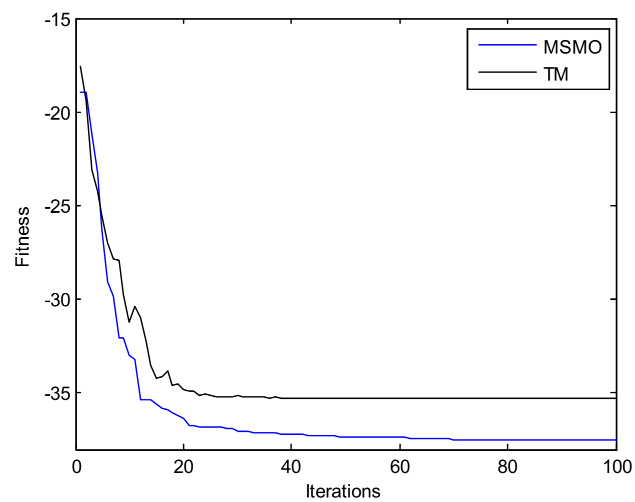


Fig. 7 Convergence graph of TM and MSMO for 24-element uniform LAA

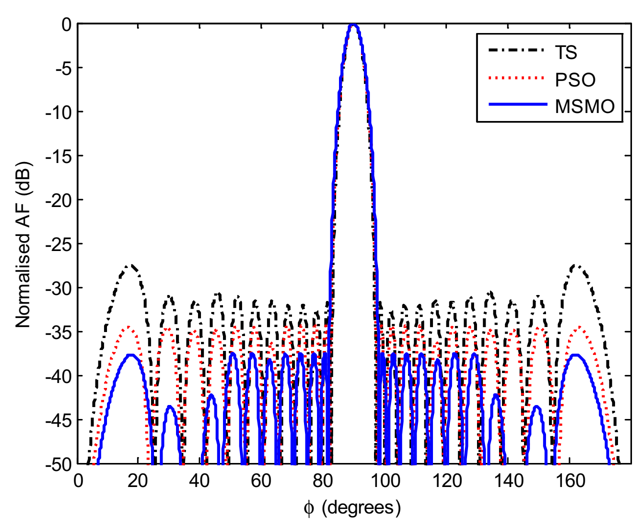


Fig. 8 Radiation pattern of 24-element uniform LAA

Table 9 Optimized element positions of 32-element LAA obtained using MSMO

Element	1	2	3	4	5	6	7	8
Position	0.4326	1.2628	2.0280	2.9408	4.0305	4.7722	5.5672	6.6195
Element	9	10	11	12	13	14	15	16
Position	7.6483	8.8333	9.8381	11.0075	12.5954	14.2951	15.9951	17.6929

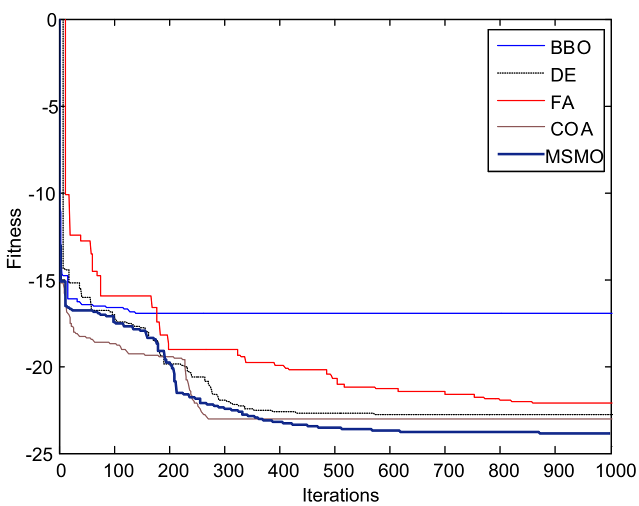


Fig. 9 Convergence graph of different algorithms for 32-element non-uniform LAA

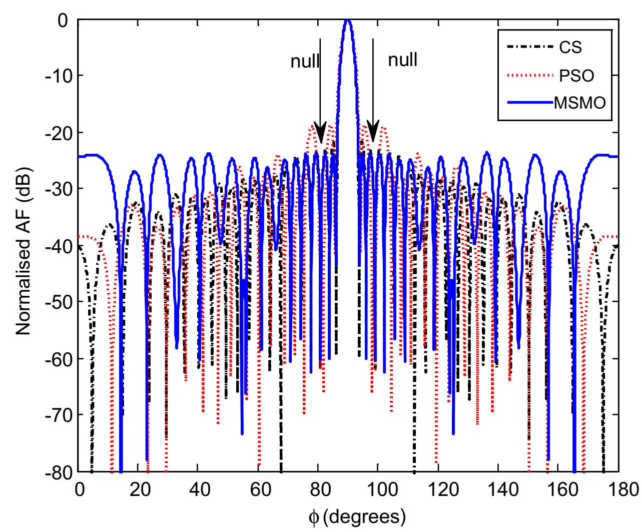


Fig. 10 Radiation pattern of 32-element uniform LAA

(15) makes use of penalty method. In this method, the solutions that do not fulfill or violate the required constraints are given higher values. This is done by choosing the value of α very large. In the present work, the FNBW is calculated computationally from the radiation pattern data. The population size is taken as 40 with number of iterations set to 1000. The value of α in the fitness function is taken as 10^6. In this optimization, the goal is to reduce SLL in the region $\emptyset \in [0, 87°] \,\&\, [93°, 180°]$ and to place null in the directions of 81° and 99° with the desired FNBW of 7.1° ± 1°. To achieve the objective, MSMO is applied to perturb the element positions of LAA while keeping the element excitations constant. Since this is a 32-element symmetric array, the number of element positions needed to be optimized are 16. The value of null level desired (Nu_{dB}) is −60 dB.

The MSMO optimized element positions are shown in Table 9. The convergence graph of MSMO is plotted in Fig. 9 and compared with the convergence of COA [32], FA [32], BBO [32], DE [32]. For fair comparison, the number of iterations for the algorithms to COA [32], FA [32], BBO [32], DE [32] are also limited to 1000. As it is clear from the convergence graphs, the speed of convergence of MSMO is fastest as compared to other algorithms. Table 10 compares the performance values of different algorithms for 32-element LAA. The SLL achieved by MSMO is lowest and is −23.85 dB and at the same time it has null value of −60 dB which is desired. The radiation pattern of MSMO optimize antenna array is shown in Fig. 10.

5 Conclusions

This paper proposed a new algorithm namely MSMO based on dual-search strategy. The experimental results show that MSMO algorithm is better than SMO, ABC, BBO, JADE, SADE and CMA-ES in terms of consistency in finding global optimum solution and convergence speed for numerical optimization problems. In addition, as an application to real world, MSMO has been used to synthesize LAA for three different cases. The results obtained by MSMO are better than

Table 10 Comparison of results of different algorithms for 32-element LAA

Algorithm	PSO [25]	CLPSO [27]	CPSO [28]	ILPSO [26]	FA [32]	BBO [32]	CS [20]	COA [32]	MSMO
SLL (dB)	−18.80	−22.73	−23.17	−23.75	−22.64	−16.93	−22.83	−23.81	−23.85
Nulls (dB)	−62.12	−60	−63.16	−73	−60.59	−61.73	−62.63	−79.85	−60

those of TS, TM, PSO, CLPSO, CPSO, ILPSO, CS, COA, DE and FA in terms of lower SLL and convergence speed. Results demonstrate that MSMO is capable of solving difficult antenna optimization problems, and it has a potential to become an alternative tool to popular competitive algorithms for solving real-world problems.

References

1. Dorigo, M.; Birattari, M.: Ant Colony Optimization. Encyclopedia of Machine Learning, pp. 36–39. Springer, USA (2010)
2. Dorigo, M.; Di Caro, G.: Ant Colony optimization: a new meta-heuristic. In: Proceedings of Congress on Evolutionary Computation, Washington, DC, USA (1999)
3. Kennedy, J.; Eberhart, R.C.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
4. Passino, K.M.: Bacterial foraging optimization. Int. J. Swarm Intell. Res. (IJSIR) **1**(1), 1–16 (2010)
5. Karaboga, D.: An idea based on honey bee swarm for numerical optimization. Erciyes University, Eng. Faculty, Comput. Eng. Dept., Tech. Rep. TR-06 (2005)
6. Bansal, J.; Sharma, H.; Jadon, S; Clerc, M.: Spider monkey optimization algorithm for numerical optimization. Memet. Comput. **6**(1), 31–47 (2014)
7. Porto, V.: Evolutionary Programming, VII. Springer, Berlin (1998)
8. Mezura-Montes, E.; Velázquez-Reyes, J.; Coello, C.A.: A comparative study of differential evolution variants for global optimization. In: Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, pp. 485–492. ACM Press, New York (2006)
9. Lampinen, J.; Zelinka, I.: On stagnation of the differential evolution algorithm. In: Proceedings of MENDEL, Citeseer, pp. 76–83 (2001)
10. Shi, Y., Eberhart, R.: Parameter selection in particle swarm optimization. In: Evolutionary Programming, VII, pp. 591–600. Springer, Heidelberg (1998)
11. Angeline, P.; Karaboga, D.; Akay, B.: A comparative study of artificial bee colony algorithm. Appl. Math. Comput. **14**(1), 108–132 (2009)
12. Gao, W.-F.; Liu, S.-Y.: A modified artificial bee colony algorithm. Comput. Oper. Res. **39**(3), 687–697 (2012)
13. Simon, D.: Biogeography-based optimization. IEEE Trans. Evol. Comput. **12**(6), 702–713 (2008)
14. Hansen, N.: The CMA evolution strategy: a comparing review. In: Lozano, J.A., Larrañaga, P., Inza, I.; Bengoetxea, E. (eds.) Towards a New Evolutionary Computation. Advances in Estimation of Distribution Algorithms, pp. 75-102. Springer, Berlin (2006)
15. Zhang, J.; Sanderson, A.C.: JADE: Self-adaptive differential evolution with fast and reliable convergence performance. In: Proceedings of IEEE Congress in Evolutionary Computation, Singapore, pp. 2251–2258 (2007)
16. Qin, A.K.; Suganthan, P.N.: Self-adaptive differential evolution algorithm for numerical optimization. In: Proceedings of IEEE Congress on Evolutionary Computation, vol. 2, pp. 1785–1791 (2005)
17. Cen, L.; Ser, W.; Yu, Z.L.; Rahardja, S.: An improved genetic algorithm for aperiodic array synthesis. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, Las Vegas, NV, pp. 2465–2468 (2008)
18. Dib, N.; Goudos, S.; Muhsen, H.: Application of taguchi's optimization method and self-adaptive differential evolution to the synthesis of linear antenna arrays. PIER **102**, 159–180 (2010)
19. Lin, C.; Qing, A.; Feng, Q.: Synthesis of unequally spaced antenna arrays by using differential evolution. IEEE Trans. Antennas Propag. **58**(8), 2553–2561 (2010)
20. Khodier, M.: Optimisation of antenna arrays using the cuckoo search algorithm. IET Microwaves Antennas Propag. **7**(6), 458–464 (2013)
21. Zaman, M.; Abdul Matin, M.: Nonuniformly spaced linear antenna array design using firefly algorithm. Int. J. Microwave Sci. Technol. **2012**, 1–8 (2012)
22. Merad, L.; Bendimerad, F.; Meriah, S.: Design of linear antenna arrays for side lobe reduction using the tabu search method. Int. Arab. J. Inf. Technol. **5**(3), 219–222 (2008)
23. Jin, N.; Rahmat-Samii, Y.: Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations. IEEE Trans. Antennas Propag. **55**(3), 556–567 (2007)
24. Khodier, M.; Al-Aqeel, M.: Linear and circular array optimization: a study using particle swarm intelligence. PIER B **15**, 347–373 (2009)
25. Khodier, M.M.; Christodoulou, C.G.: Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization. IEEE Trans. Antennas Propag. **53**(8), 2674–2679 (2005)
26. Liu, D.; Feng, Q.; Wang, W.-B.; Yu, X.: Synthesis of unequally spaced antenna arrays by using inheritance learning particle swarm optimization. PIER **118**, 205–221 (2011)
27. Goudos, S.K.; Moysiadou, V.; Samaras, T.; Siakavara, K.; Sahalos, J.N.: Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control. IEEE Antennas Wirel. Propag. Lett. **9**, 125–129 (2010)
28. Wang, W.; Feng, Q.; Liu, D.: Application of chaotic particle swarm optimization algorithm to pattern synthesis of antenna arrays. PIER **115**, 173–189 (2011)
29. Sharaqa, A.; Dib, N.: Design of linear and elliptical antenna arrays using biogeography based optimization. Arab. J. Sci. Eng. **39**(4), 2929–2939 (2013)
30. Singh, U.; Kamal, T.S.: Optimal synthesis of thinned arrays using biogeography based optimization. PIERM **24**, 141–155 (2012)
31. Singh, U.; Kumar, H.; Kamal, T.S.: Linear array synthesis using biogeography based optimization. PIER M **11**, 25–36 (2010)
32. Singh, U.; Rattan, M.: Design of linear and circular antenna arrays using cuckoo optimization algorithm. PIER C **46**, 1–11 (2014)
33. Symington, M.M.F.: Fission–fusion social organization. Ateles Pan. Int. J. Primatol. **11**(1), 47–61 (1990)
34. Chapman, C.; Lefebvre, L.: Manipulating foraging group size: spider monkey food calls at fruiting trees. Anim. Behav. **39**(5), 891–896 (1990)
35. Bonabeau, E. et al.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
36. Yurtkuran, A.; Emel, E.: A modified artificial bee colony algorithm for p-center problems. Sci. World J. **2014**, Article ID 824196, 9 pp. (2014). doi:10.1155/2014/824196
37. Jamil, M.; Yang, X.-S.: A literature survey of benchmark functions for global optimization problems. Int. J. Math. Model. Numer. Optim. **4**(2), 150–194 (2013)
38. Liang, J.J.; Qu, B.Y.; Sugathan, P.N.: Problem definitions and evaluation criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization. Technical Report, Nanyang Technological University, Singapore (2013)
39. Derrac, J.; García, S.; Molina, D.; Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evolut. Comput. **1**, 3–18 (2011)
40. Balanis, C.: Antenna Theory-Analysis and Design, 2nd edn. Wiley, London (1997)