Progress In Electromagnetics Research B, Vol. 54, 385–405, 2013

MODIFIED BAYESIAN OPTIMIZATION ALGORITHM FOR SPARSE LINEAR ANTENNA DESIGN

Bui Van Ha[1, *], **Paola Pirinoli**[2], **Riccardo E. Zich**[1], **Marco Mussetta**[1], **and Francesco Grimaccia**[1]

[1]Dipartimento di Energia, Politecnico di Milano, Via Lambruschini 4, Milano 20156, Italy

[2]Dipartimento di Elettronica e Telecomunicazioni, Politecnico di Torino, C. Duca degli Abruzzi 24, Torino 10129, Italy

Abstract—In this paper, a modified Bayesian Optimization Algorithm (BOA), named M-BOA, is proposed to introduce a suitable mutation scheme for the traditional procedure in order to speed up the convergence of the algorithm and to avoid it to be trapped in local minima or to stagnate in suboptimal solutions. The proposed algorithm has been applied both to a specific mathematical test function and to sparse linear antenna arrays design, showing outperforming capabilities not only with respect to the standard BOA, but also with respect to other assessed global optimization methods.

1. INTRODUCTION

The design of complex electromagnetic (EM) structures for real life applications often requires exploiting the features of evolutionary computation techniques such as the classic Genetic Algorithms (GA) [1], Particle Swarm Optimization (PSO) [2], Ant Colony Optimization (ACO) [3] as well as more recent developed population based approaches such as MetaPSO [4], Memetic Algorithm [5], Invasive Weed Optimization (IWO) [6], Biogeography-Based Optimization [7] and other hybrid techniques [8–10]. All these population based techniques share the same basic idea, i.e., they attempt to reach the optimum solution acting at each step of the iterative process on the current population, i.e., on a considered set of candidate solutions, through general, problem-independent operators. These, however, could be

Received 18 September 2012, Accepted 16 September 2013, Scheduled 21 September 2013

* Corresponding author: Bui Van Ha (ha.bui@mail.polimi.it).

insufficiently effective and therefore they could lead to lose the best solutions, or, at least, to slow down the procedure convergence.

Obviously this risk is greater in case of complex optimization problems, as the EM ones are, but it can also occur in simpler cases. In literature two different approaches have been usually proposed to tackle this limitation of the pseudo-stochastic algorithms. One of these is based on an idea to use the information available from the entire set of the most promising solutions to generate the new offspring, in opposite to what is done for instance in the GA, where only the genes of two parents concur to the chromosome structure of a child. This principle has been implemented in the Estimation of Distribution Algorithms (EDAs) [11], based on the estimate of the distribution of the promising solutions and on the generation of a new population according to this estimate.

Among the EDAs, particularly interesting is the Bayesian Optimization Algorithm (BOA) [12, 13], that uses a technique derived from the idea of extracting from the available data the information theory content to mathematically model the possible interactions among all the most promising solutions and to estimate their joint distribution. The standard version of the BOA, however, usually lacks of exploration, since it attempts to build up a probabilistic model from the available knowledge, and may be easily trapped in local optima if the starting sampling of the problem is not suitably chosen. In order to overcome such a limitation and to increase the Bayesian exploration capability, in this paper a modified and enhanced version of the BOA (M-BOA in the following) is proposed by adding a suitable mutation scheme. Some preliminary results on its application to the design of microstrip filters have already been presented in [14]; here the M-BOA scheme is described for the first time in detail, and a larger set of results of its applications to different optimization problems is presented. In fact, in order to test its efficiency and to compare its performances with those of the standard BOA, as well as of other well-established algorithms, as the GA and the PSO, the M-BOA has been at first applied to several benchmark functions, and then to the design of several linear arrays, with constrained radiation pattern.

The choice of these applications for the testing of the M-BOA is due to the remarkable practical importance of linear arrays, whose optimization has received a great attention in the electromagnetic community. In previous works [15–21], the GA and the PSO have been successfully applied to the design of linear or planar antenna arrays. Recently, Taguchi method has also been implemented for the synthesis of linear arrays [22]. All these optimization approaches generally allow to obtain optimum solution using as free parameters the position and/or the excitation of the array elements, while their number is fixed "a priori". Recently, some published papers [23–26] proved the possibility to fulfill the design constraints while reducing the number of elements in non-uniformly spaced arrays. In [23, 24] the number of elements in the array is achieved by the singular value decomposition (SDV) approach, while the proper excitation and location of the elements is obtained with the matrix pencil method (MPM); in [25] sparseness constrained optimization is adopted, and finally in [26] the "probability" of different values of the number of array elements is provided, by sampling the distribution using Bayesian Interference (BI); this method, however, has to take into account the information provided by thousands of samples in order to get the proper probability distribution. Moreover, the distribution inferred by the BI approach is usually heavily affected by the initial assumptions, e.g., by the probability distribution assignment to all the parameters; therefore this approach could not guarantee that the obtained value is the absolute optimum one, but only the optimum of a solution subspace defined by the first guess choice, and this can dramatically affect not only the efficiency but even the robustness of all the procedure.

This is the main aspect that we took into account for improving the performances of BOA: in fact, even if the here introduced M-BOA also starts from initial assumptions, a suitable operator has been introduced in order not to limit its exploration to the solution subspace defined by the first guess vector choice. Moreover, by considering the number of elements as a variable, the proposed method will estimate the distribution of all possible values for the number of elements without any particular problem.

The paper is organized as it follows. In Section 2, the M-BOA is introduced, after a brief review of the standard BOA. In Section 3, the results of the application of the proposed algorithm to benchmark functions are reported. In Section 4 examples of application of the M-BOA to the design of unequally spaced linear array are shown and compared with the results obtained with other optimization techniques.

2. MODIFIED BAYESIAN OPTIMIZATION ALGORITHM

The Bayesian Optimization Algorithm (BOA), introduced in [12], uses probability theory for estimating the distribution of promising solutions for a specific problem. In the BOA, the interaction between parameters (i.e., variables) is considered to build a probabilistic model, i.e., a Bayesian Network (BN), that evolves during an iterative process towards increasingly good solutions, until the achievement of a global optimum.

In the BOA procedure, the BN is built taking advantages of the information from the selected most promising solutions and eventually the prior knowledge on the problem to be optimized, if already available. At each iteration, new candidate solutions are generated by sampling the BN, and then they are included in the population, in place of its worst elements. A full description of BOA can be found in [12, 13], but for sake of clarity and uniformity of the notation, it is briefly summarized in the following, before introducing the M-BOA.

2.1. Standard BOA

The BOA starts with randomly generating an initial population as a set of strings. The best solutions in the current population are then selected using a specific selection method such as the truncation selection or the tournament selection.

A Bayesian Network is then constructed to fit the selected set of strings. In the building network process, a metric and a search method are used to measure and maximize the quality of the Bayesian network. The new offspring is generated using the information encoded in the BN. Finally, a new population is obtained, substituting in the older one the worst strings with the new ones. The iterative algorithm proceeds until when stopping criteria are satisfied.

The pseudo-code of the BOA could be summarized in the following steps:

1. Randomly generate initial population (**P**);
2. Select a set of promising solutions (**S**) from (**P**);
3. Construct the network (**B**) using the information from the set of solutions at point (2) and the prior knowledge of the problem, if available;
4. Generate a new offspring (**O**) sampling **B**;
5. Create a new population (**P'**) by replacing some instances from (**P**) with (**O**);
6. If the stop criteria are not met, go to (2).

In the above pseudo-code, steps (3) and (4) are the most important and critical, since the accuracy and effectiveness of the entire algorithm depend on them.

The Bayesian Network [27] represents the structure of the problem, since it is essentially a mathematical and graphical model that combines probabilistic theory with graph theory to encode the relationship between variables contained in the modeled data. In the graph, each node represents one variable, and the edges between the nodes correspond to the conditional dependencies between two variables. Once constructed, the BN is used to generate new instances, resorting the conditional probability of each variable.

Mathematically, a BN encodes the joint probabilistic distribution:

$$p\left(\mathbf{X}\right) = \prod_{i=1}^{l} p\left(X_i \,|\, \pi_i\right) \tag{1}$$

where $\mathbf{X} = (X_1, X_2, \ldots, X_l)$ is the vector of variable of problem, π_i the set of parents of variable X_i, and $p(X_i|\pi_i)$ the conditional probability of X_i conditioned on the variables π_i.

In the BOA procedure, the BN has to be trained in order to properly fit the promising solutions which satisfy the design requirements. There are two basic components of the algorithm to perform the learning of the BN: the scoring metric and the search procedure.

The scoring metric quantifies the quality of the given network. As already said previously, all the prior knowledge about the problem can be included into the metric as well.

The search engine is used to explore the space of all possible networks in order to maximize the value of the scoring metric. The exploration is usually restricted by the problem constraints such as the maximum number of incoming edge to one node. This number directly influences the complexity of the algorithm in constructing the network and generating the related offspring. In this work, we chose K2 as scoring metric [28] and greedy algorithm as search procedure.

2.2. Modified BOA

As described above, BOA performances greatly depend on the distribution of the current good solutions. The critical point is that, in absence of available prior knowledge of the problem, the initial population for the BOA, in order to start to build up the BN, is randomly generated. Therefore, in some cases it could be possible that all the best solutions in the initial population would not provide good enough distribution, because they do not properly represent the solution space dimensionality, affecting the convergence capability of the algorithm itself. To overcome this problem, one possibility is to increase the population size: this may increase the quality of the sampling in terms of information quantity, and therefore may improve the distribution modeling of good solutions, but it also increases the algorithm computational cost for sure. On the contrary, the variation of the standard BOA proposed here presents the advantage of enhancing the method performances without increasing its computational cost, since it does not need a further sampling of the cost function. The resulting approach will be indicated in the following as Modified BOA (M-BOA).

The basic idea of the M-BOA is that of using not only the Bayesian Network for generating a new offspring but also a mutation scheme. Mutation is also characteristic of other optimization algorithms as the GA [1], Population Based Incremental Learning (PBIL) [29] and Compact Genetic Algorithm (cGA) [30]: thank to it, some individuals will be used to explore candidate solutions out of the considered distribution space and therefore, the algorithm:

- will avoid being trapped in local optima;
- could find good solutions out of the initial population, allowing it to have a reduced size.

Referring to the pseudo code of the BOA, the introduction of the mutation essentially affects steps 4 and 5 that are modified as in following:

1. Generate a set of new offspring (**O**) according to the joint distribution encoded by **B**; generate a set of new offspring (**O′**) applying mutation to the same set of promising solution (**S**) use to generate (**O**);
2. Create a new population (**P′**) by replacing some instances, the worst ones, from (**P**) with (**O**) and (**O′**).

The scheme adopted by the M-BOA for generating a new population is therefore the one sketched in Fig. 1. It is worth to notice that both the new population elements obtained through the application of the Bayesian Network or through mutation are derived from the same set of selected solutions, i.e., the best ones. This differs from other algorithms that use mutation, since they generally apply it to the worst elements of the population. The new population is

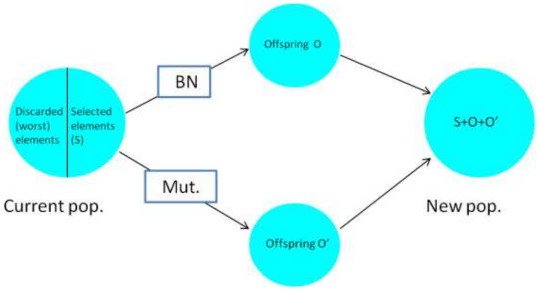


Figure 1. M-BOA new population generation scheme.

therefore given by the sum of the most promising elements of the previous population (**S**) plus the new offspring obtained applying to **S** the BN (**O**) or the mutation (**O′**).

Finally, notice that M-BOA works with variable vectors, which is much preferable for real variable problems or components, instead of probability vector as PBIL and cGA. In all the examples and tests problems considered in the following, the M-BOA with tournament selection and individual mutation have been adopted.

3. M-BOA VALIDATION: BENCHMARK FUNCTIONS

Very preliminary results for the modified BOA have been presented in [14]. In this section, the results of the testing of the M-BOA through its application to benchmark functions are presented, while in the next section real-life antenna problems will be considered.

In particular, the results shown here refer to the application of the M-BOA to the *Rosenbrock*, *Rastrigin*, *Ackley* (all with dimension $N = 15$) and *Shenkel* functions, and its comparisons with the standard BOA and the PSO (the expressions of these benchmark cost functions are reported in Appendix A for the sake of completeness). The three algorithms have been compared in term of both speed of convergence and reliability, considering for each of them the results obtained with 50 independent trials, each with 1000 iterations.

The curves of convergence relative to the application of the three optimization methods applied to the *Rosenbrock* function are shown in Fig. 2. The blue, green and red lines represent the average curves of convergence over the 50 trials for the BOA, the PSO and M-BOA, respectively. Comparing these three curves, it clearly appears that the M-BOA outperforms the other two schemes, since both BOA and PSO are inclined to stagnate.

Further performance comparisons are shown in Table 1, in which the average minimum values obtained by the three algorithms when applied to the different considered benchmark functions are listed, together with the standard deviation (in parenthesis). Both the BOA and M-BOA outperform the PSO, in terms of convergence and reliability. For what concerns the comparison between these two, they show both an (almost) zero standard deviation, but the average minimum value achieved with the M-BOA is much smaller, and this confirms the results in Fig. 2, i.e., the M-BOA converges better than the standard BOA.

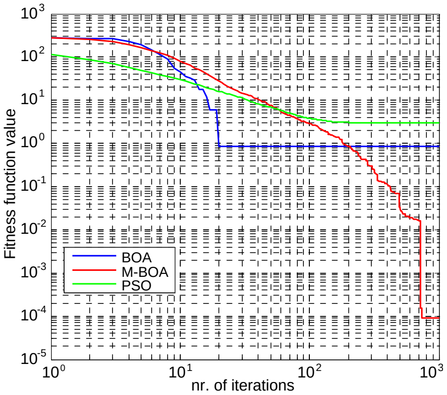


Figure 2. Curves of convergence relative to the PSO (green line), the BOA (blue lines) and M-BOA (red lines) applied to the *Rosenbrock* function.

Table 1. Comparison among PSO, BOA and M-BOA in terms of minimum value and standard deviation.

	Ackley	**Rosenbrock**	**Rastrigin**	**Shekel**
PSO	3.149 (0.768)	0.218 (0.039)	30.062 (9.48)	0.0022 (0.0178)
BOA	0.164 (0)	0.8384 (0)	0.1488 (0)	0.0013 (0)
M-BOA	0.032 (0.02)	0.000085 (0.0000128)	0.0092 (0.0012)	0.000026 (0.000016)

4. M-BOA BASED LINEAR ARRAY SYNTHESIS

In view of the promising results obtained on the benchmark functions, the M-BOA has been applied to more realistic antenna problems, i.e., the optimized design of three linear antenna arrays. To compare the results obtained by the M-BOA with those provided by other approaches, we have considered three examples of linear, broadside, sparse, symmetric arrays, already presented in [21–24, 26]. In all the considered cases the array is symmetric, and the optimization goal is that of determining the array element excitation coefficients, a_n, the position, normalized with respect to the wavelength λ, d_n/λ, and the minimum number $2N$ of array elements that allows to satisfy the array radiation constraints. Note that the estimation of the lowest number of necessary array elements is an important aspect, as also discussed in [24]. Most of the optimization techniques adopted for array synthesis [15–21] work with a fixed and predetermined number of array elements, which is generally taken larger than the one actually needed, in order to be sure to satisfy the radiation pattern constraints. However, the use of a not redundant number of elements is generally advantageous, since it allows to reduce the feeding network complexity and the antenna weight. For this reason, in the last years, several efforts have been made to propose techniques for the design of arrays, linear and planar, with a reduced number of elements not equally spaced [23–26]. Due to their probabilistic nature, BOA and M-BOA seem particularly suitable to determine, during the optimization process, also the proper number of array elements, and therefore the design of such kind of array seems a particularly suitable test case for comparing their performances.

4.1. Chebyshev-like Pattern Synthesis

The first considered example consists in the synthesis of an array showing a radiation pattern with the HPBW of at least 6.3°, the $-30\,\text{dB}$ beam width lower than 8° and the side lobe level $\leq -30\,\text{dB}$ (see the mask plotted later on in Fig. 6). Such requirements can be easily obtained with a uniformly spaced Chebyshev array with $2N = 20$ elements (this is the reason why we named "Chebyshev-like" this type of radiation pattern) while in [23, 26] the constraints have been satisfied with unequally spaced arrays with $N = 6$.

Here both the BOA and M-BOA have been used for the array optimized synthesis, according to the Chebyshev-like constraints above specified. In this test, we used a population of 100 individuals in both the procedures, considering 1000 independent trials of 200 iterations each. The fitness function that models the problem is defined as:

$$f\left(X\right) = \sum\nolimits_{i} \varepsilon\left(\theta_i\right) \tag{2}$$

where $\varepsilon(\theta_i)$ is the difference between the constraints mask (R_{mask}) and the radiation pattern (R_p) obtained at any iteration of the optimization process:

$$\varepsilon\left(\theta_i\right) = R_p\left(\theta_i\right) - R_{\text{mask}}\left(\theta_i\right) \tag{3}$$

It is worth noticing that Eq. (3) applies only when the obtained radiation pattern exceeds the constraints, while $\varepsilon(\theta_i) = 0$ otherwise.

The ranges of variation for the optimization variables are set equal to (0.25–5) for the normalized array element location (d_n/λ), (0–1) for the element excitation coefficient (a_n) and (5–9) for the array element pair number (N). In order to have a stronger validation of the M-BOA, it was also compared with the GA and the PSO.

First, the results of the comparison between the standard BOA and the M-BOA are presented. In Fig. 3 the worst (dot-dashed line), average (continuous line) and best (dashed line) curves of convergence relative to the two methods are plotted, showing that the M-BOA outperforms the BOA both in terms of capability of convergence (the average curve relative to M-BOA reaches a final value for the cost function that is almost two order of magnitude smaller than the one of the BOA) and reliability (the average curve converges, even if with a greater number of iterations, almost the same value as the best one; moreover, the average curve of convergence for the M-BOA is much closer to the best solution than to the worst one, and this means that the number of trials in which a solution close to the best one is reached is greater than the number of cases in which it is close to the worst one).

In Fig. 4 the average curve of convergence obtained with the M-BOA (red solid line in Fig. 3) is compared with the best curves of convergence obtained with a decreasing number of array elements: with $N = 6$ the algorithm convergence is still good and fast: this means that it is possible to satisfy the radiation pattern constraints with such a number of radiator pairs. The curve corresponding to $N = 5$ is not plotted since in that case the algorithm does not converge.

The second column of Table 2 shows the probability (computed as

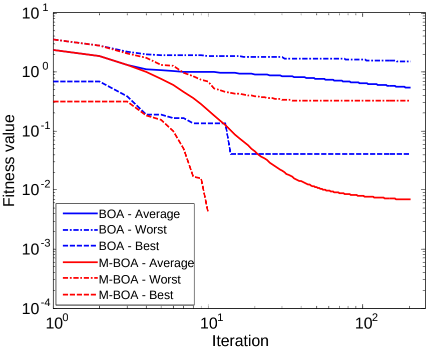


Figure 3. Chebyshev-like pattern synthesis: comparison between the BOA and the M-BOA in terms of convergence.

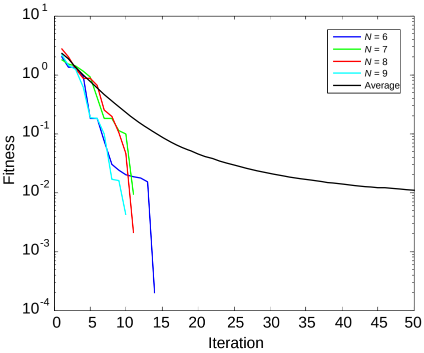


Figure 4. Chebyshev-like pattern synthesis: curves of convergences of the M-BOA, relative to different values of array element pairs.

the ratio between the times in which a good result is obtained and the total number of trials) of obtaining a good solution with different array sizes within a predefined number of iterations. Note that, as it can be also deduced from the above considerations, in that table any row corresponds to $N = 5$, since with such a small number of elements the desired radiation pattern is never achieved (the probability is therefore 0). Moreover, note that the probability to have good solutions first increases and then decreases. This can be explained with the fact that with a high value of N, i.e., 9, the number of variables that the M-BOA has to manage increases, and therefore, the convergence can not be reached in the considered number of iterations; on the other hand, when N is small, i.e., 6, it becomes difficult to find the proper values of a_n and d_n/λ that get the required pattern. In both cases, the algorithm hardly converges, and less good solutions are obtained. On the contrary, medium values of N, i.e., 7, 8, represent a good trade-off between the problem size and the ease of satisfying the pattern constraints. In the third column of the table, the probability values computed in [26] for the same array problem are reported: even if the case $N = 10$ is out of the range of variability of the number of element pairs considered here, its relative row has been added, just to show that also in [26] the sum of the probabilities is equal to 1. It is worth to note that both algorithms shown the ability to obtain the desired array performances with 6 elements, but the probability value relative to M-BOA is higher than that what derived in [26] for the Bayesian Interference (BI), and this confirms the better reliability of the M-BOA with respect to the BI.

Finally, the convergence of the M-BOA has been compared with that of the PSO and GA. For doing that, N has been fixed to 6. A population of 100 individuals has been considered, and the results have been averaged over 30 independent trials, each with 1000 iterations. The resulting average curves of convergence for the M-BOA, BOA,

Table 2. Probability of good convergence for different values of N.

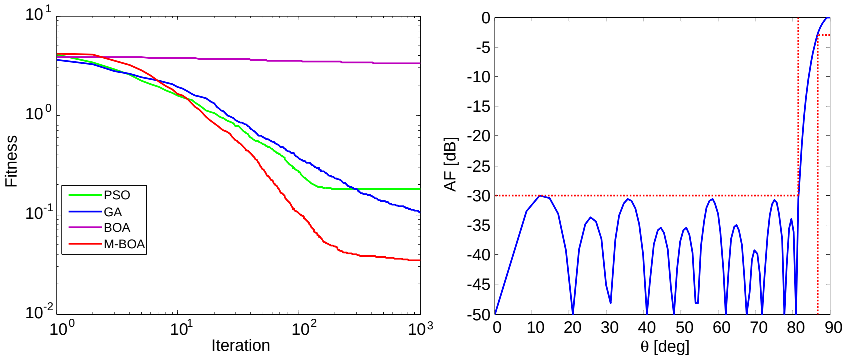
N	Convergence probability	
	M-BOA	BI [26]
6	0.182	0.1327
7	0.349	0.5377
8	0.296	0.2843
9	0.173	0.0393
10		0.006

PSO and GA are plotted in Fig. 5(a). As expected, the BOA can not converge since the number of array elements is too small, while the M-BOA outperforms both the GA and the PSO. The results show that not only the M-BOA has the capability of automatically setting the minimum number of elements needed to satisfy the radiation constraints, but, for a fixed N, it also converges faster than the GA and the PSO.

In Fig. 5(b), it is finally shown the radiation pattern for the $N = 6$ array designed with the M-BOA, together with the required mask, while the corresponding values of the element positions (d_n/λ) and of the excitation coefficients a_n are listed in Table 3. Note that the radiation pattern in Fig. 5(b) does not show uniform SLL as that obtained with the Chebyshev synthesis, but in any case the mask is satisfied, with a much smaller number of elements.

Table 3. Parameters of the $N = 6$ array designed with the M-BOA.

Element #	d_n/λ	a_n
1	0.4313	0.3657
2	1.3055	0.3282
3	2.1777	0.2769
4	3.0494	0.2037
5	3.9076	0.1316
6	4.7919	0.0815



(a)

(b)

Figure 5. Chebyshev-like pattern synthesis ($N = 6$). (a) Average curves of convergence relative to different methods. (b) Best array factor (AF) obtained by M-BOA for the $N = 6$ array.

4.2. Flat-beam (Sector Beam) Pattern Synthesis

The second considered example is the synthesis of a sector beam pattern with the following requirements: a first region ranging from 78.3° to 90°, where the beam has to be almost flat, with a ripple lower than 0.5 dB, and a second region, ranging from 0° to 69.3°, in which the side lobe level has to be less than −25 dB. Therefore, the cost function is the same as (3), considering $R_{\text{mask}}(\theta_i)$ as defined in Fig. 8(b).

This synthesis problem is effectively solved controlling the excitation amplitude and the phase of all the array elements [21, 22]. In [22], the use of Taguchi method allows fulfilling the radiation pattern constraints with an array of $N = 7$ pairs of elements, i.e., with 6 elements less than the one designed in [21] using PSO. As in the previous example, here the M-BOA is also used to estimate the minimum number of elements necessary to satisfy the radiation pattern requirements.

Since we have already proved that M-BOA outperforms BOA in terms of both speed of convergence and solution quality, the M-BOA is, from here on, only compared with the GA and PSO. In all cases, the ranges of variation for the optimization variables are set equal to (0.25–5) for the normalized array element location (d_n/λ), $(-0.5 \div 0.5)$ for the element excitation coefficient (a_n), while for the M-BOA only the number of element pairs can also vary between 4 and 9, since the problem can be easy solved with 10 pairs of elements.

In Fig. 6, the best curves of convergence of the M-BOA for different values of N are reported. They have been obtained using a population of 100 individuals and considering 800 independent trials, each of 200

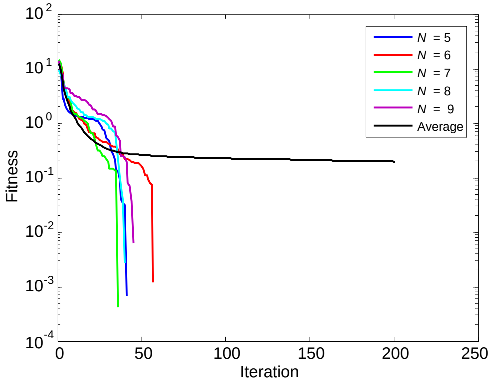


Figure 6. Sector beam pattern synthesis: curves of convergences of the M-BOA, relative to different values of array element pairs.

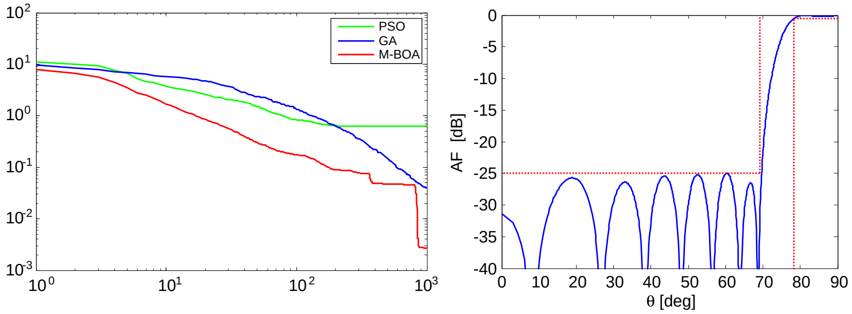
iterations. From this plot it appears that the problem can be solved even with $N = 5$ pairs of elements, since the convergence of M-BOA with $N = 5$ is really fast, and the pattern requirements are reached with two pairs less of elements than those of the array proposed in [22].

In Table 4, the probability of obtaining a good solution with different array sizes is listed. For a value smaller than 5 ($N = 4$), the algorithm cannot convergence to the desired solution, and the probability is 0. The probability for $N = 5$ is the highest, indicating that this is the sufficient number of element pairs necessary to obtain the desired radiation pattern. Moreover, when N increases, the probability of finding good solution decreases, since the problem complexity increases, and probably the population size and/or the number of iterations considered here are not still enough to guarantee the convergence. Finally, note also in [26] that an array of 10 elements satisfying the same constraints as here has been designed, but comparing the probabilities of obtaining good solutions in Table 4 with those in [26, Table III], it is possible to conclude that also in this case the M-BOA outperforms the Bayesian Interference method.

Table 4. Probability of good convergence with different values of N.

N	*M-BOA Convergence probability*
5	0.818
6	0.110
7	0.055
8	0.015
9	0.002

In Fig. 7(a) the average curve of convergence for the M-BOA in the case $N = 5$ is compared with those relative to the GA and PSO. For all the three methods, the adopted population has 50 individuals, while 30 independent trials have been considered, each with 1000 iterations. It can be seen that the GA converges better than the PSO, while the M-BOA outperforms both of them. The radiation pattern obtained with the M-BOA is plotted in Fig. 7(b), together with the mask it has to satisfy, while the design parameters are listed in Table 5. A very good flat beam has been obtained with a ripple of 0.22 dB, i.e., 0.36 dB less with respect to the ripple of the array designed with the Taguchi method [22].



(a) (b)

Figure 7. Sector beam pattern synthesis ($N = 5$). (a) Average curves of convergence relative to different methods. (b) Best array factor (AF) obtained by M-BOA.

Table 5. Parameters of the $N = 5$ array, designed with the M-BOA.

Element #	d_n/λ	a_n
1	0.3586	0.2767
2	1.0748	0.1420
3	2.4697	−0.0472
4	3.0955	−0.0246
5	4.1708	0.0245

4.3. Null-controlled Pattern Synthesis

The last example that we considered is the synthesis of an array whose radiation pattern presents nulls for specific directions [22, 26]. The main beam has to point to 90° with a HPBW of 7.4°, while the sidelobe levels have to be lower than −40 dB. Moreover, two nulls are desired between 50° and 60°. Such a radiation pattern has been obtained adopting GA, PSO, and Taguchi methods with arrays of 20 elements [18, 19, 22]. Here, the M-BOA is used in order to estimate the sufficient number of element to fulfill the requirements [26].

The ranges of variation are set equal to (0.25–5) for the normalized array element location (d_n/λ), ($-0.5 \div 0.5$) for the element excitation coefficient (a_n) and (5–9) for the array element pair number (N), assuming to be able to design the array with a number of elements lower than that used in [18, 19, 22]. The M-BOA uses a population of 100 individuals, and 800 independent trials, each of 200 iterations, have been considered. As previously, the cost function is given in Eq. (3), considering the $R_{\text{mask}}(\theta_i)$ shown later on in Fig. 9(b).

In Fig. 8, the average curve of convergence of M-BOA is plotted along with the best curve of convergence obtained for different values of N. The results reveal that the desired pattern can be obtained even with 6 pairs of elements, i.e., with a reduction of the 40% of the number of elements compared with the one in [22]. It is worth to note that the best result ever achieved for this problem is with $N = 7$ in [26], and this again confirms the better performances of the here proposed method with respect to direct sampling from Bayesian Interference.

In Table 6, the probability of achieving good solutions with different values of N is shown. For $N = 5$, the desired pattern is never obtained, while the probability increases with N, probably due to the fact that the constraints on the radiation pattern are harder to fulfill with a too small number of elements (we can also see in Fig. 8 that for larger values of N the convergence is faster). However, the probabilities to obtain good solutions with $N = 8$ or $N = 9$ are very close, showing that a further increases of the number of array element will not significantly improve the possibility of having good solutions.

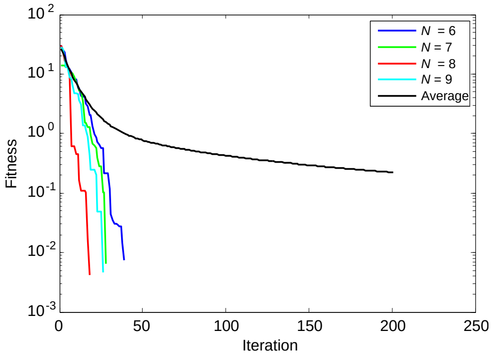
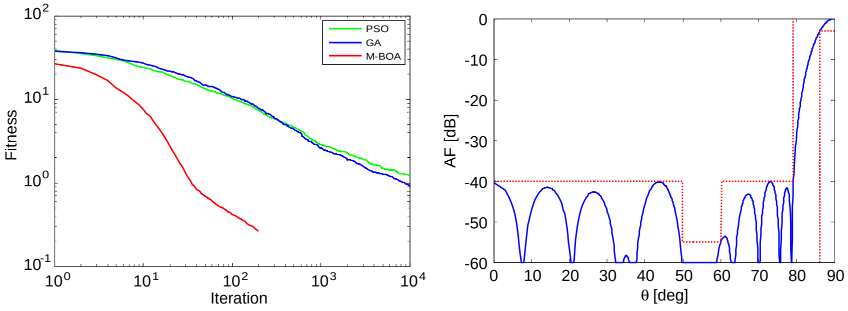


Figure 8. Null-controlled pattern synthesis: curves of convergences of the M-BOA, relative to different values of array element pairs.

Table 6. Probability of good convergence with different values of N.

N	*M-BOA Convergence probability*
6	0.105
7	0.255
8	0.327
9	0.313

Finally, the performances of the M-BOA have been tested again against those of the GA and PSO, setting $N = 6$, using a population of 100 individuals and averaging the results over 30 independent trials, while for the M-BOA it is sufficient to perform 200 iterations to reach the convergence, both the GA and the PSO needed 10000 iterations. The resulting average curves of convergence are plotted in Fig. 9(a): also in this case the M-BOA outperforms both the GA and the PSO in terms of computational cost and in term of solution quality, i.e., the M-BOA requires less iterations than PSO and GA, to reach a final value of the fitness function that is better than those of the PSO and the GA. The desired pattern for the twelve-elements null-controlled array, obtained by M-BOA, is plotted in Fig. 9(b), together with the mask. All the design parameters are instead reported in Table 7.



(a) (b)

Figure 9. Null-controlled pattern synthesis. (a) Average curves of convergence relative to different methods. (b) Best array factor (AF) obtained by M-BOA for the $N = 6$ array.

Table 7. Parameters of the $N = 6$ array, designed with the M-BOA.

Element #	d_n/λ	a_n
1	0.4221	0.2114
2	1.2611	0.1834
3	2.1023	0.1390
4	2.9476	0.0881
5	3.7805	0.0461
6	4.6220	0.0146

5. CONCLUSION

In this paper, a Modified Bayesian Optimization Algorithm (M-BOA) has been introduced. It shows improved performances with respect to the standard BOA, as well as other optimization algorithms as the GA and the PSO, both for what concerns the computational cost and the solution quality. Moreover, it has been shown, considering different examples, that it seems particularly suitable for the design of sparse arrays, since it also allows to easily include the number of elements in an array among the optimization parameters.

APPENDIX A.

The expressions of the standard cost functions [4] used in this article are:

a) **Ackley** Function ($N = 15$):

$$f(\mathbf{X}) = 20 + e - 20\, e^{-0.2\sqrt{\frac{1}{N}\sum_i x_i^2}} - e^{\frac{1}{N}\sum_i \cos(2\pi x_i)}$$

b) **Rosenbrock** Function ($N = 15$):

$$f(\mathbf{X}) = \sum_{i=1}^{N} \left(100\left(x_{2i} - x_{2i-1}^2\right)^2 + \left(1 - x_{2i-1}\right)^2\right)$$

c) **Rastrigin** Function ($N = 15$):

$$f(\mathbf{X}) = \sum_{i=1}^{N} \left(x_i^2 - 10\cos\left(2\pi \cdot x_i\right) + 10\right)$$

d) **Shekel** Function:

$$f(\mathbf{X}) = 12 - \sum_{i=1}^{9} \left(\left(\mathbf{X} - \mathbf{a}_i\right)^T \left(\mathbf{X} - \mathbf{a}_i\right) + c_i\right)^{-1}$$

where $\mathbf{X} = (x_1, x_2)$ is the variable vector, $\mathbf{a}_i$ are vectors of i-th local minima and c_i are constant proportional to minimum $f((a_i)^T) \approx 12 - \frac{1}{c_i}$. There are 9 minima at points $(-3,-3)$, $(-3,0)$, $(-3,+3)$, $\ldots$, $(+3,+3)$.

REFERENCES

1. Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wiley, 1989.

2. Kennedy, J. and R. C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Francisco, CA, 2001.
3. Dorigo, M., V. Maniezzo, and A. Colorni, "The ant system: Optimization by a colony of cooperating agents," *IEEE Trans. Syst., Man., Cybern. B*, Vol. 26, No. 2, 29–41, 1996.
4. Selleri, S., M. Mussetta, P. Pirinoli, R. E. Zich, and M. Matekovits, "Differentiated Meta-PSO method for array optimization," *IEEE Trans. Antennas Prop.*, Vol. 56, No. 1, 67–75, Jan. 2008.
5. Ong, Y. S. and A. J. Kean, "Meta-Lamarckian learning in memetic algorithms," *IEEE Trans. Evol. Comput.*, Vol. 8, No. 2, 99–110, Apr. 2004.
6. Basak, A., S. Pal, S. Das, A. Abraham, and V. Snasel, "A modified invasive weed optimization algorithm for time-modulated linear antenna array synthesis," *Proc. of IEEE Congress on Evol. Comput. (CEC)*, 1–8, Barcelona, Spain, Jul. 18–23, 2010.
7. Simon, D., "Biogeography-based optimization," *IEEE Trans. Evol. Comput.*, Vol. 12, No. 6, 702–713, Dec. 2008.
8. Grimaccia, F., M. Mussetta, and R. E. Zich, "Genetic swarm optimization: Self adaptive hybrid evolutionary algorithm for electromagnetics," *IEEE Trans. Antennas Prop.*, Vol. 55, No. 3, 781–785, Mar. 2007.
9. Lee, Y. H., B. J. Cahill, S. J. Porter, and A. C. Marvin, "Novel evolutionary learning technique for multi-objective array antenna optimization," *Progress In Electromagnetics Research*, Vol. 48, 125–144, 2004.
10. Perez-Lopez, J. R. and J. Basterrechea, "Hybrid particle swarm-based algorithms and their application to linear array synthesis," *Progress In Electromagnetics Research*, Vol. 90, 63–74, 2009.
11. Mühlenbein, H. and G. Paaß, "From recombination of genes to the estimation of distributions. I. Binary parameters," *Parallel Problem Solving from Nature*, Vol. 1141, 178–187, 1996.
12. Pelikan, M., D. E. Goldberg, and E. Cant-Paz, "BOA: The Bayesian optimization algorithm," *Proc. of Genetics and Evol. Comput. Conf., GECCO-99*, 525–532, Orlando, Florida, USA, Jul. 13–17, 1999.
13. Pelikan, M., *Hierarchical Bayesian Optimization Algorithm: Toward a New Generation of Evolutionary Algorithms*, Springer, 2005.
14. Ha, B. V., M. Mussetta, P. Pirinoli, and R. E. Zich, "Modified Bayesian optimization algorithm for microstrip filter design," *Proc. of IEEE AP-S 2012 Conf.*, 1–2, Chicago, Illinois, USA,

Jul. 8–14, 2012.

15. Yan, K. K. and Y. Lu, "Sidelobe reduction in array-pattern synthesis using genetic algorithm," *IEEE Trans. Antennas Prop.*, Vol. 45, No. 7, 1117–1122, Jul. 1997.
16. Ares-Pena, F. J., A. Rodriguez-Gonzalez, E. Villanueva-Lopez, and S. R. Rengarjan, "Genetic algorithms in the design and optimization of antenna array patterns," *IEEE Trans. Antennas Prop.*, Vol. 47, No. 3, 506–510, 1999.
17. Donelli, M., S. Caorsi, F. D. Natale, G. Franceschini, and A. Massa, "A versatile enhanced genetic algorithm for planar array design," *Journal of Electromagnetic Waves and Applications*, Vol. 18, No. 11, 1533–1548, 2004.
18. Boeringer, D. W., D. H. Werner, and D. W. Machuga, "A simultaneous parameter adaptation scheme for genetic algorithms with application to phased array synthesis," *IEEE Trans. Antennas Prop.*, Vol. 53, No. 1, 356–371, Jan. 2005.
19. Khodier, M. M. and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Trans. Antennas Prop.*, Vol. 53, No. 8, 2674–2679, Aug. 2005.
20. Donelli, M., R. Azaro, L. Fimognari, and A. Massa, "A planar electronically reconfigurable Wi-Fi band antenna based on parasitic microstrip structures," *IEEE Antennas and Wireless Prop. Letters*, Vol. 6, 623–626, 2007.
21. Gies, D. and Y. Rahmat-Samii, "Particle swarm optimization for reconfigurable phase-differentiated array design," *Microw. Opt. Tech. Lett.*, Vol. 38, No. 3, 168–175, Aug. 2003.
22. Chung, W. W., F. Yang, and A. Z. Elsherbeni, "Linear antenna array synthesis using Taguchi's method: A novel optimization technique in electromagnetics," *IEEE Trans. Antennas Prop.*, Vol. 55, No. 3, 723–730, Mar. 2007.
23. Liu, Y., Z. Nie, and Q. H. Liu, "Reducing the number of elements in a linear antenna array by the matrix pencil method," *IEEE Trans. Antennas Prop.*, Vol. 56, No. 9, 2955–2962, 2008.
24. Liu, Y., Q. H. Liu, and Z. Nie, "Reducing the number of elements in the synthesis of shaped-beam patterns by forward-backward matrix pencil method," *IEEE Trans. Antennas Prop.*, Vol. 58, No. 2, 604–608, Feb. 2010.
25. Zhang, W., L. Li, and F. Li, "Reducing the number of elements in linear and planar antenna arrays with sparseness constrained optimization," *IEEE Trans. Antennas Prop.*, Vol. 59, No. 8, 3106–

3111, Aug. 2011.

26. Chan, C.-Y. and P. M. Goggans, "Using bayesian inference for linear antenna array design," *IEEE Trans. Antennas Prop.*, Vol. 59, No. 9, 3211–3217, Sep. 2011.

27. Pearl, J., *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference,* Morgan Kaufmann, 1988.

28. Heckerman, D., D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," Technical Report MSR-TR-94-09, Microsoft Research, Redmond, WA, 1994.

29. Baluja, S., "Population based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Technical Report No. CMUCS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.

30. Harik, G. R., F. G. Lobo, and D. E. Goldberg, "The compact genetic algorithm," *IEEE Trans. Evol. Comput.*, Vol. 3, No. 4, 287–297, Nov. 1999.