

Modified Bayesian Optimization Algorithm for Planar Array Design

Bui Van Ha*, M. Mussetta*, F. Grimaccia*, P. Pirinoli†, R.E. Zich*

* Politecnico di Milano, Dipartimento di Energia, Milano, Italy

Email: ha.bui@mail.polimi.it, marco.mussetta@polimi.it, francesco.grimaccia@polimi.it, riccardo.zich@polimi.it

† Politecnico di Torino, Dipartimento di Elettronica e Telecomunicazioni, Torino, Italy

Email: paola.pirinoli@polito.it

Abstract—Design of electrical and electronic systems with complex EMC constraints requires often to exploit the peculiarities of some population based global optimizers. One of the main drawbacks of the adoption of these optimizers for system design is represented by the difficulty of introducing in the algorithm all the heuristic knowledge already available in the field. In order to overcome this problem, Bayesian optimization algorithms (BOAs), classified as estimation of distribution algorithm, can be very effective since they are based on the definition of distributions of promising solutions using the information extracted from the entire set of good solutions. Unfortunately, their straightforward implementations usually lack of exploration feature and they are easily trapped in local maxima. In order to overcome this drawback and to develop a Bayesian optimization algorithm with both exploitation and exploration mechanisms, in this paper a modified BOA is proposed by adding a suitable mutation scheme to the traditional one in order to ensure the effectiveness of the algorithm. The here proposed new algorithm has been tested on different mathematical test functions and on a typical EM design problem, a planar array synthesis to show its performance.

I. INTRODUCTION

The solution of real electrical and electronic complex design problems with strict constraints often require to exploit the performances of population based global optimizers, such as the classic Genetic Algorithms (GA) or Particle Swarm Optimization (PSO) and other more recent developed techniques such as MetaPSO, GSO, MetaLamarckians and so on. One of the main drawbacks of the adoption of these optimizers in the design of a real-life devices is the difficulty in the introduction of all the heuristic knowledge already available in the field, and this is usually not trivial since this specific knowledge in some way represents often the core of the design team value and this is the main limit in the spread of population based algorithms for complex system optimization.

In order to overcome this problem, Bayesian optimization algorithms, classified as estimation of distribution algorithm, can be very effective, since they are based on the definition of the distribution of promising solutions by using the information extracted from the entire set of already good configurations. Bayesian algorithms have been suitably developed exactly for overcoming this problem, since they are based on the inductive logical process, instead of the usual “deductive” one, aiming to exploit all the heuristic knowledge that has been achieved on a particular problem.

Unfortunately, their straightforward implementations usually lack of exploration feature since they attempt to build up a probabilistic model from the available knowledge, and may be easily trapped in local maxima lacking of what is usually called as high level design prospective.

In order to overcome even this drawback and to develop a Bayesian optimization algorithm with both the required exploitation of the heuristic knowledge, and the exploration feature to avoid local maxima, in this paper a Modified BOA is proposed by adding a suitable mutation scheme, a typical GA operator, to the traditional one in order to ensure the effectiveness and the convergence of the algorithm. The new algorithm in the following will be presented, discussed and tested both on purely mathematical and on EM design problem, namely a planar array synthesis.

II. MODIFIED BAYESIAN OPTIMIZATION ALGORITHM

BOA, first published in [1], uses Bayesian Networks (BN) to estimate the joint distribution of promising solutions. In the BOA algorithm, the underlying interactions between parameters (variables) are considered to build up a probabilistic model, i.e. a BN, and to make this all the prior available information is introduced together with the set of promising solutions in order to estimate a suitable distribution. This estimation is then used to generate new candidates by sampling this distribution itself. A full description of the method can be found in [1], [2], but for sake of clarity and notation uniformity it is briefly summarized in the following.

The BOA starts by randomly generating a set of population i.e. a set of strings. The current population is evaluated and the best solutions are selected using a particular selection method (e.g. truncation selection or tournament selection). A Bayesian Network is then built to fit the selected set of strings. In the building network process, a suitable metric and a search method are used to measure and maximize the quality of the network. New offspring is generated by sampling the distribution of best population modelled by BN. Finally, the new population is added to the original one by replacing the worst performing population. The next iteration proceeds again until stopping criteria are satisfied. The pseudo-code of the BOA follows:

- 1) Randomly generate initial population (P);
- 2) Select a set of promising solutions (S) from (P);

- 3) Construct the network (B) using the chosen metric and constraints;
- 4) Generate a set of new offspring (O) according to the joint distribution encoded by B;
- 5) Create a new population (P) by replacing some (P) elements with suitably chosen (O) ones;
- 6) If the termination criteria are not met, go to (2).

In the BOA algorithm, the creation of Bayesian Networks and offspring generation are the two most important steps. Thus in the next subsections we will focus our attention on their peculiarities.

A. Bayesian Networks

Bayesian Networks BNs [3] are graphical models that combine probabilistic theory with graph theory to encode the relationship between the variables characterizing the modelled data. BNs are described by their structures i.e. directed acyclic graph and corresponding parameters. In the graph, each node represents one variable of modelled data, and the edges between these nodes correspond to conditional dependencies. The parameters are represented by the conditional probabilities for each variable given any instance of variables that this variable depends on. Mathematically, a BN encodes the joint probabilistic distribution:

$$\prod_{i=1}^l p(X_i | \pi_i) \quad (1)$$

Where $X = (X_1, X_2, \dots, X_l)$ is the vector of variables of the problem, π_i is the set of parents of the variable X_i , and $(X_i | \pi_i)$ is the conditional probability of X_i conditioned on the variables. In BOA, both the structure and the parameters of BN are trained in order to best fit the promising solution. There are two basic components of the algorithm for training a BN: the scoring metric and the search procedure. The scoring metric quantifies the quality of the given network, i.e. it is a sort of fitness function. The already available prior knowledge on the considered 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 its metric score. The exploration is usually restricted by the problem constraints, i.e. the maximum number of incoming edge to one node. This number directly influences the complexity of the algorithm in building up the network and generating the offspring. In this work, we use K2 as scoring metric [4] and greedy algorithm as search procedure. The next section will explain how to generate new offspring using the distribution extracted from the chosen network.

B. Generating offspring

Once the structure and the parameters of BN have been trained, new offspring will be generated by sampling the trained network. This procedure is in two steps: first ordering the nodes and then sampling the solution space according to the order.

The purpose of the node ordering is to generate a suitable sequence of variables so that the value of parents of each

node will be generated before the generation of the node itself. Following the ancestral ordering, given values of parents of a variable, the distribution of each variable is computed by the corresponding conditional probability, and new value is generated according to this distribution.

C. Modified BOA

As described above, the BOA performance greatly depends on the distribution of the current good solutions. However, the initial population for the BOA is randomly generated, and this means that there would be some cases in which this initial sampling of the solution space would not provide a good enough distribution of the considered problem, thus the algorithm hardly converges. To overcome this difficulty, one possibility is to increase the population size; therefore it will increase the quality of the distribution of the good solution, but this solution will be extremely time consuming. In this paper, we propose a new approach developed by adding a suitable mutation operator to the traditional BOA. By introducing a mutation operator, some individuals will be used to explore the space out of the distribution of the supposed good ones and therefore, the algorithm will be less easily trapped at local optimum with respect to the traditional implementation. This is similar to Genetic Algorithms (GA) [5], Population Based Incremental Learning (PBIL) [6], and Compact Genetic Algorithm (CGA) [7], which use mutation as one of most important operator for the solution space. However, in the proposed algorithm, we use a Bayesian Network to represent the probability model and to generate the new offspring. Moreover, we worked with a variable vector, which is much more effective for real-variable problems, as for microwave components design, compared with the use of a probability vector as PBIL and CGA. In our work, we implemented the Modified BOA introducing tournament selection and individual mutation for the optimization of a planar array.

III. NUMERICAL RESULTS

In this paper, we present result of modified BOA for Onemax problem as test function and then we apply for planar array design. For all tested problems, 30 independent runs are performed and the results showed here are mean values.

A. Onemax problem

Onemax problem is defined as sum of bits in the input binary string. The optimum of Onemax is the string of all ones. Figure 1 reports the number of fitness valuations until MBOA finds the optimum solutions. The size of the problems ranges from 100 to 500 bits. The number of fitness evaluation can be approximated by $O(n \log(n))$. The result indicates that the Modified BOA can solve Onemax with an almost linear number of evaluations.

B. Planar array design

The described algorithm has been applied to the optimization of the array factor of a planar array, as shown in [8]. The

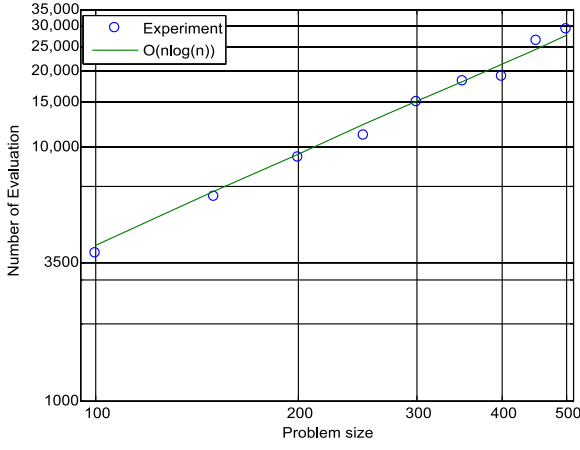


Fig. 1. Number of fitness evaluation of modified BOA for Onemax problem.

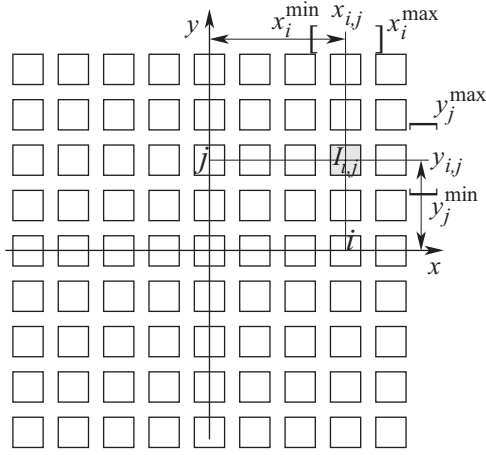


Fig. 2. Planar array geometry (from [8]).

considered geometry consists of $(2N_e + 1) \times (2N_e + 1)$ identical elements whose position and excitation (both amplitude and phase) are free to vary.

The basic geometry is depicted in Fig. 2. Each element is characterized by its excitation $I_{i,j}$ and position $(x_{i,j}, y_{i,j})$, with $i = -N_e, \dots, N_e$ and $j = -N_e, \dots, N_e$. Without loosing in generality, for what concerns the choice of the array specifics, a beam in the broad side direction (z) has been considered.

The array is considered symmetrical both on x and y , therefore $I_{i,j} = I_{-i,j} = I_{i,-j} = I_{-i,-j}$, $x_{i,j} = -x_{-i,j} = -x_{i,-j} = x_{-i,-j}$, and $y_{i,j} = y_{-i,j} = -y_{i,-j} = -y_{-i,-j}$. To further reduce the number of degrees of freedom the following hypotheses are taken:

$$I_{i,j} = \hat{I}_i \times \hat{I}_j \quad (2)$$

$$y_j = x_i \quad (3)$$

with the additional constraints: $\hat{I}_i = \hat{I}_j$, $\hat{I}_i = a_i e^{j\beta_i}$. The amplitudes a_i are let free to vary in the $[0, 1]$ range and the phases β_i in the $[-\pi, \pi]$ range. These constraints effectively re-convert the planar array to a linear array of linear arrays.

For what concerns position optimization constraints, we

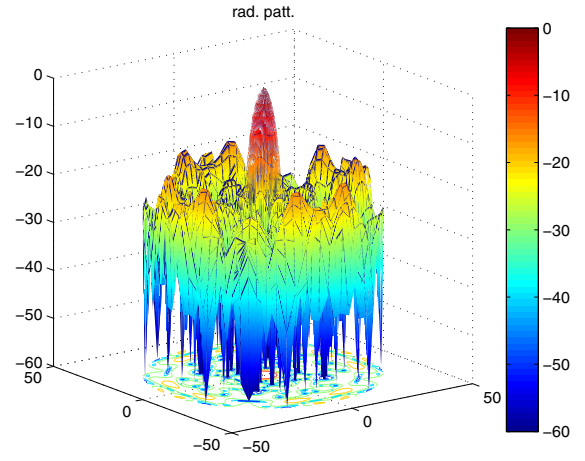


Fig. 3. Resulting radiation pattern

chose $x_i \in [x_i^{\min}, x_i^{\max}]$ with:

$$x_i^{\min} = i \cdot \lambda / 2 \quad (4)$$

$$x_i^{\max} = i \cdot \lambda \quad (5)$$

The central element is taken as a reference and is characterized by $x_0 = 0$, $a_0 = 1$ and $\beta_0 = 0$.

The aim of the optimization is to design a 9×9 BS array ($N_e = 4$) with a $\theta_{3dB}/2 = 3.8^\circ$ and a side lobe level (SLL) envelope below -20 dB for $\theta > 9^\circ$. This is of course a multi-objective problem and the cost function has been defined to take into account both objectives by considering 90 sampling points in the far-field pattern, scattered over different ϕ -cuts. Thanks to array symmetry only two cuts may suffice, at $\phi = 0$ and $\phi = 45^\circ$. Points are taken in $\theta \in [0, 90^\circ]$ so that the values of $\sin(\theta)$ are equally spaced (see [9] for details). The objective function to be minimized is defined as the sum of the magnitude of the far field radiation pattern exceeding the prescribed side-lobe envelope (-20 dB). This penalizes side-lobes above the envelope, while neither penalty nor reward is given for side-lobes below the specification. On the main beam, on the other hand, the threshold is placed at -3 dB and a penalization is assigned if sampled point goes below this limit within the prescribed beam-width. This kind of constrain is of course non-linear but evolutionary approaches are well known to be very well-suited for nonlinear objective functions.

Fig. 3 shows the array factor of this optimized configuration in both $\phi = 0^\circ$ and $\phi = 45^\circ$ planes. The array factor in the $\phi = 90^\circ$ plane is identical to the one in the $\phi = 0^\circ$ plane due to symmetry reasons.

The preliminary results encourage us to further investigate the effectiveness of the here proposed algorithm. Further results on MBOA applications of EM test problems and mathematical test functions will be presented in the full paper.

REFERENCES

- [1] M. Pelikan, D. E. Goldberg, and E. Cantu-Paz, "Boa: The bayesian optimization algorithm," in *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO 1999)*, 13-17 July 1999, Orlando, Florida, USA. Morgan Kaufmann, 1999, pp. 525-532.

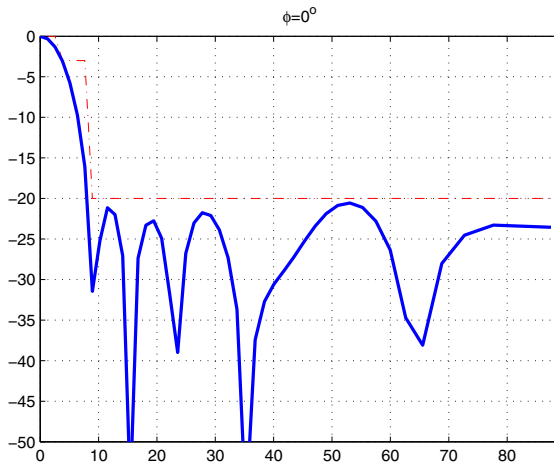


Fig. 4. Resulting radiation pattern

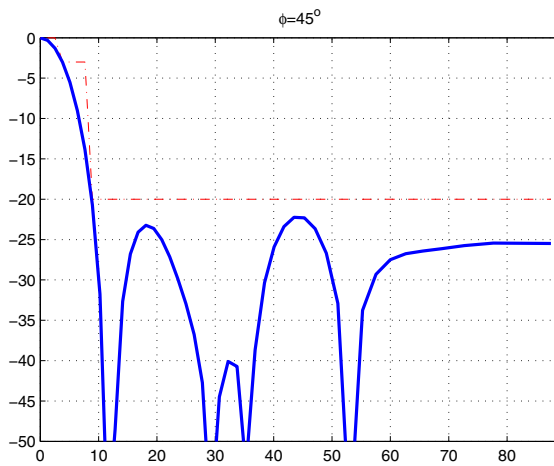


Fig. 5. Resulting radiation pattern

- [2] M. Pelikan, *Hierarchical Bayesian optimization algorithm: toward a new generation of evolutionary algorithms*, ser. Studies in fuzziness and soft computing. Springer-Verlag, 2005.
- [3] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*, ser. The Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann Publishers, 1988.
- [4] D. Heckerman, D. Geiger, and D. Chickering, "Learning bayesian networks: The combination of knowledge and statistical data," *Machine learning*, vol. 20, no. 3, pp. 197–243, 1995.
- [5] D. Goldberg, *Genetic algorithms in search, optimization, and machine learning*, ser. Artificial Intelligence. Addison-Wesley Pub. Co., 1989.
- [6] S. Baluja, "Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning," Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-94-163, 1994.
- [7] G. Harik, F. Lobo, and D. Goldberg, "The compact genetic algorithm," *Evolutionary Computation, IEEE Transactions on*, vol. 3, no. 4, pp. 287–297, nov 1999.
- [8] S. Selli, M. Mussetta, P. Pirinoli, R. Zich, and L. Matekovits, "Differentiated meta-pso methods for array optimization," *Antennas and Propagation, IEEE Transactions on*, vol. 56, no. 1, pp. 67–75, jan. 2008.
- [9] D. Boeringer and D. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *Antennas and Propagation, IEEE Transactions on*, vol. 52, no. 3, pp. 771–779, march 2004.