Antenna Optimization Using Particle Swarm Optimization Algorithm

Ružica M. Golubović and Dragan I. Olčan

Abstract—We present the results for two different antenna optimization problems that are found using the Particle Swarm Optimization (PSO) algorithm. The first problem is finding the maximal forward gain of a Yagi antenna. The second problem is finding the optimal feeding of a broadside antenna array. The optimization problems have 6 and 20 optimization variables, respectively. The preferred values of the parameters of the PSO algorithm are found for presented problems. The results show that the preferred parameters of PSO are somewhat different for optimization problems with different number of dimensions of the optimization space. The results that are found using the PSO algorithm are compared with the results that are found using other optimization algorithms, in order to estimate the efficiency of the PSO.

Index Terms—EM Optimization, Particle Swarms Optimization, Yagi Antenna, Broadside Antenna Array.

I. INTRODUCTION

Along with the development of the algorithms for EM simulations, optimization algorithms have been developed, too. In the process of a design of EM structure, an EM simulator is run for many times, in order to obtain the structure with the specified characteristics. For that reason, engineers need efficient optimization algorithms, in terms of the total number of EM solver calls.

The Particle Swarm Optimization (PSO) is a relatively new approach to EM optimization and design [1-3]. It is based on the analogy of movement of bird flocks or fish schools, on one side, and the optimization, on the other side. The PSO algorithm belongs to the class of the heuristic optimization algorithms. The possibilities and the limitations of its applications to EM problems are not fully explored yet. The aim of this paper is to summarize the results for the found preferred parameters of the PSO when it is applied to the optimization of EM systems and to compare these results with the ones obtained using other optimization algorithms. For all antenna simulations WIPL-D Pro v6.1 software package is used [4].

II. PSO ALGORITHM

The PSO algorithm searches for the global minimum of the cost-function, i.e., minimizes the cost-function of an EM problem by simulating movement and interaction of particles (agents) in a swarm. The position of a particle corresponds to one possible solution of the EM problem, i.e., it corresponds to one point in the optimization space. Since we assume that there is no a priori knowledge of the optimization problem, there is equal possibility for any point in the optimization space to be selected for the beginning of the optimization. Hence, PSO starts with randomly chosen positions of particles. Each agent has information about the best position found by itself, $p_{best}$ (which is the position vector in the optimization space), and the best position found by whole swarm, $g_{best}$. Each particle keeps track of its personal best position found in the optimization space, $p_{best}$. The swarm keeps track of the global best position, $g_{best}$. The velocity vector for the calculation of the particle position in the next iteration is calculated as:

$$v_n = w \cdot v_{n-1} + c_1 \cdot \text{rand} \cdot (p_{best} - x_n) + c_2 \cdot \text{rand} \cdot (g_{best} - x_n)$$

where $v_{n-1}$ is the particle velocity in the previous iteration, $w$ is the inertia coefficient, rand() is the function that generates uniformly distributed random numbers in the interval from 0.0 to 1.0, $c_1$ is the cognitive coefficient (it controls the pull to the personal best position), and $c_2$ the social rate coefficient (it controls the pull to the global best position).

The next position of the particle in the optimization space is calculated as:

$$x_n = x_{n-1} + v_n \Delta t$$

where $\Delta t$ is most often considered to be of a unit value.

It is found that particles might fly-out from the given optimization space if there are no limits for the velocity. Therefore, the maximal velocity $V_{max}$ is introduced as another parameter of the PSO algorithm. The maximal velocity, $V_{max}$, represents the maximal percentage of the dynamic range of each optimization variable for which the velocity can change in successive movements of the particle. In out implementation of the PSO algorithm, all dynamic ranges of the optimization variables are scaled to the interval $[-1.0, 1.0]$, and one unique value for the maximal velocity for all optimization variables is used.

Default parameters of PSO found in the literature [3] are: the number of particles $p = 15$, the inertia coefficient $w = 0.729$, the maximal velocity $V_{max} = 0.2$, the cognitive...
coefficient and the social rate coefficient \((c_1, c_2) = (1.494, 1.494)\).

### III. OPTIMIZATION #1: MAXIMAL FORWARD GAIN OF YAGI ANTENNA

The first optimization problem is a Yagi antenna with a driving element, a reflector, and 10 directors. The aim of the optimization is to obtain the maximal average forward gain in the frequency range 295-305 MHz. There are six optimization parameters: the length of the reflector \(l_{\text{ref}}\), the length of the driving element \(l_{\text{drv}}\), the length of directors \(l_{\text{dir}}\) (all directors are the same), the distance between the reflector and the driving element \(d_{\text{ref}}\), the distance between the driving element and the first director \(d_{\text{drv}}\), and the distance between two neighboring directors \(d_{\text{dir}}\) (the distance between each pair of neighboring directors is the same). All optimization parameters are varied in the same range \(0.8 \leq l_{\text{ref}}, l_{\text{drv}}, l_{\text{dir}}, d_{\text{ref}}, d_{\text{drv}}, d_{\text{dir}} \leq 2.0\) m. Therefore, the optimization space has six dimensions.

The cost-function is calculated as

\[
 f_{\text{cost}} = \frac{1}{5} \sum_{k=1}^{5} (G_k - 20[\text{dB}])^2 \tag{3}
\]

where \(G_k\) is gain in dB at five equidistant frequencies.

PSO algorithm parameters are varied one-at-a-time with all the others equal to the default values as given in Section 2. The PSO coefficients are taken from the following sets:

\(p \in \{5, 10, 15, 20, 25, 30, 35\}\),

\(w \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}\),

\(V_{\text{max}} \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}\), and

\((c_1, c_2) \in \{(0, 0.0), (0.5, 2.5), (1.0, 2.0), (1.5, 1.5), (2.0, 1.0), (2.5, 0.5), (3.0, 0.0)\}\).

This optimization problem has one global minimum and relatively high number of local minima. It is found with the numerical experiments that hybrid combination of GA and Nelder-Mead simplex, and algorithm based on estimation of local minima (termed ELM) outperformed virtually all other optimizers for this example [5]. For that reason, the outcome of the PSO is compared to the outcomes of these algorithms.

In the case of GA: the population consists of 60 solutions, the best 20 solutions forms the next generation, crossover probability is 0.8, mutation probability is 0.1, binary coding of optimization parameters with 16 bits per variable is used.
and the total number of generations is 5. Hence, the total number of iterations for the GA is 300. The best-found solution with GA is used as the starting point for the simplex. The parameters of Nelder-Mead simplex algorithm are taken from [6]. The algorithm finishes if: a) the absolute changes of the cost-function in all points of the simplex are less than $10^{-3}$, b) the relative difference per all coordinates of all points of the simplex are less than $5 \cdot 10^{-4}$% of the dynamic range, and c) if the total of 300 iterations is reached. Therefore, the total number for the hybrid combination of GA and simplex is 600.

In the case of Estimation of local minima (ELM) algorithm, initial random set consists of 100 solutions, the 10 closest points (for each point from the random set) are used for estimation if the point is (close to) a local minimum. Each estimated position of a local minimum is then used as a starting point for the simplex algorithm. The setup for the simplex is the same as in the hybrid combination of GA and simplex. The total number of iterations for ELM varies depending on the number of estimated local minima, but it is found to be 700 on average.

All algorithms are repeated for 20 times. The averaged best-found solution versus the number of iteration is shown in Fig. 5. The first 600 iterations are shown. It can be seen that after 400 iterations all algorithms converge to practically the same solution. However, the PSO has somewhat faster convergence than other two optimizers that is observed for the first 200 iterations.

IV. OPTIMIZATION #2: OPTIMAL EXCITATIONS OF BROADSIDE ANTENNA ARRAY

The second optimization problem is a broadside antenna array that consists of 42-point sources located along the $z$-axis and at the equal distance of one half of a wavelength at the operating frequency. Optimization of excitations’ magnitudes is done for the lowest possible side-lobe levels. The solution has to have symmetry of the excitations’ amplitudes, and therefore only one half of the excitations are varied. To avoid infinite number of solutions due to scaling, the 21st coefficient is predefined. Hence, we have a 20-dimensional optimization problem.

The criterion for the optimization is that the side lobe levels should be lower then $-80$ dB everywhere for $\theta < 65^\circ$, where $\theta$ is the angle between the array axis and the radiation direction. The cost-function is calculated as:

$$f_{\text{cost}} = \frac{1}{n} \sum_{i=0}^{n-1} \left[ \max \left( 0, 80 - F_{\text{max}}(\theta_i) \right) \right]^2$$

where $F_{\text{max}}$ is in the direction $\theta = 90^\circ$, $\theta_i = \frac{\pi}{n}$, $n = 65$, and $F(\theta_i)$ is given in dB. The theoretical result exists in the form of the binomial distribution of the amplitudes. The ratio of the highest and the lowest amplitude is of the order of $10^{11}$, which is inconvenient from the standpoint of the numerical optimization. Hence, we represent each coefficient as $s_k = \ln(a_k)$, $k = 1,2,...,20$ so that the maximal ratio of the coefficients is less than 30. Each optimized parameter, $s_k$, has the lower bound equal to zero and the upper bound equal to the largest (21st) coefficient.

PSO algorithm parameters are varied in the same way as in the previous example, with the difference that the default value for the number of particles in the swarm is 15. The same sets of the parameters’ values are used.

One optimization lasts for 15000 iterations (EM solver calls). Each optimization is repeated for 100 times to get the good estimation of the average outcome of the optimization. The average minimal cost-function found after the number of iterations is shown in Figs. 6-9.

From the presented results it can be seen that preferred values for the PSO algorithm in this problem are: the number of particles $p = 30$, the inertia coefficient $w = 0.7$, the maximal velocity $V_{\text{max}} = 0.7$, and cognitive and social rate coefficients $c_1, c_2 = (1.0, 2.0)$. Note that all values for $c_1, c_2$ coefficients except $(c_1, c_2) = (0.0, 3.0)$ show very similar behavior.
This optimization problem has the large number of optimization parameters and a huge number of minima of different depth. It is found, with additional numerical experiments, that the binary GA outperforms all other algorithms for this problem [5]. Therefore, outcome of the PSO is compared to the outcome of the binary GA, in order to establish the efficiency of the PSO algorithm.

The parameters for PSO algorithm are chosen to be the ones that outperform all others for this problem.

In the case of GA: the population consists of 300 solutions, the best 100 solutions forms the next generation, crossover probability is chosen to be 0.8, mutation probability is 0.1, binary coding of optimization parameters with 16 bits per variable is used, and the total number of generations is 100. The total number of iterations for both GA and PSO algorithms is 30 000. The averaged best-solution versus the number of iterations is shown in Fig. 10. Due to presented results, GA outperforms PSO, but the PSO still finds very good solutions.

V. CONCLUSION

On the basis of results obtained by numerical experiments, presented in the paper, it is concluded that preferred PSO parameters change with the increase of the number of optimization variables. The number of particles in the swarm seems to be proportional to the number of dimensions of the optimization problem. For the other parameters of the PSO algorithm, the default values performed very well, but careful tuning can yield slightly more efficient optimization. Due to results presented in this paper, we also conclude that the PSO can be efficiently used for optimization of antenna, although it was slightly outperformed by other algorithms (combination of simplex and GA). Having in mind the relative simplicity of the PSO algorithm and its suitability for the implementation on the parallel processors, it seems to be really good approach for optimization problems.

REFERENCES