Black-Hole PSO for Antennas Optimization

M. Ruello[1], *C.A. Gonano*[1], *M. Mussetta*[1], *P. Pirinoli*[2], *R.E. Zich*[*1]

[1]Politecnico di Milano, Dipartimento di Energia
Via La Masa 34, 20156 Milano, Italy
marco.mussetta@polimi.it, riccardo.zich@polimi.it

[2]Politecnico di Torino, Dipartimento di Elettronica
Corso Duca degli Abruzzi 24, 10129 Torino, Italy

Abstract

In the past years Particle Swarm Optimization (PSO) has gained increasing attention for engineering and real-world applications. Among these, the design of antennas and electromagnetic devices is a well-established field of application. Black-hole PSO (bhPSO) is a novel version of PSO, which is here considered for antennas optimization. It is based on the concept of repulsion among particles when they get stuck in local optima. The design of a planar array is here addressed in order to assess its performances on a benchmark EM optimization problem. Reported results show its effectiveness in dealing with antenna optimization.

1. Introduction

The optimization process of antenna is generally not trivial since the interaction of many parameters, complex boundary conditions, and width/peak gain relation [1]. Among these the design of antennas and micro to sub-millimetre wave components is a potential field of application.

Optimization problems with more than one, often conflicting, objective function are quite common. In these cases there is no a single solution, but the real challenge is to find a good trade-off solution that represents the best compromise among the considered objectives. To address this complexity, the use of Evolutionary Optimization algorithms it is now quite common also in the Antennas community [6], where the most known are Genetic Algorithms (GA) and Particle Swarm Optimization (PSO).

In this paper a variation of the well-known PSO, namely the Black-hole PSO, is presented and its performances are assessed over classical benchmark functions and antennas problems.

2. Basic PSO Algorithm

Particle Swarm Optimization (PSO) is a well known evolutionary algorithm based on a model of social interaction between independent agents (particles) that uses social knowledge (also called swarm intelligence, *i.e.* the experience accumulated during the evolution) in order to find the global maximum or minimum of a function [2]. This computational technique, adopts a pseudo-biological approach and takes its origin from the simulation of social behaviors such as those related to synchronous bird flocking and fish schooling; it is similar to other population-based algorithms, like GAs, but it operates emulating social interaction between independent agents and utilizes swarm intelligence to achieve the goal of optimizing a specific fitness function in a way easy to understand and implement [3]. In fact, any set of coordinates in the M-dimensional space is a particular position of an agent and represents a solution; it corresponds to a particular value of the fitness function. Each particle also has an associated velocity, that takes into account the best position reached by all ones and the best position, resulting in a migration of the swarm towards the global optimum.

The standard PSO algorithm is an iterative procedure in which a set of $i = 1, \dots, N_p$ particles, or agents, are characterized by their position $\vec{X}_i$ and velocity $\vec{V}_i$, defined in the M-dimensional space domain of a cost function $F(\vec{X})$.

At the beginning positions and velocities have completely random values $\vec{X}_i^0$ and $\vec{V}_i^0$, then they are updated iteratively according to the rules:

$$\vec{V}_i^{k+1} = \omega_k \vec{V}_i^k + \phi\eta_1(\vec{P}_i - \vec{X}_i^k) + \phi\eta_2(\vec{G} - \vec{X}_i^k) \tag{1}$$

$$\vec{X}_i^{(k+1)} = \vec{X}_i^{(k)} + \vec{V}_i^{(k+1)} \tag{2}$$

where $\vec{P}_i$ the best position so far attained by particle i itself (personal knowledge) and $\vec{G}$ is the best position so far attained by the whole swarm (social knowledge); ω_k is a friction factor slowing down particles (it can depend on k, as

978-1-4673-5225-3/14/$31.00 ©2014 IEEE

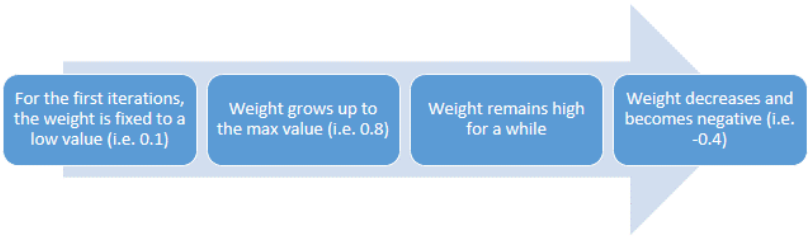

Figure 1: A view of the black-hole operator here proposed and implemented.

shown in [4]), η_1 and η_2 are positive parameters tuning the pulls towards the personal and global best positions and ϕ is a random number of uniform distribution in the $[0, 1]$ range.

3. Black-hole PSO

As previously reported [1], standard PSO usually gets stuck because exploitation outperform exploration and, when a local best is found, there is a high probability of bouncing around this point. In fact the agent that found this point has two out of three vectors pointing the same point, which is, indeed, its own personal best ($\vec{P}_i$) and global best ($\vec{G}$).

In order to improve the algorithm's exploration capability, black-hole concept is here introduced; its main ideas (briefly summarized in Figure 1) are the following:

- Around the global best a suitable hyper-space can be defined (radius ρ);
- Every bees that stop in this space dies and is born randomly again.

Moreover, the introduced variation affects the usual velocity update rule: which is updated as:

$$\vec{V}_i^{k+1} = \omega_k \vec{V}_i^k + w_p(\vec{P}_i - \vec{X}_i^k) + w_g(\vec{G} - \vec{X}_i^k) \tag{3}$$

where $\vec{P}$ is the position of the best value found by the single particle, w_p is its weighting coefficient, $\vec{G}$ is the position of the best value found by all agents, w_g is its weight. It is worth noticing that, since random search (*i.e.* exploration) is guaranteed by the new Black-Hole operator, having other random coefficient ϕ is useless.

In particular, in our black-hole implementation, a variable weight system can be introduced: since global best stagnation is avoided by the black hole it is better to use an high value for w_g, but this is not good at an early stage of search. In fact only "stable" best should attract the particles. So it is good having a low w_g value in the beginning and then making it increase with iterations. After a certain number of iterations in which the global best is stable, we can assume that all the information linked to that point has been used, so we should explore other parts of the domain, similarly to the concept of *implicit restart* introduced firstly in [5]. To help this, the global best weight starts lowering and becomes negative, thus rejecting other particles from its region. Every time a new global best is found the cycle starts over.

Of course, a new randomly generated position can easily be slightly worst than the best one found so far; nevertheless, it could contain more useful information than the current global best, thus we should give a chance to these new points to be considered. This is provided by increasing (or decreasing if we are founding a maximum) the value of the stored global best cost by a small percentage during the last part of the procedure shown in Figure 1.

In the following figures, a preliminary analysis of the performances of bhPSO compared to traditional PSO is presented: in particular, the considered functions are: Ackley, Rastrigin (Figure 2) and Rosenbrok (Figure 3), with a $M = 10$ dimension space and 50 individuals.

In the final paper a detailed parametric analysis of bhPSO parameters will be presented to further show the fine tuning of the algorithm operators.

4. Planar array design

In this paper, the proposed algorithm has been used for the optimization of the array factor of a planar array. The 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.

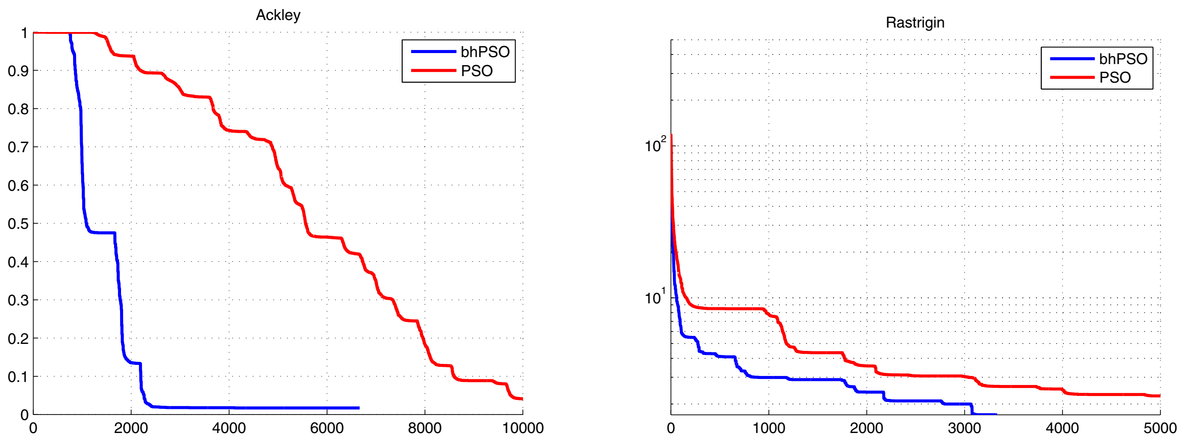

Figure 2: bhPSO and PSO performances over the Ackley (left) and Rastrigin (right) benchmark function (Average over 20 independent trials; $M = 10$).

The aim of the optimization is to design a 9×9 BS array ($N_e = 4$) with a $\theta_{3dB}/2 = 3.8°$ and a side lobe level (SLL) envelope below -20 dB for $\theta > 9°$. 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. As shown in [1], the design can be done analytically by resorting to the Tseng-Cheng formulas [8], constraints can be met by a 12×12 array (144 elements) with a 0.6λ element spacing.

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 (-20dB). 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 $-3dB$ 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.

Taking advantage of the array symmetries the number of free parameters is reduced to 12, i.e. the position and the amplitude and the phase of the excitation of four array elements.

5. Numerical Results and Conclusion

Dealing with the planar array optimization, Figure 4 shows the array factor of an optimized configuration in $\phi = 0°$ plane. The array factor in the $\phi = 90°$ plane is identical to the one in the $\phi = 0°$ plane due to symmetry reasons.

It is worth noticing that trying to achieve this array pattern with only 9×9 elements leads to a total number of elements equal to 81 instead than 144 (equivalent Tseng-Cheng array), with a net reduction of elements to be build and fed equal to 43% and a corresponding significant reduction in array and feeding network complexity. The preliminary reported results confirm the ability of the algorithm to address complex EM optimization problems.

References

[1] S. Selleri, M. Mussetta, P. Pirinoli, R.E. Zich, L. Matekovits, "Differentiated meta-PSO methods for array optimization", *IEEE Transactions on Antennas and Propagation*, Vol. 56, No. 1, 2008, pp. 67–75.

[2] J. Kennedy, R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann: San Francisco, CA, 2001.

[3] R.C. Eberhart, Y. Shi, "Comparison between genetic algorithms and particle swarm optimization", *Proc. of 7th Annual Conf. Evol. Program*, pp. 611–616, Mar. 1998.

[4] R.C. Eberhart, Y. Shi, "Particle swarm optimisation: developments, applications and resources", *Proc. of Congress on Evolutionary Computation*, pp. 81-86, 2001.

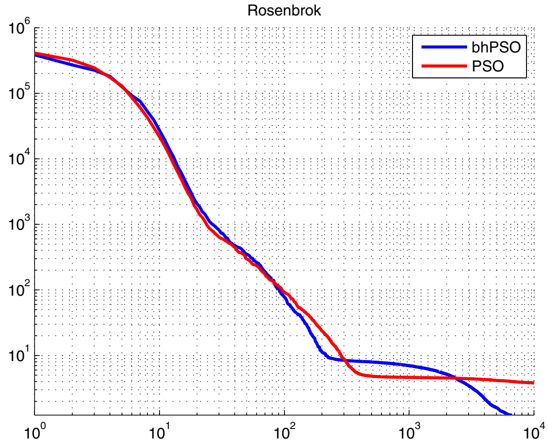

Figure 3: bhPSO and PSO performances over the Rosenbrok benchmark function (Average over 20 independent trials; $M = 10$).

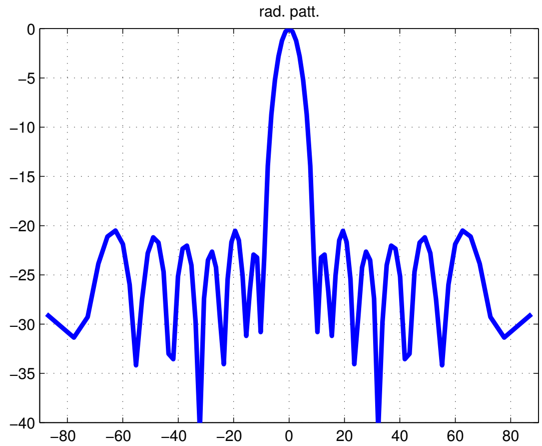

Figure 4: Resulting radiation pattern

[5] L. Matekovits, M. Mussetta, P. Pirinoli, S. Selleri, R.E. Zich, "Particle swarm optimization of microwave microstrip filters", *2004 IEEE AP-S Symposium Digests*, Monterey (CA), 20–26 June 2004.

[6] M. Mussetta, F. Grimaccia, R.E. Zich, "Comparison of different optimization techniques in the design of electromagnetic devices", *2012 IEEE Congress on Evolutionary Computation (CEC)*, Brisbane, QLD, June 2012.

[7] D. Simon, "Biogeography-Based Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 12, No. 6, pp. 702–713, December 2008.

[8] F.I. Tseng, D.K. Cheng, "Optimum scannable planar arrays with an invariant side lobe level," *IEEE Proc.*, Vol. 56, Nov. 1968, pp. 1771–1778.