Particle Swarm Optimization for the Design of Frequency Selective Surfaces

Simone Genovesi, *Member, IEEE*, Raj Mittra, *Life Fellow, IEEE*, Agostino Monorchio, *Senior Member, IEEE*, Giuliano Manara, *Fellow, IEEE*

***Abstract*— The particle swarm optimization (PSO) is a stochastic strategy that has recently found application to electromagnetic optimization problems. It is based on the behaviour of insect swarms and exploits the solution space by taking into account the experience of the single particle as well as that of the entire swarm. This combined and synergic use of information yields a promising tool for solving design problems that require the optimization of a relatively large number of parameters. In this paper, the problem of synthesizing Frequency Selective Surfaces (FSSs) is addressed by using a specifically derived particle swarm optimization procedure, which is able to handle, simultaneously, both real and binary parameters. Representative numerical examples are presented to demonstrate the effectiveness of the method. Finally, the performance of the PSO is compared with that of the genetic algorithm.**

***Index Terms*—Particle swarm, Frequency Selective Surfaces (FSSs), Optimization.**

I. Introduction

THE particle swarm optimization (PSO) is an evolutionary computation technique inspired by social behavior of a flock of birds and insect swarms. Originally proposed by Kennedy and Eberhart [1], it has recently been introduced to the electromagnetic community [2] and has found useful applications in antenna synthesis [3-4].

In PSO, each solution is represented by an "agent" which traverses in a multidimensional space. Each dimension of this space represents a parameter of the problem to be optimized. During the search procedure that seeks the best location in the solution domain, the agents change their position with time as they fly around in a multi-dimensional solution space. During their flight, particles adjust their velocity according to their own experience and those of the other particles, making use of the best position reached, both by themselves and by the rest of the swarm. As a result, the agent is stochastically attracted both toward the best personal as well as toward the best overall location, while evaluating the goodness of its current location in the process.

In this paper, a customized particle swarm optimizer is presented for the design of frequency selective surfaces (FSSs). This problem has been previously addressed by the authors [5] via the Genetic Algorithms (GA).

We have found that the use of the PSO is promising from several points of view. First of all, the PSO requires only a single operation for each iteration *viz.*, the velocity update, instead of three needed in the GA (cross-over, mutation and selection). In addition the number of parameters that has to be set is less than that of its GA counterpart. The PSO has to be initialized with the number of agents in the swarm and the three parameters involved in the velocity updating are: inertial weight, social and cognitive rate. In contrast to the PSO, the GA requires the user to make a critical choice of the optimal dimensions of population, cross-over, mutation rate, as well as of selection and cross over strategies such as elitism. It is important to point out that in contrast to the binary GA, the PSO does not require a conversion from the binary to real values of the parameters involved in the optimization process. It is also useful to note that the search direction is intrinsically multiple for the PSO -each agent has its own search path-depending upon the history of its past explorations. In the case of GA, the genetic pool seldom from the path traced by the best chromosome.

In the following section we describe the PSO algorithm followed in this paper. We pay particular attention to the simultaneous presence of real and binary parameters in the optimization process, and how this affects dynamics of the swarm. We follow this by presenting some results to illustrate the reliability and efficiency of the proposed algorithm.

II. The Particle Swarm Algorithm

In the PSO algorithm, each particle of the swarm flies in an n-dimension space, and the position at a certain instant i is identified by the vector of the coordinates X

$$X(i) = \left(x_1(i), x_2(i), ..., x_n(i)\right) \quad (1)$$

Each $x_n(i)$ component represents a parameter of our physical problem that has to be optimized. At the beginning of the process, each particle is randomly located at a position, and moves with a random velocity, both in direction and magnitude. The particle is free to fly inside the n-dimensional

Manuscript received ****.

S. Genovesi is with Department of Information Engineering, University of Pisa, Pisa, Italy and with ISTI, Italian National Research Council, Pisa, Italy (e-mail: simone.genovesi@iet.unipi.it).

R. Mittra is with the Electromagnetic Communication Laboratories, 319 Electrical Engineering East, Pennsylvania State University, University Park, PA 16802 (mittra@engr.psu.edu).

A. Monorchio and G. Manara are with the Department of Information Engineering, University of Pisa, Via Caruso, I-56126 Pisa, Italy (a.monorchio@iet.unipi,it, g.manara@iet.unipi.it).

space defined by the user, within the constraints of the n boundary conditions, which limit the extent of the search space and, hence, the values of the parameters during the optimization process.

At the generic time step $i+1$, the velocity is expressed by the following equation

$$v_l(i+1) = w * v_l(i) + c_1 * \text{rand}() * \left(p_{best,l}(i) - x_l(i)\right) + c_2 * \text{rand}() * \left(g_{best,l}(i) - x_l(i)\right) \quad (2)$$

where:

- $v_l(i)$ is the velocity along the l direction at time step i;
- w is the inertial weight;
- c_1, c_2 are the cognitive and the social rate, respectively;
- $p_{best,l}(i)$ is the best position along the l direction found by the agent during its own wandering up to *i-th*;
- $g_{best,l}(i)$ is the best position along the l direction discovered by the entire swarm;
- rand() is a generator of random numbers uniformly distributed between 0 and 1.

The detailed interpretations of these step terms may be found in [6]. At each iteration step, the new velocity is the sum of the actual velocity, scaled by the factor w which represents the weight of the particle, and two terms that express the attraction due to p_{best} and g_{best}. The former term determines how much the agent is influenced by the memory of its own best (referred as the "cognitive rate") and the value of c_1 encourages the independent search for the best, regardless of the experience of the swarm. On the other hand, the latter term is related to the influence the swarm has on the particle (called the "social rate") and c_2 controls the exploitation of the actual best. The random generator introduces the proper chaotic component of a real swarm. The position of each particle is then simply updated according to the equation:

$$x_l(i+1) = x_l(i) + v_l(i) * \Delta t \quad (3)$$

where $x_l(i)$ is the current position of the agent along the direction l at the iteration # i, and Δt is the time step. The boundary conditions implemented are those for reflecting walls, which change the sign of the velocity of the particle whenever it hits the designated border.

III. Application to FSS Synthesis

The described procedure is suitable for solving optimization problems involving real parameters. However, for the case of the FSS design, we need to manage not only the real but also the binary parameters for describing the shape of the unit cell (see [5]), consequently, it is necessary to incorporate both of these features into the algorithm. A discrete binary version of the PSO was first introduced by Kennedy and Eberhart [7], in which the concept of velocity loses its physical meaning and assumes the value of a probability instead. Specifically, the position along a direction can now be either 0 or 1, and the velocity represents the probability of change for the value of that bit. In light of this, it becomes necessary to modify the expression in (2) by imposing the condition that the value of $v_l(i)$ must be in the interval [0.0, 1.0], and insisting that any values outside this interval be unacceptable. As a consequence, a function (T) is needed to map the results of (2) within the permissible range. If we let w=1 and c_1=c_2=2, $v_l(i)$ is in the interval [–4, 5]. The T function linearly compresses this dynamic range into the desired set [0, 1] and then the position is updated by using the following rule:

$$\begin{aligned} & \textit{if } \left(\text{rand}() < T(v_l(i))\right) \textit{ then} \\ & \quad x_l(i) = \overline{x}(i) \\ & \textit{else} \\ & \quad x_l(i) = x_l(i) \end{aligned} \quad (4)$$

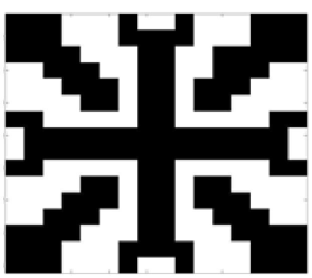
where *rand()* is the same random function adopted in (2) and the operator $\overline{x}_l$ indicates the binary negation of x_l. This implies that if the random number is less than the probability expressed by the velocity, the bit is changed. Thus, the faster the particle moves in that direction, the larger is the possibility of change.

In order to design an FSS structure, the parameters that can be optimized by the algorithm are the shape of the unit cell, its dimensions, the permittivity of dielectric layers and their thickness. The size of the multi-dimensional space in which the particle moves is variable, and it is related to the different options given to the user. In fact, the number and the kind of the parameters depend on the choices offered at the beginning of the optimization process. First of all, two real-valued parameters that can be tuned according to the imposed requirements are the dimensions of the unit cell along the main directions of periodicity. For each dielectric substrate, it is possible to choose the value of the permittivity from a predefined database, using integer parameters in this case. Consequently, the particle is only allowed to assume integer values, and a finite number of these values in the search direction. As for the thickness, it can be chosen from a database (integer parameter) or be a real value within the imposed boundary for that component. The shape of the unit cell is completely defined as a binary parameter and their number is fixed on the basis of the symmetry consideration of the unit element. The discretization adopted for this work is 16×16; hence, we have 256 binary parameters for arbitrary FSS elements. However, this number reduces to 64 and 36, for a quarter-fold or eight-fold symmetry, respectively. The analysis of the entire FSS structure has been performed by a MoM code, employing roof top basis functions [5], which proved to furnish reliable results [8]. The fitness function, employed to test the performance of the solution proposed by the PSO, is based on the mean square error between the reflection coefficient of the synthesized structure and the frequency mask which translates the requirements imposed by the user.

IV. Numerical Results

The PSO algorithm implemented on the basis of the prescription given in the last section has proved to be an efficient tool for synthesizing FSSs. It exhibits a good convergence rate even when the number of agents is small compared to the standard dimension of a GA population.

In order to demonstrate the PSO algorithm capabilities, we present the design of a FSS screen operating as a diplexer with requirements on three different bands. Specifically, it is desired that the FSS provides a transmission coefficient in the frequency range from 2.0 GHz to 3.0 GHz as high as possible, at normal incidence. Concurrently, the screen should present a high reflection coefficient in the band from 0.1 GHz to 1.0 GHz and from 4.5 GHz to 5.5 GHz (again for normal incidence). A free standing FSS has been considered for this example and the elementary cell derived by the PSO is shown in Fig. 1(a); the period of the unit cell chosen by the PSO is 3.85 cm. In Fig. 1(b), we show a complete view of the FSS screen, whereas the frequency response of the screen is shown in Fig. 2. For this example, the solution converged in 256 iterations, for a swarm comprised of 16 agents, so that a total amount of 4096 calls to the electromagnetic solver have been performed in the optimization process. In our experience, several hundreds iterations are typically needed, with a total number of calls to the solver spanning from 5 to 10 thousands, to achieve good convergence with the GA for the same problem.


(a)

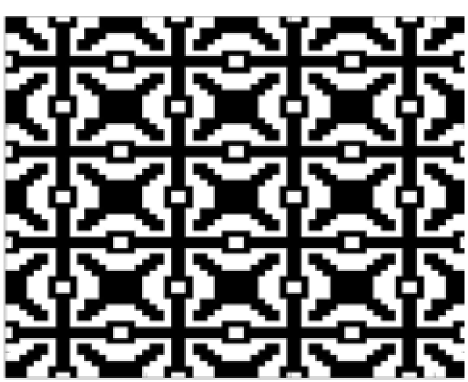

(b)

Fig.1 (a) Shape of the unit cell derived by the PSO algorithm; (b) complete view of the FSS screen (dark areas correspond to printed metallic elements).

For validation purposes, the results have been repeated by using an independent solver based on the Finite Element Method, namely Ansoft HFFS ®, shown with dots in the same Fig. 2. It is worthwhile to point out that the design presented herein is only intended to demonstrate the potential of our optimization scheme; indeed, while a similar performance of the FSS can be realized by using a conventional shape of the resonating element, the proposed procedure is more flexible, and has the ability of evolving towards unconventional elements, which, in some cases, are better suited for certain specific applications [8].

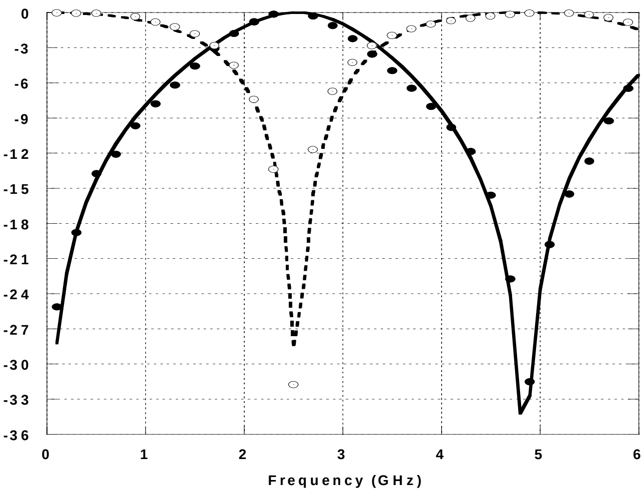


Fig.2 Frequency response of the FSS screen optimized by PSO. Transmission (continuous line) and reflection (dashed) coefficients as computed by MoM at normal incidence. Dots: results obtained by Ansoft HFSS®.

V. Conclusions

In this paper, we have presented a PSO algorithm for the synthesis of Frequency Selective Surfaces that simultaneously optimizes both the real and binary parameters. The optimization technique has been found not only to be reliable and flexible, but quite efficient in terms of computational time as well.

References

[1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", *Proc. IEEE Conf. Neural Networks* , Perth, Australia, November 1995, pp.1942-1948.

[2] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna", *IEEE AP-S Int. Symp. Dig.,* vol. 1, 2002, pp.314-317.

[3] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis", *IEEE Trans. Antennas Propag.,* Mar. 2004, vol. 52, no.3, pp. 771-779.

[4] N. Jin, Y. Rahmat- Samii, "Particle Swarm Optimization of Miniaturized Quadrature Reflection Phase Structure for Low-Profile Antenna", *IEEE Ant. Prop. Symp*., Washington DC, July 2005.

[5] G. Manara, A. Monorchio, and R. Mittra, "Frequency selective surface design based on genetic algorithm", *Electronic Letters*, vol. 35, August 1999, pp.1400-1401.

[6] J. Robinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics", *IEEE Trans. Antennas Propag.*, Feb. 2002, vol. 52, no.2, pp. 397-407.

[7] J. Kennedy and R. Eberhart, "A discrete binary version of the particle swarm algorithm", *IEEE Int. Conf. Systems, Man, and Cybernatics,* October 1997, vol. 5, pp.4104-4108.

[8] D.J. Kern, D.H.Werner, A. Monorchio, L. Lanuzza, M.J. Wilhelm, "The design synthesis of multiband artificial magnetic conductors using high impedance frequency selective surfaces", *IEEE Trans.Antennas and Propag.*, Volume 53, Issue 1, Part 1, Jan. 2005 Page(s) 8 – 17.