

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

4,800

Open access books available

122,000

International authors and editors

135M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Enhanced Particle Swarm Optimization for Design and Optimization of Frequency Selective Surfaces and Artificial Magnetic Conductors

Simone Genovesi¹, Agostino Monorchio¹ and Raj Mittra²

¹*Microwave and Radiation Laboratory, Dept. of Information Engineering, University of Pisa,*

²*Electromagnetic Communication Lab, PennState University*

¹*Italy, ²USA*

1. Introduction

Optimization methods are invaluable tools for the engineer who has to face the increasing complexity in the design of electromagnetic devices, or has to deal with inverse problems. Basically, an objective function $f(x)$ is defined where x is the set of parameters that has to be optimized in order to satisfy the imposed requirements. In design problems the parameters defined in x completely describe the features of the device (a printed antenna for example), and $f(x)$ is a measure of the system performance (gain or return loss). However, the objective function for a real-world problem may be nonlinear, may have many local extrema and may even be nondifferentiable. Numerous optimization methods that have been proposed in the literature can be divided into two groups – deterministic and stochastic. The former performs a local search which yields results that are highly influenced by the starting point, and sometimes requires the objective function to be differentiable. They might lead to a rapid convergence to a local extremum, as opposed to the global one and impose constraints on the solution domain that may be difficult to handle. The latter are largely independent of the initial conditions and place few constraints on the solution domain. They carry out a global search, and are able to deal with solution spaces with discontinuities, as well as a large number of dimensions and hence many potential local minima and maxima. Among the stochastic methods, for instance Monte Carlo and Simulated Annealing techniques, a particular subset also referred to as evolutionary algorithms have been recently growing in importance and interest. This class comprises the Genetic Algorithms (GA) (Goldberg, 1989), the Ant Colony Optimization (ACO) (Dorigo and Stutzle, 2004) and the Particle Swarm Optimization (PSO).

The PSO algorithm has been originally proposed by Kennedy and his colleagues (Kennedy and Eberhart, 1995) and it is inspired by a zoological metaphor of the social behavior of animals (birds, insects, or fishes) that are organized in groups (flocks, swarms, or schools). All of the basic units of the swarm, called particles (or agents) are trial solutions for the problem to be optimized, and are free to fly through the multidimensional search-space toward the optimal solution. The search-space represents the global set of potential results, where each dimension of this space corresponds to a

parameter of the problem to be determined. The swarm is largely self-organized, and coordination arises from the different interactions among agents. Each member of the swarm 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.

The organization of this chapter is as follows: Section 2 describes the implementation of a PSO algorithm employed in the design of Frequency Selective Surfaces. A parallelization of the PSO method is described in Section 3 that makes efficient use of all the available hardware resources to overcome the computational burden incurred in the process. A useful procedure for increasing the convergence rate is described in Section 4 and numerical results are provided to illustrate the reliability and efficiency of the new algorithm. Finally, concluding remarks are given in Section 5.

2. Optimization of Frequency Selective Surfaces

In this section 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. After a brief introduction of the nature of the FSSs and the applications in which they are employed, the PSO method developed for their optimization is described and a representative numerical example is given to demonstrate the effectiveness of this tool.

2.1 Frequency Selective Surfaces

At the end of the 18th century the American physicist David Rittenhouse (Rittenhouse, 1786), found that the light spectrum is decomposed into lines of different brightness and color, while observing a street lamp through his silk handkerchief. This was the first proof of the fact that non-continuous and periodic surfaces show different transmission properties for different frequencies of incident wave. The first device which takes advantage of this phenomenon is the parabolic reflector of wire sections, built by Marconi and Franklin in 1919 and, since then, FSSs have been further investigated and exploited for use in many practical applications. For instance, FSSs find use as subreflectors in dual frequency Cassegrainian systems and in radomes designed for antennas, where FSSs are used as pass band or stop band filters. They are employed to reduce the Radar Cross Section (RCS) of antennas outside their operating frequency band, and provide a reflective surface for beam focusing in reflector antenna system, realize waveguide filters and artificial magnetic conductors. At microwaves FSSs protect humans from harmful electronic radiation, as for instance, in the case of a microwave oven, in which the FSS printed on the screen doors totally reflects microwave energy at 2.45 GHz while allowing light to pass through. Recently, the FSSs have been employed at infrared (IR) frequencies for beam-splitters, filters and polarizers.

An FSS is either a periodic array of metallic patches printed on a substrate, or a conducting sheet periodically perforated with apertures. Their shape, size, periodicity, thickness of the metal screen and the dielectric substrate determine their frequency and angular response (Mittra et al., 1988; Munk, 2000).

2.2 Particle Swarm Optimization with mixed parameters

In the basic PSO algorithm, each agent in 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 :

$$X(i)=[x_1(i),x_2(i),\dots,x_n(i)]. \quad (1)$$

Each $x_n(i)$ component represents a parameter of the 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 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:

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

where $v_l(i)$ is the velocity along the l direction at the time step i ; w is the inertial weight; c_1 and c_2 are the cognitive and the social rate, respectively; $p_{\text{best},l}(i)$ is the best position along the l direction found by the agent during its own wandering up to i -th; $g_{\text{best},l}(i)$ is the best position along the l direction discovered by the entire swarm; and $\text{rand}()$ is a generator of random numbers uniformly distributed between 0 and 1. 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 -th, and Δt is the time step. An interesting insight into the basic PSO algorithm details may be found in (Robinson and Rahmat-Samii, 2002). This basic 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 in order to describe the shape of the unit cell (Manara et al., 1999). Therefore it is necessary to incorporate both of these features into the algorithm (Fig. 1). A discrete binary version of the PSO was first introduced by Kennedy and Eberhart (Kennedy and Eberhart, 1997), in which the concept of velocity loses its physical meaning and assumes the value of a probability instead. More 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, the expression in (2) has to be modified by imposing the condition that the value of $v_l(i)$ must be in the interval $[0.0, 1.0]$, and enforcing the constraint that any value outside this interval be unacceptable. As a consequence, a function T is defined to map the results of (2) within the allowed range. If $w=1$ and $c_1=c_2=2$, $v_l(i)$ is within 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:

$$\begin{aligned} &\text{if } (\text{rand}() < T(v_l(i))) \text{ then} \\ &\quad x_l(i) = \text{NOT}(x_l(i)) \\ &\text{else} \\ &\quad x_l(i) = x_l(i) \end{aligned} \quad (4)$$

where $\text{rand}()$ is the same random function adopted in (2) and the operator NOT indicates the binary negation of x_l .

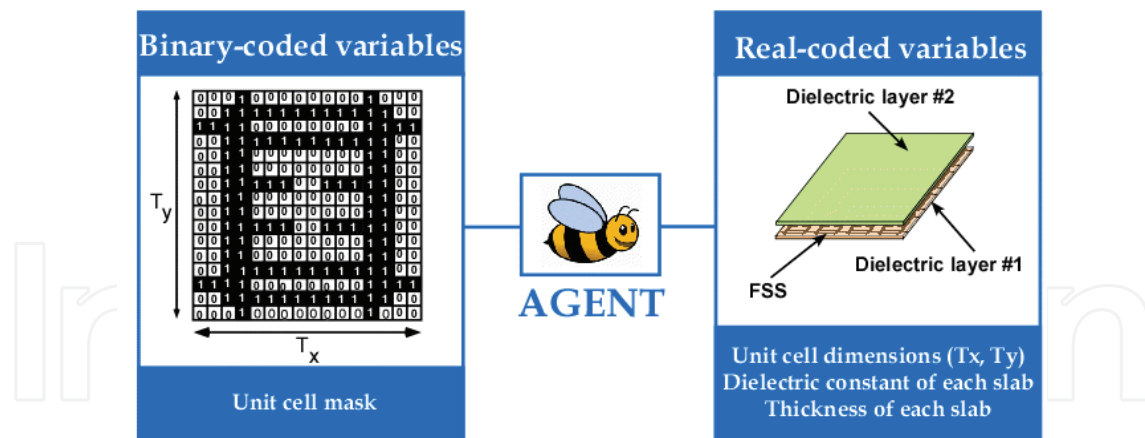


Figure 1. Each agent represents number and type of the parameters involved in the optimization process

This implies that if the random number is less than the probability expressed by the velocity, then the bit is changed. Hence, the faster the particle moves in that direction, the larger is the possibility of change.

The parameters that can be optimized by the algorithm for the design of an FSS structure are the shape of the unit cell, its dimensions, the permittivities of dielectric layers and their thicknesses. 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, the 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 (T_x , T_y). 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 either 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, where '1' implies the presence of perfect electric conductor and '0' designates an absence of conductor. The discretization adopted for the FSS binary mask can be 16×16 for a total of 256 binary parameters. This number reduces to 64 and 36, for a four-fold or eight-fold symmetry imposed to the unit cell, respectively. The analysis of the entire FSS structure is performed via an MoM code, employing roof top basis functions (Manara et al., 1999). The objective function (also referred to as the fitness function), which is employed to test the performance of the solution proposed by the PSO, is based on the mean square error between the reflection coefficient (or the transmission one) of the synthesized structure and the frequency mask which translates the requirements imposed by the user in one (or more) frequency band and for a set of incidence angles. It is apparent that in this case the aim is to minimize the fitness value and therefore we are in search of a the global minimum.

In order to demonstrate the capabilities of the PSO algorithm, a frequency mask is imposed to have a transmission coefficient less than -15 dB in the 0.1 GHz - 2.0 GHz band, less than -10 dB within the 10.0 GHz-12.0 GHz and to be transparent in the 5.0 GHz - 6.0 GHz band.

The algorithm has to optimize the shape of the unit cell and the thickness and dielectric constant values of two dielectric slabs which contains the FSS. The unit cell designed by the PSO is a square and has a period of 1 cm. The two dielectric slab have permittivities of $\epsilon_r=3.3$ and $\epsilon_r=7.68$, and thicknesses of 0.2 cm and 0.1 cm, respectively. The result is shown in Fig.2 as well as the unit cell shape represented in the binary variables.

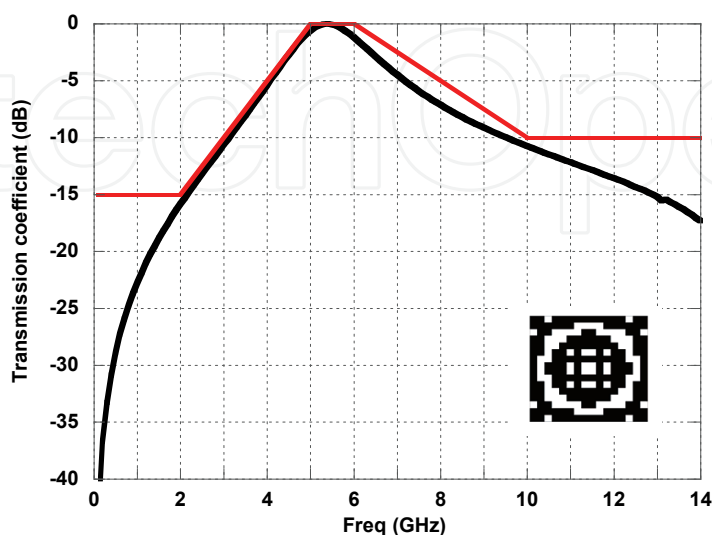


Figure 2. Comparison between the mask expressing the requirements imposed by the user (red line) and the transmission coefficient of the FSS optimized by the PSO algorithm (black line). In the inset the unit cell is reported

3. Parallel Particle Swarm Optimization

There have been many attempts in the past toward increasing the convergence of the PSO algorithm by modifying it (Clerc and Kennedy, 2002; Shi and Eberhart, 1999). This section will focus on an alternative approach, that involves an enhancement in the performance via the implementation of a parallel version of the PSO algorithm (PPSO) which is designed to address the CPU time issue. The parallel version can be useful, for example, for designing FSSs requiring a unit cell geometry with a fine discretization (e.g., 32×32), or for synthesizing a dual-screen version, both of which demand a significant computational effort, which is not easily handled by a single processor, at least within a reasonable time frame. The basic structure the parallel PSO algorithm is reported in Fig. 3(a). Starting from the observation that the updating of the velocity and the position of the agents, together with the evaluation of the scores of the fitness values to determine p_{best} and g_{best} requires a relatively small fraction of the time needed to compute the fitness function; hence the evaluation of the objective function is the only operation that is parallelized. The basic idea is to make a partitioning of the swarm among all the CPUs. The global partitioning strategy is clearly shown Fig. 3(b), where the case of four processors used in the optimization is considered for a swarm comprising eight particles.

A partition (two agents) of the swarm is assigned to each processor, which evaluates the fitness function of the given set of particles at each stage of iteration. Upon finishing these tasks, the processors communicate with each other to broadcast the best location they have found individually (red lines in Fig.4).

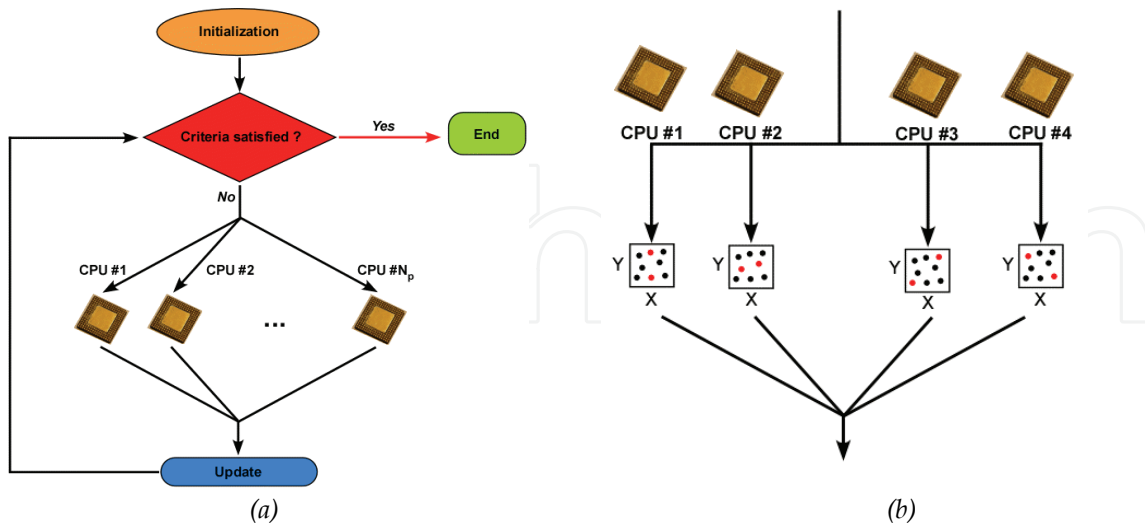


Figure 3. PPSO implementation: (a) Flow chart; (b) detail of work subdivision among all the available processors. Each CPU considers only the agents assigned (red dots)

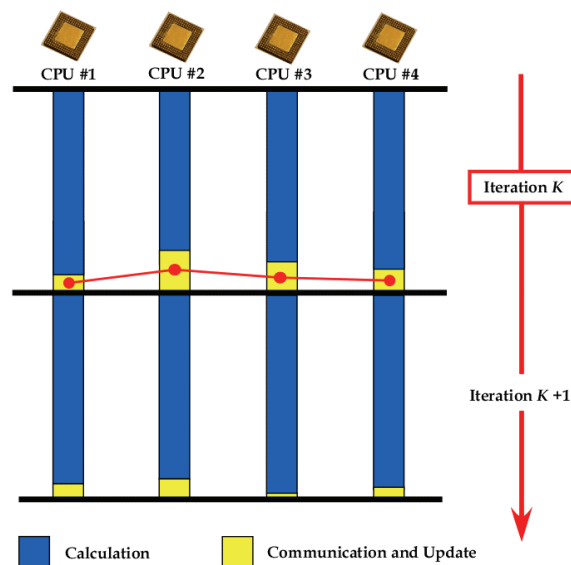


Figure 4. At iteration K , after the computation (blue), all the CPUs communicate to the others their results (red lines) and each processor perform the ranking to find the g_{best} (yellow)

Since the configuration analyzed by each processor is different, the time they require for their computation (highlighted in blue in Fig. 4) may vary slightly between the processors, even if the wait-time experienced by the faster processors is relatively small. All the processors have their own information, at the end of each evaluation step, as well as the latest information from the others about the best areas; hence, it is relatively easy to find the g_{best} . There is no master processor to carry out the ranking task and, hence, only a single transfer of information is needed at each iteration step. As is evident from Fig. 5, the general trend is a decrease of the overall simulation time.

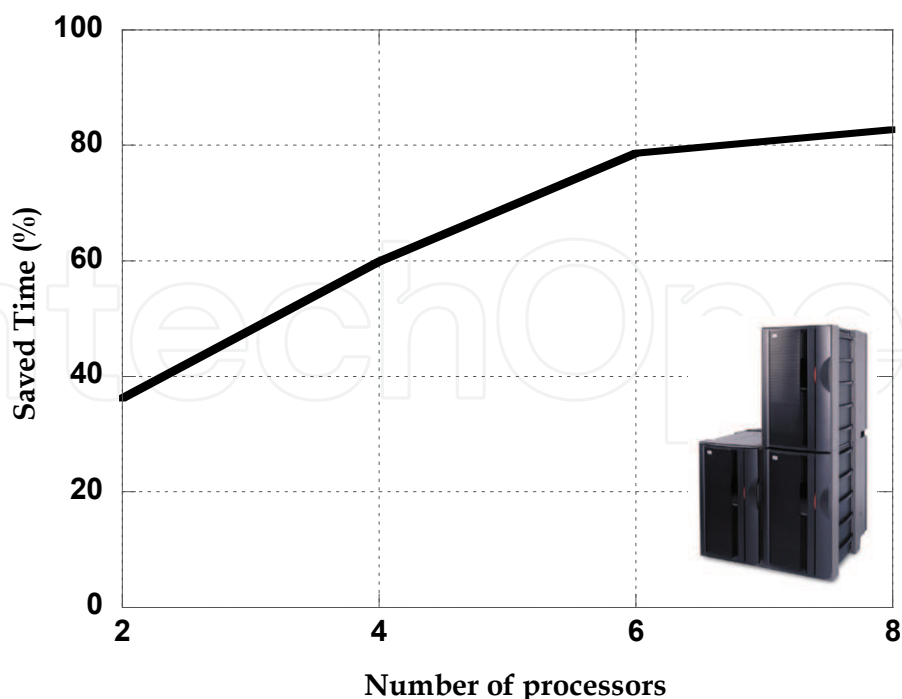


Figure 5. General trend of the saved time achieved by employing the PPSO approach

4. Space partitioning for increasing convergence rate

The problems of control of parameters and their tuning has been widely investigated (Clerc and Kennedy, 2002; Shi, and Eberhart, 2001) in the context of PSO, who have dealt with open issues such as premature convergence and stagnation into local minima. Furthermore, the effect of changing the neighborhood topology has been discussed extensively (Clerc, 1999; Kennedy, 1999; Lovbjerg et al., 2001; Abdelbar et al., 2005). However, to the best of our knowledge, the initialization of the position of the particles within the search space has not been subject of the same attention. The initialization of the position of the particles has a deep impact on the rate of convergence a in PSO and, therefore, has to be carefully taken into account. Since the agents are randomly located in most cases, it is possible that some areas may have higher densities of particles than others, especially if the multidimensional domain is large. Of course, this inhomogeneity in the distributions of the agents does not prevent them from pursuing the goal but can affect the time required for approaching the final solution. We propose to circumvent this difficulty by subdividing the solution space into sub-domains within which groups of agents are initially located in order to guarantee the homogeneous distribution of agents all over the computational domain. Each particle cooperates only with those particles in its own group independently from the other groups. After a fixed number of iterations, the sub-boundaries are removed, the best positions found by each group are scored and the actual global best location is revealed to all. It is demonstrated that the first part of the optimization process, managed by particles inside the sub-boundaries, improves the speed with which we find the optimal solution and hence increases the convergence rate of the process. The efficiency of the proposed implementation, referred to enhanced PSO in this Section, has been verified through the optimization of commonly employed test functions as well as of a complex electromagnetic problem, *viz.*, the design of Artificial Magnetic Conductors (AMCs).

4.1 Space partitioning

We now discuss the space partitioning scheme using a slightly modified notation than used in Section 2. Let us denote to the position of the generic agent k at a certain instant i by using the vector X given by:

$$X^k(i)=[x^{k_1}(i),x^{k_2}(i),\dots,x^{k_n}(i)], \quad (5)$$

and let $p^{k_{\text{best},n}}$ be the best position along the direction n found by the agent k during its own wandering up to the i -th time step, and let $g_{\text{best},n}$ be the best position along the direction n , discovered by the entire swarm at time step i . The particle is free to fly inside the defined n -dimensional space, within the constraints imposed by the n boundary conditions, which delimit the extent of the search space between a minimum ($x_{n,\text{min}}$) and maximum ($x_{n,\text{MAX}}$) and, hence, the values of the parameters during the optimization process. Accordingly, at the generic time step $i+1$, the velocity of the simple particle k along each direction is updated by following the rule:

$$v_n^{k}(i+1)=w * v_n^{k}(i)+c_1 * \text{rand}() * (p_{\text{best},n}^{k}(i)-x_n^{k}(i))+c_2 * \text{rand}() * (g_{\text{best},n}(i)-x_n^{k}(i)), \quad (6)$$

```

Define the allowed range of each dimension (boundaries)
Set  $i=1$ 
for  $k=1, \text{number\_of\_agents}$ 
  for  $n=1, \text{number\_of\_dimensions}$ 
    Random initialization of  $x_n^k(i)$  within the allowed range  $[x_{n,\text{min}}; x_{n,\text{MAX}}]$ 
    Random initialization of  $v_n^k(i)$  proportional to the dynamic of dim.  $n$ 
  next  $n$ 
next  $k$ 
for  $j=1, \text{number\_of\_iterations}$ 
  for  $k=1, \text{number\_of\_agents}$ 
    Evaluate  $\text{fitness}^k(i)$ , the fitness of agent  $k$  at instant  $i$ 
  next  $k$ 
  Rank the fitness values of all agents
  for  $k=1, \text{number\_of\_agents}$ 
    if  $\text{fitness}^k(i)$  is the best value ever found by agent  $k$  then
       $p_{\text{best},n}^k(i) = x_n^k(i)$ 
    end if
    if  $\text{fitness}^k(i)$  is the best value ever found by all agents then
       $g_{\text{best},n}(i) = x_n^k(i)$ 
    end if
  next  $k$ 
  Update agent velocity by using (6) and limit if required
  Update agent position, check it with respect to the Boundaries
   $i=i+1$ 
next  $j$ 

```

Figure 6. PSO implementation with initialization by using boundary conditions

To refresh the memory of the standard particle swarm optimization algorithm, we present its pseudocode in Fig. 6, since it is useful to understand the novelty introduced by the initialization of the sub-boundaries. The solution we propose is based on the simple observation that there exists a high probability that the initial step, which entails a random position of all the agents, can determine a non-uniform coverage of the search domain. This fact affects the convergence rate, especially if the domain is large compared to the number of agents involved in the search process. Even if the algorithm is able to find the optimal solution, the process could be speeded up by adopting an approach which will be detailed

in this section. The underlying concept upon which the algorithm is based is to distribute the agents uniformly at the start of the optimization process. The agents are organized into equal groups and these groups are then forced to exploit a sector of the domain defined by the sub-boundaries. This concept is described in Fig. 7 for a three-dimensional domain.

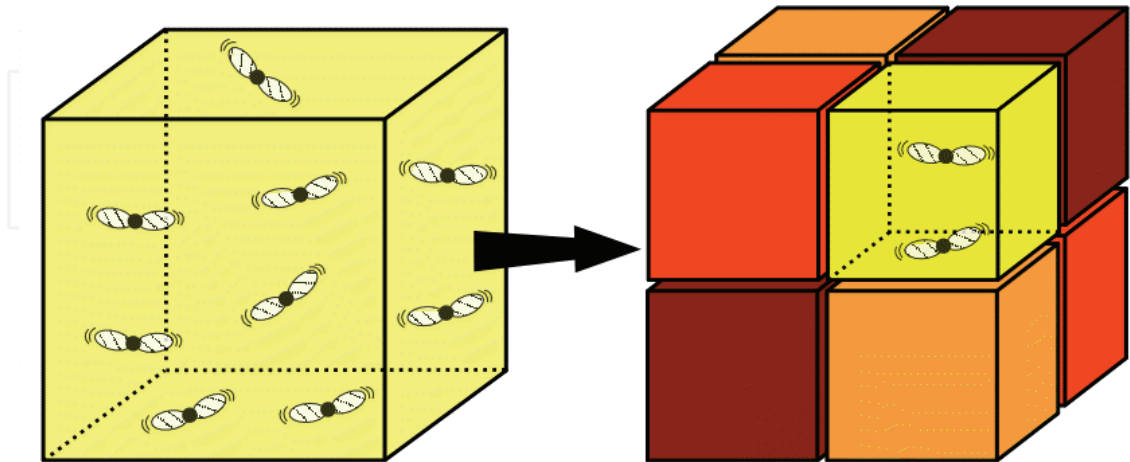


Figure 7. The domain defined by the boundaries is split into sectors defined by sub-boundaries within groups of agents wandering in search of the best location

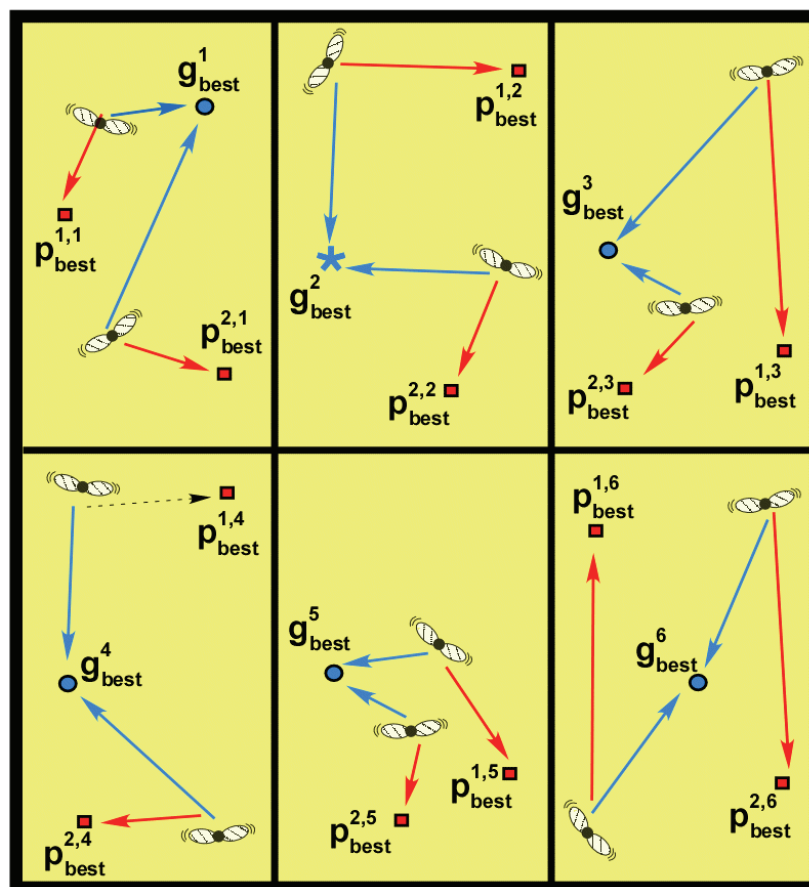


Figure 8. After the last iteration in sub-domain mode, and before starting the entire domain discovery, each particle is attracted by its own sub-domain best (blue dots) and its local best (red squares). The blue star in sector 2 is the best of all the sectors' bests

The domain is subdivided into sectors (or sub-domains) by using sub-boundaries that split one or more dimensions into equal intervals. The number of sub-boundaries cannot exceed the number of agents but, as it will be evident later, they should not produce groups that contain very few agents. During the initial stages, each group flies inside the assigned sub-domain and, hence, each group g has its own “sub-domain best” (indicated by $g_{best,n}^g$). Furthermore, each agent k in the group g has its own position $x^{k,g}$ and the local best location ($p_{best}^{k,g}$). The sub-boundaries pose impassable limits and consequently, none of the agents of one group can cross these boundaries to enter another sector. This guarantees that the number of agents in each sector is constant and so that the homogeneity of their spread within the multidimensional domain is preserved. Once the number of iterations dedicated to this process is exceeded, the barriers imposed by the sub-boundaries are removed and the particles are free to fly all over the entire domain. The “global best” is then chosen from those found in the sectors by the groups while the “local best” position of each agent is preserved. The operation executed at the exact instant of the passage from the sub-boundary conditions to the global boundary conditions is described in Figs. 8 and 9 for a two-dimensional case. To illustrate the differences introduced in this modified version of the PSO, its pseudocode is presented in Fig. 10.

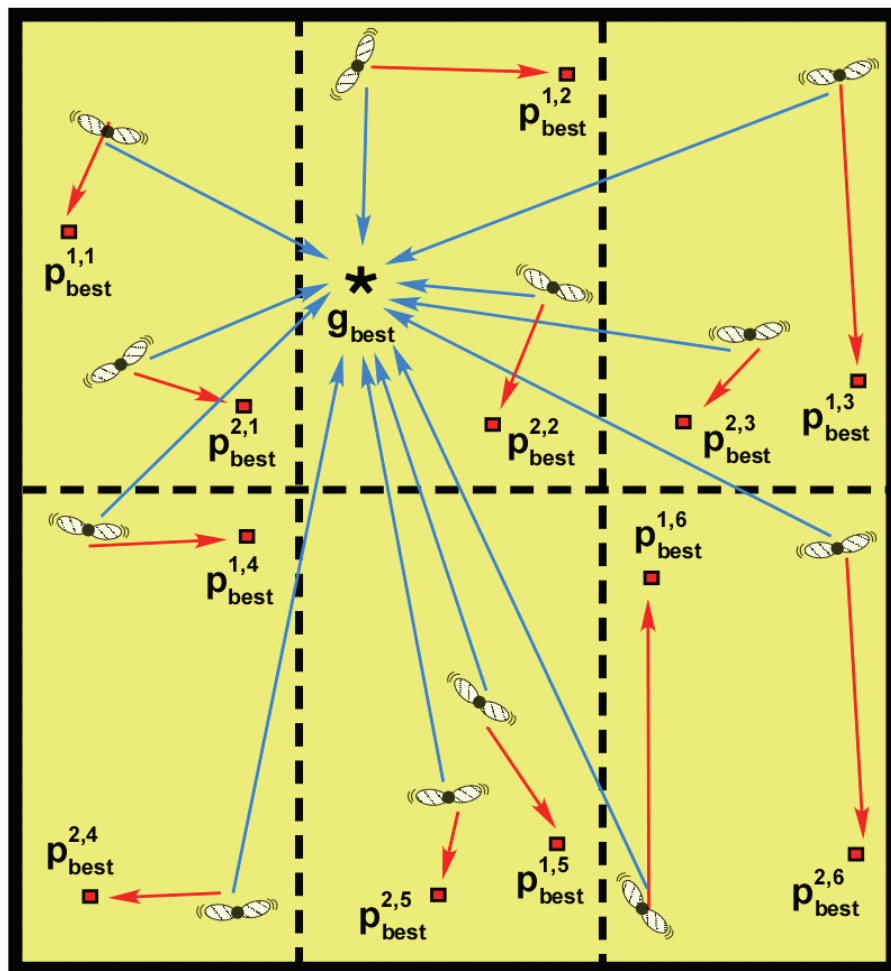


Figure 9. Opening the sub-boundaries: all the agents gain the information about the global best as soon as the barriers imposed by the sub-boundaries are removed. They are attracted both by that location as well as by the own local best previously found

To point out the changes in the results obtained by using this new PSO implementation, we have optimized several functions used as test beds for studying the performance of optimizers (Clerc and Kennedy, 2002). In particular, the following functions have been considered. The first type is the Rastrigin function defined as:

$$f_1(x) = \sum_{i=1}^N \left(x_i^2 - 10 \cos(2\pi x_i) + 10 \right), \quad (7)$$

with $(-5.12 < x_i < 5.12)$. The second type is the Griewank function $(-600 < x_i < 600)$:

$$f_2(x) = \frac{1}{4000} \sum_{i=1}^N x_i^2 - \prod_{i=1}^N \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1. \quad (8)$$

The last function considered is the Rosenbrock function:

$$f_3(x) = \sum_{i=1}^N \left(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right). \quad (9)$$

with $(-50 < x_i < 50)$.

All the introduced functions have a global minimum equal to zero. Several simulations have been run for each of these functions, both with the standard PSO algorithm as well as with the new proposed one.

Three different sizes of the swarm have been considered, comprising of 16, 20 and 32 agents, respectively. Furthermore, to better understand the influence of the sub-boundary initialization, we have addressed the problem with a variable number of sectors (2, 4, 8, and 16) and, hence, different number of groups. As mentioned previously each sector contains only one group. The maximum allowed number of iterations to reach the minimum has been set to 150. Except for the boundary case, we have run half of the total amount of iterations with active sub-boundaries. This choice is to be regarded only as a suggestion, which is important for efficient cooperation of all the agents acting together – one of the most important features of the PSO algorithm. The results for $N=3$ are shown in Table I.

The first value expresses the average number of iterations necessary to approach the minimum with a tolerance of less than 0.01. The abbreviation N.R. (not reached) means that this requirement has not been satisfied up to the 150-th iteration. The second value within the brackets is the number of fitness evaluations which indicates the number of calls to the solver. We have deliberately omitted to consider the case of 20 agents and 8 sectors because it is not possible to have groups with the same number of agents. From the above results, it is possible to state that the initialization with the sub-boundaries not only helps us to reach the convergence more rapidly, but also that the more we increase the number of divisions the less we improve the performance. Moreover, in the case of 16 groups the efficiency drops dramatically and the results are even worse than without sub-boundaries. This fact suggests a logical conclusion, *viz.*, that there is a limit to the improvement that we can achieve by increasing the number of subdomains beyond a certain point. Of course, the number of sectors is also limited by the number of agents, since a group has to be composed at least by two agents. The above results lead us to conclude that the initialization with the sub-boundaries helps us to reach the convergence more rapidly, but we have to prevent the

use of very small groups. Therefore, even if a group has to be composed at least by two agents, these results suggested us to use a minimum of 4 agents in each group.

```

Set  $i=1$ 
for  $g=1, \text{number\_of\_groups}$ 
  for  $k=1, \text{number\_of\_agents\_in\_the\_group}$ 
    for  $n=1, \text{number\_of\_dimensions}$ 
      Random initialization of  $x^{k,g_n}(i)$  within the range of subdomain
    #g
      Random initialization of  $v^{k,g_n}(i)$  prop. to the dynamic of subd. #g
    next  $n$ 
  next  $k$ 
next  $g$ 
do while (Sub_boundary_case)
  Flag_set_global_best = FALSE
  for  $g=1, \text{number\_of\_groups}$ 
    for  $k=1, \text{number\_of\_agents\_in\_the\_group}$ 
      Evaluate  $\text{fitness}^{k,g}(i)$ , the fitness of agent  $k$  in group  $g$  at instant  $i$ 
    next  $k$ 
  next  $g$ 
  for  $g=1, \text{number\_of\_groups}$ 
    Rank the fitness values of all agents included only in group  $g$ 
  next  $g$ 
  for  $g=1, \text{number\_of\_groups}$ 
    for  $k=1, \text{number\_of\_agents\_in\_the\_group}$ 
      if  $\text{fitness}^{k,g}(i)$  is the best value ever found by agent  $k$  in group  $g$  then
         $p^{k,g_{\text{best},n}}(i) = x^{k,g_n}(i)$ 
      end if
      if  $\text{fitness}^{k,g}(i)$  is the best value ever found by all agents then
         $g^{g_{\text{best},n}}(i) = x^{k,g_n}(i)$ 
      end if
    next  $k$ 
  next  $g$ 
   $i=i+1$ 
  if ( $i \geq \text{sub\_boundaries\_iterations}$ ) then Sub_boundary_case= FALSE
end do
if (Flag_set_global_best = FALSE) then
  Flag_set_global_best = TRUE
  Rank all the  $g^{g_{\text{best},n}}(i)$  and set the actual  $g_{\text{best},n}(i)$ 
else
  follow, with the actual value of  $i$ , the procedure showed in Fig.6
end if

```

Figure 10. Pseudocode of the modified PSO. During the preliminary iterations the agents seek together, organized in groups, in an area defined by the sub-boundaries. After this stage, they are set free and can move all over the solution space

Rastrigin function

# of agents	No Sub-Boundaries	2 Groups	4 Groups	8 Groups	16 Groups
16	N.R.	142 (2272)	117 (1872)	98 (1568)	N.R.
20	N.R.	110 (2200)	70 (1400)	–	–
32	N.R.	60 (1920)	40 (1280)	34 (1088)	138 (4416)

Griewanck function

# of agents	No Sub-Boundaries	2 Groups	4 Groups	8 Groups	16 Groups
16	122 (1952)	110 (1760)	93 (1488)	80 (1280)	130 (4160)
20	96 (1920)	84 (1680)	74 (1480)	–	–
32	80 (2560)	52 (1664)	45 (1440)	38 (1216)	112 (3584)

Rosenbrock function

# of agents	No Sub-Boundaries	2 Groups	4 Groups	8 Groups	16 Groups
16	44 (704)	31 (496)	18 (288)	15 (240)	58 (1856)
20	30 (600)	23 (460)	12 (240)	–	–
32	19 (608)	14 (448)	7 (224)	7 (224)	55 (1760)

Table 1. Results obtained by using sub-boundaries initialization in solving benchmark functions

4.2 Artificial Magnetic Conductor case study

In recent years, much attention has been devoted to the problem of designing Artificial Magnetic Conductors (AMC) that find a variety of applications, especially in the field of low-profile antennas (Sievenpiper et al., 1999; Kern et al. 2005). The zero-phase reflection coefficient at the resonance frequency allows one to place the source close to the artificial magnetic ground plane, and this offers the possibility of reducing the total dimension of the device. In order to realize an AMC ground plane, one can exploit the use of planar architectures which incorporate an FSS printed on a grounded dielectric slab (Kern et al. 2005). As shown in Fig. 6(a), once the number and the configuration of the dielectric layers have been chosen, it is necessary to design the FSS unit cell, choose the values of dielectric

constants as well as the thickness of each dielectric slab so as to realize the AMC behavior at the desired frequency.

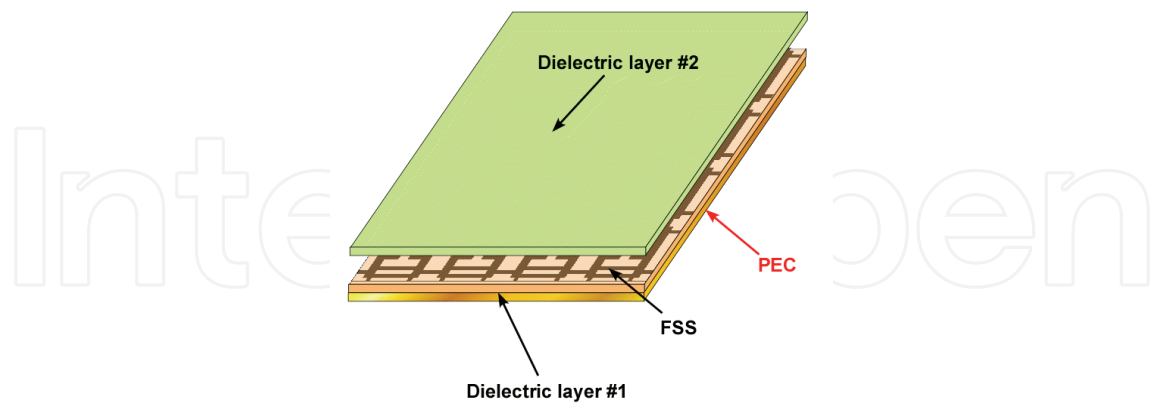


Figure 11. Geometry of an AMC screen A FSS is printed on a dielectric substrate backed by a perfect electric conductor (PEC)

A quantity proportional to the root mean square of the difference between the actual electric field reflection coefficient (Γ_E) and the desired one ($\text{Re}\{\Gamma_{\text{AMC}}\}=1, \text{Im}\{\Gamma_{\text{AMC}}\}=0$), for both TE and TM modes, is used to evaluate the performance of the structure. In order to evaluate the performance of the PSO enhanced with sub-boundaries we have run several simulations, each one carrying out 300 iterations, with different number of sectors. Our aim is to design an AMC screen acting as a PMC at 2.5 GHz, optimizing both the unit cell and the characteristic of two dielectric slabs (a superstrate and a substrate). At each simulation (except for the case with no sub-boundaries), one half of these iterations are carried out by using sub-boundaries and the average value of the fitness considered is the one of the best sector. The number of particles in the swarm is 32. The results are summarized in Fig. 12 where we show the convergence rate for each sub-domain configuration.

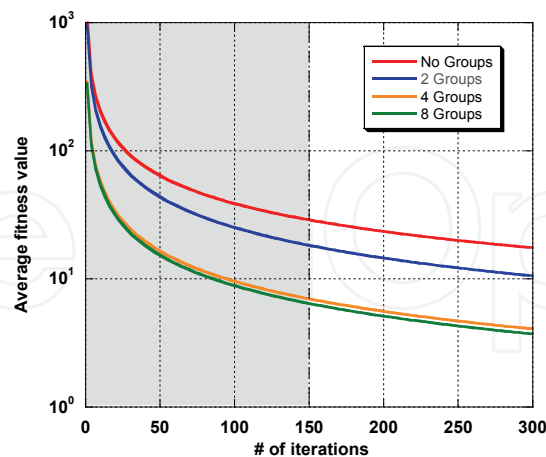


Figure 12. Trend of the convergence rates for four different cases. The grey zone represents the part of the iteration run by using sub-boundaries (except for the no-groups case)

We note that there is an improvement in the performance as we increase the number of groups and that, as in the previous case, the advantages of this approach are not directly proportional to the number of sub-boundaries utilized. In fact, we gain an advantage over

the conventional PSO when we use two groups and the performance is better if we change the number of groups to four. However, it is not worthwhile to go beyond this value and to further subdivide the domain into eight groups. Moreover the sub-boundary approach is not applied to the binary map in this case and, hence, it reduces the impact of further subdivisions of the domain. As an example, in Fig. 13 we show an AMC screen, together with its electromagnetic performance, obtained in the case of a swarm initialized by using 4 groups.

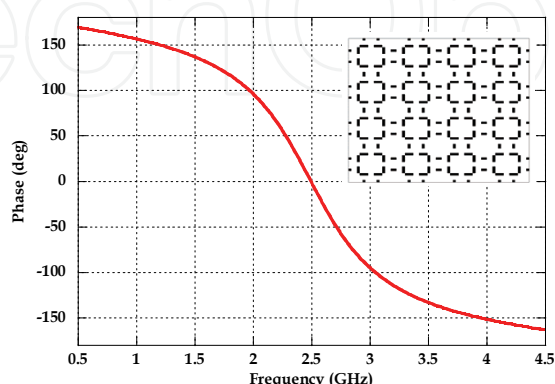


Figure 13. Phase of the reflection coefficient vs. frequency for the AMC screen shown in the inset. The phase response is reported for normal incidence

5. Conclusion

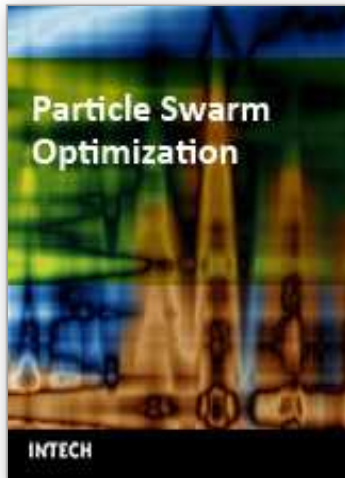
In this chapter, we have address the problem of efficiently synthesizing Frequency Selective Surfaces by using PSO. We have presented our specifically derived particle swarm optimization procedure which is able to handle, simultaneously, both real and binary parameters. We proposed a parallel version of the PSO algorithm to face challenging problems, which may require hardware resources and computational time that cannot be handled by a single CPU. Finally, we have introduced a novel strategy for the initialization of the agents' position within the multidimensional solution domain to further improve the convergence rate. This new procedure has been shown to be reliable with benchmark functions and has been successfully applied to the synthesis of Artificial Magnetic Conductors.

6. References

- Abdelbar, A. M., Abdelshahid, S. & Wunsch D. C. (2005). Fuzzy PSO: a generalization of particle swarm optimization, *Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN '05)*, Vol. 2, pp. 1086-1091.
- Clerc, M. (1999). The swarm and the queen: towards a deterministic and adaptive particle swarm optimization, *Proc. Evolutionary Computation (CEC 99)*, Vol. 3, pp. 1951-1957.
- Clerc, M. & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space, *IEEE Trans. Evol. Comput.*, Vol. 6, No. 1, pp. 58-73.
- Dorigo, M. & Stutzle, T. (2004). *Ant Colony Optimization*, MIT Press, ISBN 0262042193.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, Reading, MA.

- Kennedy, J. & Eberhart, R.C. (1995). Particle Swarm Optimization, *Proceedings of IEEE International Conference on Neural Networks*, Vol. 4, pp. 1942-1948. Piscataway, NJ.
- Kennedy, J. & Eberhart, R.C. (1997). A discrete binary version of the particle swarm algorithm, *IEEE Int. Conf. Systems, Man, and Cybernatics*, Vol. 5, pp. 4104-4108.
- Kennedy, J. (1999). Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance, *Proc. Evolutionary Computation (CEC 99)*, Vol.3, pp. 1931-1938.
- Kern, D. J., Werner, D., Monorchio, A., Lanuzza, L. & Wilhelm M. (2005). The Design Synthesis of Multiband Artificial Magnetic Conductors Using High Impedance Frequency Selective Surfaces, *IEEE Trans. On Antennas Propag.*, Vol.53, No.1.
- Lovbjerg, M., Rasmussen, T. K. & Krink T. (2001). Hybrid particle swarm optimizer with breeding and subpopulations, *Proc. 3rd Genetic and Evol. Comp. Conf. GECCO-2001*.
- Manara, G., Monorchio, A & Mittra, R. (1999). Frequency selective surface design based on genetic algorithm, *Electronic Letters*, Vol. 35, pp.1400-1401.
- Mittra, R.; Chan, C. & Cwik, T. (1988). Techniques for Analyzing Frequency Selective Surfaces-A Review, *IEEE Proceedings*, Vol. 76, Issue 12, pp.1593-1615.
- Munk, B. A. (2000). *Frequency Selective Surfaces – Theory and Design*, Wiley, New York.
- Rittenhouse, D. (1786). An optical problem, proposed by Mr. Hopkins, and solved by Mr. Rittenhouse, *Transaction of the American Philosophical Society*, Vol. 2, pp. 201-206.
- Robinson, J. & Rahmat-Samii, Y. (2002). Particle Swarm Optimization in Electromagnetics, *IEEE Trans. Antennas Propag.*, Vol. 52, No. 2, pp. 397-407.
- Shi, Y. & Eberhart, R. C. (1999). Empirical study of particle swarm optimization, *Proc. Evolutionary Computation (CEC 99)*, Vol. 3, pp. 1945-1950.
- Shi, Y. & Eberhart, R. C. (2001). Fuzzy adaptive particle swarm optimization, *Proc. Evolutionary Computation (CEC 01)*, Vol. 1, pp. 101-106.
- Sievenpiper, D., Zhang, L., Broas, R., Alexopolous, N. & Yablonovitch E. (1999). High impedance frequency selective surfaces with a forbidden frequency band, *IEEE Trans. Microwave Theory Tech.*, Vol.47, No. 11, pp. 2059-2074.

IntechOpen



Particle Swarm Optimization

Edited by Aleksandar Lazinica

ISBN 978-953-7619-48-0

Hard cover, 476 pages

Publisher InTech

Published online 01, January, 2009

Published in print edition January, 2009

Particle swarm optimization (PSO) is a population based stochastic optimization technique influenced by the social behavior of bird flocking or fish schooling. PSO shares many similarities with evolutionary computation techniques such as Genetic Algorithms (GA). The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike GA, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. This book represents the contributions of the top researchers in this field and will serve as a valuable tool for professionals in this interdisciplinary field.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Simone Genovesi, Agostino Monorchio and Raj Mittra (2009). Enhanced Particle Swarm Optimization for Design and Optimization of Frequency Selective Surfaces and Artificial Magnetic Conductors, Particle Swarm Optimization, Aleksandar Lazinica (Ed.), ISBN: 978-953-7619-48-0, InTech, Available from: http://www.intechopen.com/books/particle_swarm_optimization/enhanced_particle_swarm_optimization_for_design_and_optimization_of_frequency_selective_surfaces_and

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2009 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.

IntechOpen

IntechOpen