A Sub-Boundary Approach for Enhanced Particle Swarm Optimization and its Application to the Design of Artificial Magnetic Conductors

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

***Abstract*— The particle swarm algorithm is a newly introduced method for electromagnetic optimization problems that is based on the observation of swarm intelligence and particle behavior. The particles, while refining their knowledge about the best location in the search area, also communicate with each other and share their own experience. This combined and synergic use of information yields a powerful tool for solving problems that require the optimization of a relatively large number of parameters.**

This paper proposes a novel strategy for the initialization of the agents' position within the multidimensional solution domain. In particular, the domain is initially subdivided into sub-domains where the agents are more uniformly distributed. At a second stage, the sub-boundaries are removed and the best position information of each group is passed to each agent; the agents are therefore allowed to explore the whole search space. This procedure results in being very efficient with a strong improvement of the convergence rate to the optimal solution. A comparison between the performance of this new implementation and that of the basic particle swarm algorithm is presented for several test cases. Finally, this new procedure is successfully applied to the synthesis of Artificial Magnetic Conductors (AMCs).

***Index Terms* — Particle Swarm Optimization (PSO), Optimization methods, Artificial Magnetic Conductors (AMCs).**

I. Introduction

THE particle swarm optimization (PSO), originally proposed by Kennedy and Eberhart [1], is a powerful method for solving global optimization problems. It is inspired by a zoological metaphor of the social behavior of animals (birds, insects, or fishes) which organize themselves in groups (flock, swarm, or school). In swarms no one is in charge and therefore, the candidate solutions (particles or agents) are free to fly through the multidimensional search space towards the optimal solution. In fact, each dimension of this space represents a parameter of the problem to be optimized. During the search procedure to seek the best location, the agents change their positions with time and constantly adjust their velocities according to their own experience and those of the other particles. Each agent makes use of the best position reached, both by itself and by the rest of the swarm, and then is stochastically attracted both toward the best personal as well as toward the best overall location. Concurrently, the agent evaluates the goodness of its current location in the problem space.

Even though the *No Free Lunch* theorem [2] affirms that there is no definite answer to the quest for the best search strategy, the PSO algorithm has nonetheless proven to be successfully in solving a wide variety of electromagnetic problems. It has been applied to the array and antenna synthesis [3-10], for designing electromagnetic absorbers [11], microstrip filters [12] and even in the field of imaging [13-14].

In common with the various optimization techniques, much effort has been invested towards improving the performance and the reliability of the PSO. The problems of PSO parameter control and their tuning has been widely investigated in [15-19], since there are open issues such as premature convergence and stagnation into local minima that need to be addressed. Furthermore, the effect of changing the neighborhood topology has been discussed extensively in [20-25]. Finally, because of its intrinsic nature, the PSO algorithm has also been implemented on parallel platforms [11,26].

However, to the best of our knowledge, the initialization of the position of the particles within the search space, and its effect on the rate of convergence has not been fully addressed yet. Since the agents are randomly located in most cases, it may occur, especially if the multidimensional domain is large, that some areas have higher densities of particles than others. 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.

The solution proposed to circumvent this difficulty is to subdivide the solution space into sub-domains within which

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). He is currently a Visiting Scholar at the EMC Lab in Penn State.

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).

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

groups of agents are initially located. These sub-domains are defined by sub-boundary conditions that are derived from the original boundaries, and their presence guarantees 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, ensuring an improvement of exploration of their part of the global domain defined by the sub-boundaries. Then, 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 to find the optimal solution and hence increases the convergence rate of the process.

The paper is organized as follows. A description of the conventional PSO algorithm is given in Section II. In Section III the novel sub-boundaries approach for the swarm initialization is presented and proofs of its reliability and efficiency are provided by using commonly adopted test functions. Section IV describes a dedicated PSO code for the design of artificial magnetic conductors (AMCs). Finally, the improvement obtained by incorporating the new initialization approach in the PSO code is discussed.

II. Conventional PSO Algorithm

In the PSO algorithm there is no central control and, in the process of finding the optimal solution, each particle of the swarm flies in an *n*-dimension space, refining its knowledge about the search space not only by its own exploitation but also by sharing with others the information it has gained. The position of the generic agent *k* at a certain instant *i* is expressed by the vector *X* as follows:

$$X^k(i) = \left(x_1^k(i), x_2^k(i), \ldots, x_n^k(i)\right) \tag{1}$$

where 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 certain position and traverses the multidimensional space with a random velocity, both in direction and magnitude, which are generally limited by the dynamic range of that dimension. 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 ($x_{n,min}$) and maximum ($x_{n,MAX}$) and, hence, the values of the parameters during the optimization process.

At the generic time step i+1, the velocity of the simple particle *k* along each direction is updated following the rule

$$\begin{aligned} v_n^k(i+1) = w * v_n^k(i) + c_1 * \text{rand}() * \left(p_{best,n}^k(i) - x_n^k(i)\right) \\ + c_2 * \text{rand}() * \left(g_{best,n}(i) - x_n^k(i)\right) \end{aligned} \tag{2}$$

This implies that, at each iteration step, the new velocity is the sum of three components. The first one is the actual velocity, scaled by the factor *w* which represents the inertia of the particle, while the two following terms express the attraction due to $p_{best,n}^k$ and $g_{best,n}$ found at the time step *i*. 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 quantities introduced in (1) are briefly summarized in Table I.

TABLE I
Particle Swarm Optimization Variables

$x_n^k(i)$	: position of agent *k* along direction *n* at time step *i*
$v_n^k(i)$	: velocity of agent *k* along direction *n* at time step *i*
w	: inertial weight (or momentum, or habit)
c_1	: cognitive rate
c_2	: social rate
$p_{best,n}^k(i)$	: best position along direction *n* found by the agent during its own wandering up to the *i-th* time step
$g_{best,n}(i)$	: best position along direction *n* discovered by the entire swarm at time step *i*
rand( )	: generator of random numbers uniformly distributed in [0;1]

The random generator produces the appropriate chaotic component of a real swarm. Finally, the position of each particle is simply updated according to the equation:

$$x_n^k(i+1) = x_n^k(i) + v_n^k(i) * \Delta t \tag{3}$$

where the current position of the agent along the direction *n* at the iteration # $(i+1)$ is the previous one incremented by the product between its actual velocity and the time step Δt. The boundary conditions implemented could be of different types [27,28] and change the state or the velocity of the particle whenever it either hits or crosses the designated borders.

For a concise description of the standard particle swarm optimization algorithm, we present its pseudocode in Fig. 1, since it is useful to understand the novelty introduced by the initialization of the sub-boundaries.

Define the allowed range of each dimension (boundaries)
Set i=1
for k=1, *number_of_agents*
  for n=1, *number_of_dimensions*
    Random initialization of $x_n^k(i)$ within the allowed range [$x_{n,min}$; $x_{n,MAX}$]
    Random initialization of $v_n^k(i)$ proportional to the dynamic of dim. n
  next n
next k
for j=1, *number_of_iterations*
  for k=1, *number_of_agents*
    Evaluate $fitness^k(i)$, the fitness of agent k at instant i
  next k
  Rank the fitness values of all agents
  for k=1, *number_of_agents*
    if $fitness^k(i)$ is the best value ever found by agent k **then**
      $p_{best,n}^k(i) = x_n^k(i)$
    end if
    if $fitness^k(i)$ is the best value ever found by all agents **then**
      $g_{best,n}(i) = x_n^k(i)$
    end if
  next k
Update agent velocity by using (2) and limit if required
Update agent position by using (3), check it with respect to the Boundaries
i=i+1
next j

Fig.1 PSO implementation with initialization by using boundary conditions

III. A New Initialization Method

The solution herein proposed has its basis on the simple observation that there exists a high probability that the initial step of randomly positioning 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 which the algorithm is based upon is to uniformly distribute the agents 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. 2 for a three-dimensional domain.

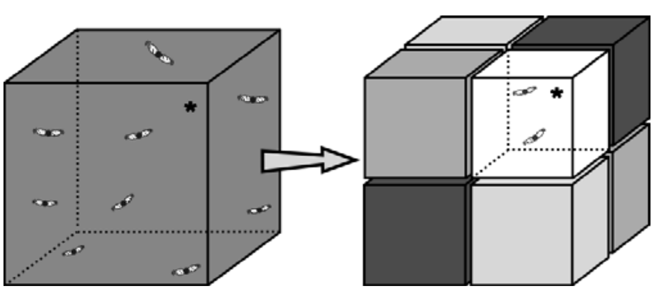


Fig. 2 The domain defined by the boundaries is split into sectors defined by sub-boundaries within groups of agents wandering in search of the best location. The sub-division could play an important role in finding faster the best solution (asterisk)

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, should not produce groups that are too small. 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 local best location ($p_{best}^{k,g}$). The sub-boundaries pose impassable limits and consequently, none of the agents of one group can pass through these boundaries and enter inside 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 Fig. 3 and Fig. 4 for a two-dimensional case.

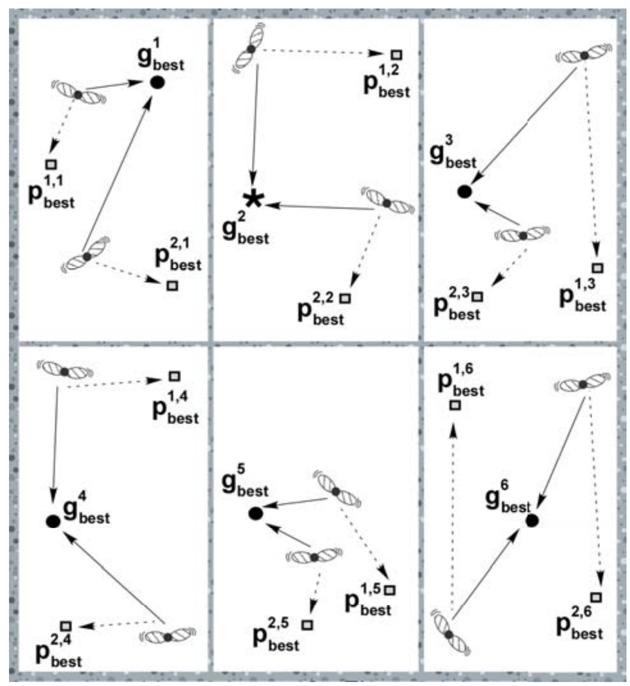


Fig. 3 After the last iteration in sub-domain mode, and before starting the entire domain discovery, each particle is attracted by its own sub-domain best (black dots) and its local best (grey squares). The star in sector 2 is the best of all the sectors' bests.

To highlight the differences introduced in this modified version of the PSO, its pseudocode is presented in Fig. 5.

To underline the change in results obtained by using this new PSO implementation, we have optimized several functions used as test beds for studying the performance of optimizers [15,27].

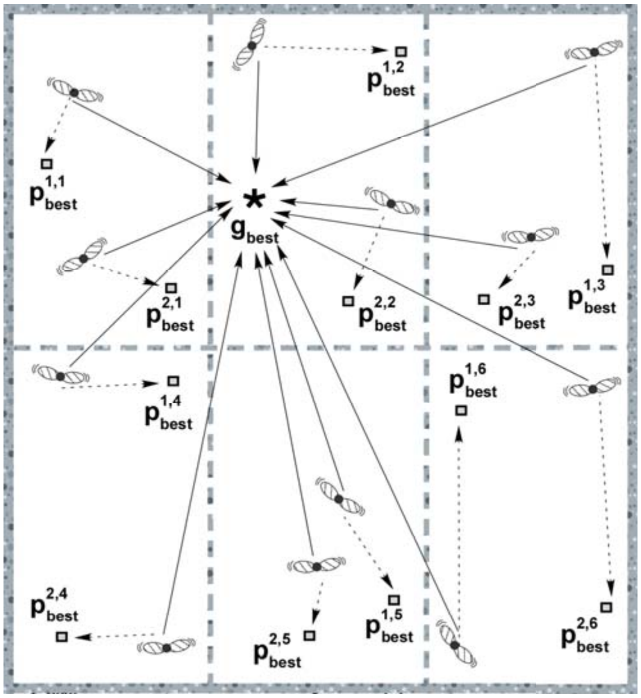


Fig.4 Dynamics of a change: all the agents gain the information about the global best as soon as the barriers imposed by the sub-boundaries are removed. They are attracted both by that location as well as by the own local best previously found.

In particular, the following functions have been considered. The first type is the Rastrigin function defined as:

$$f_1(x) = \sum_{i=1}^{N}\left(x_i^2 - 10\cos\left(2\pi x_i\right) + 10\right), \qquad (4)$$

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

$$f_2(x) = \frac{1}{4000}\sum_{i=1}^{N} x_i^2 - \prod_{i=1}^{N}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 . \qquad (5)$$

Last function is the Rosenbrock function

$$f_3(x) = \sum_{i=1}^{N}\left(100\left(x_{i+1} - x_i^2\right)^2 + \left(x_i - 1\right)^2\right), \qquad (6)$$

with (-50 < x_i < 50). All the above 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 modified 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, and 8) and, hence, different number of groups (note that, as previously mentioned, each sector contains only one group).

Set *i*=1
for *g*=1, *number_of_groups*
  for *k*=1, *number_of_agents_in_the_group*
    for *n*=1, *number_of_dimensions*
      Random initialization of $x_n^{k,g}(i)$ within therange of subdomain #*g*
      Random initialization of $v_n^{k,g}(i)$ 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, *number_of_groups*
    for *k*=1, *number_of_agents_in_the_group*
      Evaluate $fitness^{k,g}(i)$, the fitness of agent *k* in group *g* at instant *i*
    next *k*
  next *g*
  for *g*=1, *number_of_groups*
    Rank the fitness values of all agents included only in group *g*
  next *g*
  for *g*=1, *number_of_groups*
    for *k*=1, *number_of_agents_in_the_group*
      if $fitness^{k,g}(i)$ is the best value ever found by agent *k* in group *g* **then**
        $p_{best,n}^{k,g}(i) = x_n^{k,g}(i)$
      end if
      if $fitness^{k,g}(i)$ is the best value ever found by all agents **then**
        $g_{best,n}^{g}(i) = x_n^{k,g}(i)$
      end if
    next *k*
  next *g*
  i=*i*+1
  if (i >= *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 $g_{best,n}^{g}(i)$ and set the actual $g_{best,n}(i)$
else
    follow, with the actual value of *i*, the procedure showed in Fig.1
end if

Fig. 5 The modified PSO pseudocode. During the preliminary iterations the agents seek together, organized in groups, in a delimited territory defined by the sub-boundaries. After this stage, they are free to move all over the domain.

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 has to be considered a limit that should not be exceeded, so as not to lose the efficient cooperation of all the agents acting together, which is one of the most important features of the PSO algorithm. The results for *N*=3 are shown in Table II. 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 brackets is the number of fitness evaluations which indicates the calls to the solver. For the case of 20 agents and 8 sectors we have not proceeded since it is not possible to create 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 more rapidly the convergence, but also that the more we increase the number of divisions the less we improve performance. This fact suggests a logical conclusion, *viz.*, that there is a limit to the improvement even if we subdivide more and more. 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.

TABLE II
RESULTS OBTAINED BY USING SUB-BOUNDARIES INITIALIZATION IN SOLVING BENCHMARK FUNCTIONS

Rastrigin function

Population size	Boundaries	Number of groups: 2	Number of groups: 4	Number of groups: 8
16 agents	N.R.	142 (2272)	117 (1872)	98 (1568)
20 agents	N.R.	110 (2200)	70 (1400)	–
32 agents	N.R.	60 (1920)	40 (1280)	34 (1088)

Griewanck function

Population size	Boundaries	Number of groups: 2	Number of groups: 4	Number of groups: 8
16 agents	122 (1952)	110 (1760)	93 (1488)	80 (1280)
20 agents	96 (1920)	84 (1680)	74 (1480)	–
32 agents	80 (2560)	52 (1664)	45 (1440)	38 (1216)

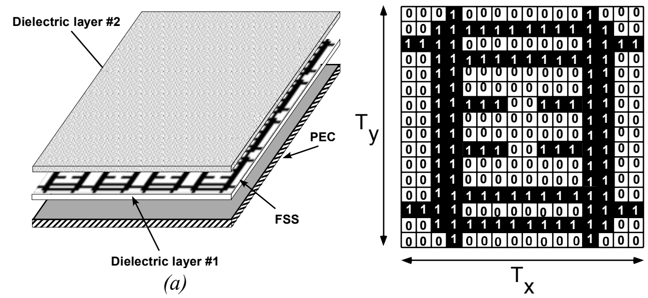
Rosenbrock function

Population size	Boundaries	Number of groups: 2	Number of groups: 4	Number of groups: 8
16 agents	44 (704)	31 (496)	18 (288)	15 (240)
20 agents	30 (600)	23 (460)	12 (240)	–
32 agents	19 (608)	14 (448)	7 (224)	7 (224)

IV. DESIGN OF ARTIFICIAL MAGNETIC CONDUCTORS

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 [29-32]. 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 high impedance Frequency Selective Surfaces (FSSs) into the design [30]. 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 behaviour at the desired frequency.

Our PSO code has been employed to optimize all these parameters but, since we have to deal not only with real parameters, such as, for instance, the unit cell dimensions (T_x and T_y) and the characteristics of the dielectric substrates (permittivity and thickness), but also with binary ones (Fig. 6(b)), we have implemented a PSO algorithm which can handle both the real and binary parameters.



(a) (b)

Fig.6 Optimization of an AMC:(a) geometrical configuration; (b) the binary codification of the unit cell ('1' represents PEC while '0' means absence of conducting material)

The solver adopted is a MoM code which employs rooftop basis functions that are both reliable and efficient [30,32]. A quantity proportional to the root mean square of the difference between the actual electric field reflection coefficient (Γ_E) and the desired one ($\mathrm{Re}\{\Gamma_{AMC}\}=1$, $\mathrm{Im}\{\Gamma_{AMC}\}=0$), for both TE and TM modes, is used to evaluate the performance of the structure, *i. e.*, the 'goodness' of the proposed solution. 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. 7.

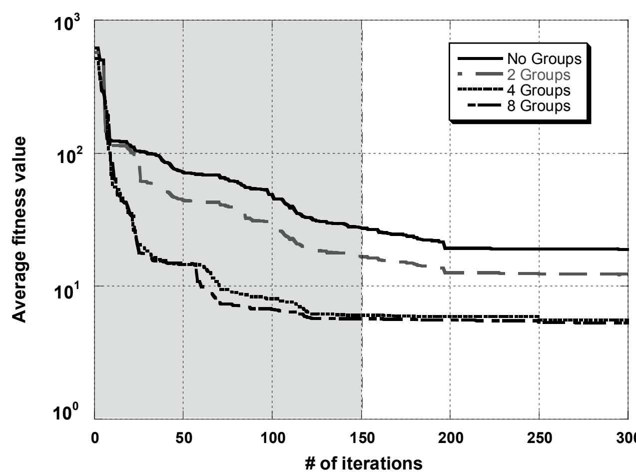
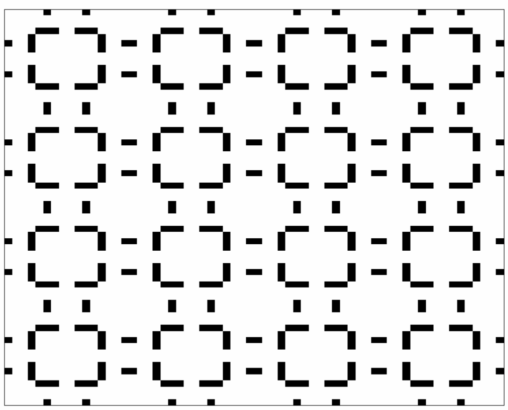
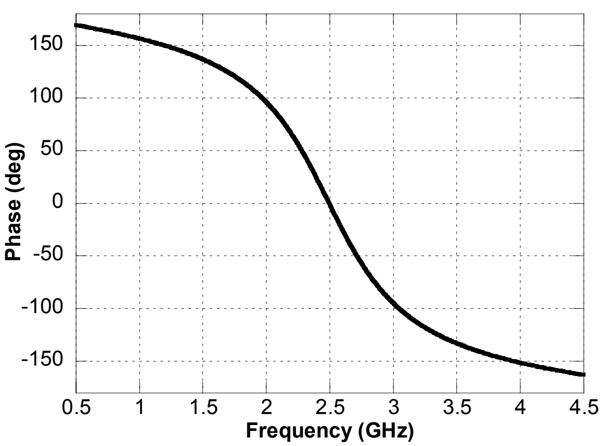


Fig.7 Comparison between the convergence rates for four different cases. The grey zone represents the part of the iteration run by using sub-boundaries (except for the no-groups case).

We note that there is an improvement in the performance 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, in this case, the sub-boundary approach is not applied to the binary map, hence it reduces the impact of further subdivisions of the domain. One of the results obtained with a swarm initialized by using four groups is shown in Fig.8.



(a)



(b)

Fig.8 Results of the enhanced PSO for an 8-group initialization. (a) the complete FSS screen. Dielectric characteristic: superstrate 0.4 cm thick with dielectric constant equal to 14.1, substrate 0.45 cm thick with a dielectric constant of 6.0); (b) phase of the frequency response.

V. Conclusion

A new implementation of the PSO has been described. The paper illustrated the importance of a uniform initialization of the particles within the multidimensional search space. This new approach suggests a division of the domain into different sector where the agents are initially placed, and where they move during the preliminary iterations. The sub-boundaries are removed, after this period, and the agents are able to fly everywhere inside the domain. The information about the best location found during the initialization is shared among the agents which preserve the memory of their own local best.

Some representative numerical examples have been presented to demonstrate the performance enhancement realized by using the proposed technique. Finally, the new PSO code has been successfully applied to the design of AMCs.

References

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

[2] W.G. Macready, and D.H. Wolpert, "The No Free Lunch theorems", *IEEE Trans. on Evol. Comput.*, vol.1, no. 1, pp. 67-82, April 1997.

[3] M. M. Khodier, and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization", *IEEE Trans. Antennas Propag.*, vol. 53, no 8, pp. 2674-2679, August 2005.

[4] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna", in *Proc. IEEE Int. Symp. Antennas Propagation*, vol.1, San Antonio, TX, pp. 314-317, 2002.

[5] N. Jin, and Y. Rahmat-Samii, "Particle swarm optimization of miniaturized quadrature reflection phase structure for low-profile antennas applications", in *Proc. IEEE Int. Symp. Antennas Propagation*, vol. 2B, Washington, DC, pp. 255-258, 2005.

[6] S. Baskar, A. Alphones, and P. N. Suganthan, "Concurrent PSO and FDR-PSO based reconfigurable phase-differentiated antenna array design", *in Proc. Evolutionary Computation (CEC 04)*, vol. 2, pp. 2173-2179, June 2004.

[7] S. Baskar, A. Alphones, P. N. Suganthan and J. J. Liang, "Design of Yagi-Uda antennas using comprehensive learning particle swarm optimization', *IEE Proc. Microwaves, Antennas and Propag.*, vol. 152, Issue 5, pp. 340-346, October 2005.

[8] D. Gies, and Y. Rahmat-Samii, "Particle swarm optimization for reconfigurable phase differentiated array design", *Microw. Opt. Technol. Lett.*, vol. 38, Issue 3, pp. 168-175, 2003.

[9] D. W. Boeringer, and D. H. Werner, "Efficiency-constrained particle swarm optimization of a modified Bernstein polynomial for conformal array excitation amplitude synthesis", *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2662-2673, August 2005.

[10] D. W. Boeringer, and D. H. Werner, "Particle swarm optimization versus genetic algorithm for a phased array synthesis", *IEEE Trans. Antennas Propag.*, vol. 52, Issue 3, pp. 771-779, March 2004.

[11] S. Cui, and D. S. Weile, "Application of a parallel particle swarm optimization scheme to the design of electromagnetic absorbers", *IEEE Trans. Antennas Propag.*, vol. 53, no. 11, pp. 3616-3624, 2005.

[12] L. Matekovits, M. Mussetta, P. Pirinoli, S. Selleri, and R. E. Zich, “Particle swarm optimization of microwave microstrip filters,” *in 2004 IEEE Antennas Propagation Soc. Int. Symp. Dig.*, vol. 3, pp. 2731–2734.

[13] M. P. Wachowiak, R. Smolikova, Y. Zheng, J. M. Zurada, and A. S. Elmaghraby, “An approach to multimodal biomedical image registration utilizing particle swarm optimization”, *IEEE Trans. Evol. Comput.*, vol. 8, pp289-301, 2004.

[14] M. Donelli, and A. Massa, “Computational approach based on particle swarm optimizer for microwave imaging of two-dimensional dielectric scatterers”, *IEEE Trans. Microwave and Tech.*, vol. 53, no. 5, May 2005.

[15] M. Clerc and J. Kennedy, “The particle swarm-explosion, stability, and convergence in a multidimensional complex space”, *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 58-73, Feb.2002.

[16] Y. Shi, and R. C. Eberhart, “Fuzzy adaptive particle swarm optimization”, *in Proc. Evolutionary Computation (CEC 01)*, vol. 1, pp. 101-106, May 2001.

[17] K. Yasuda, and N. Iwasaki, “Adaptive particle swarm optimization using velocity information of swarm”, *in Proc. Int. Conf. Systems, Man and Cyber. ‘04*, vol. 4, pp. 3475-3481, October 2004.

[18] Y. Shi, and R. C. Eberhart, “Empirical study of particle swarm optimization”, *in Proc. Evolutionary Computation (CEC 99)*, vol. 3, pp. 1945–1950, 1999.

[19] H. Zhenya, W. Chengjian, Y. Luxi, G. Xiqi, Y. Susu, R. C. Eberhart, and Y. Shi, “Extracting rules from fuzzy neural network by particle swarm optimization”, *in Proc. Evolutionary Computation (CEC 98)*, pp. 74–77, 1998.

[20] M. Clerc, “The swarm and the queen: towards a deterministic and adaptive particle swarm optimization”, *in Proc. Evolutionary Computation (CEC 99)*, vol. 3, pp. 1951-1957, July 1999.

[21] J. Kennedy, “Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance”, *in Proc. Evolutionary Computation (CEC 99)*, vol.3, pp. 1931-1938, July 1999.

[22] M. Lovbjerg, T. K. Rasmussen, and T. Krink, “Hybrid particle swarm optimizer with breeding and subpopulations”, *in Proc. 3rd Genetic and Evol. Comp. Conf. GECCO-2001,* 2001.

[23] P. N. Sugantan, “Particle swarm optimizer with neighborhood operator’, *in Proc. Evolutionary Computation (CEC 99),* Washington DC, pp. 1958-1962, 1999.

[24] R. Mendes, P. Cortez, M. Rocha, and J. Neves, “Particle swarms for feed-forward neural network training”, *in Proc. Int. Joint Conf. Neural Networks (IJCNN ‘02)*, vol. 2, pp. 1895-1899, 2002.

[25] A. M. Abdelbar, S. Abdelshahid, and D.C. Wunsch, “Fuzzy PSO: a generalization of particle swarm optimization”, *in Proc. IEEE Int. Joint Conf. Neural Networks (IJCNN ’05)*, vol. 2, pp. 1086-1091, August 2005.

[26] N. Jin, and Y. Rahmat-Samii, “Parallel particle swarm optimization and finite difference time-domain (PSO/FDTD) algorithm for multiband and wide-band patch antenna designs’, *IEEE Trans. Antennas Propag.*, vol. 53, no. 11, November 2005.

[27] J. Robinson, and Y. Rahmat-Samii, “Particle swarm optimization in electromagnetics”, *IEEE Trans. Antennas Propag.*, vol.52, no.2, pp. 397-407, Feb. 2002.

[28] T. Huang, and A. S. Mohan, “A hybrid boundary condition for robust particle swarm optimization”, *Antennas and Wireless Prop. Letters*, vol. 4, pp. 112-117, 2005.

[29] D. Sievenpiper, L. Zhang, R. Broas, N. Alexopolous, and E. Yablonovitch, “High-impedance frequency selective surfaces with a forbidden frequency band”, *IEEE Trans. Microwave Theory Tech*., vol.47, no. 11, pp. 2059-2074, Nov. 1999.

[30] A. Monorchio, G. Manara, L. Lanuzza, "Synthesis of Artificial Magnetic Conductors by Using Multilayered Frequency Selective Surfaces," *IEEE Antennas and Wireless Propagation Letters*, Vol. 1, No. 11, pp. 196-199, 2002.

[31] Z. Ying, J. von Hagen, M. Younis, C. Fischer, W. Wiesbeck, “Planar Artificial Magnetic Conductor and Patch Antennas”, *IEEE Trans. Antennas Prop*., vol. 51, issue 10, Oct. 2003, pp 2704-2712.

[32] D. J. Kern, D. Werner, A. Monorchio, L. Lanuzza, M. Wilhelm, “The Design Synthesis of Multiband Artificial Magnetic Conductors Using High Impedance Frequency Selective Surfaces”, *IEEE Trans. On Antennas Propag*., vol.53, no.1, January 2005.