Particle swarm optimization algorithms applied to antenna and microwave design problems

Sotirios Goudos
Aristotle University of Thessaloniki

Zaharias D. Zaharis
Aristotle University of Thessaloniki

Swarm Intelligence for Electric and Electronic Engineering

Girolamo Fornarelli
Politecnico di Bari, Italy

Luciano Mescia
Politecnico di Bari, Italy

ENGINEERING SCIENCE REFERENCE

Managing Director: Lindsay Johnston
Editorial Director: Joel Gamon
Book Production Manager: Jennifer Romanchak
Publishing Systems Analyst: Adrienne Freeland
Development Editor: Myla Merkel
Assistant Acquisitions Editor: Kayla Wolfe
Typesetter: Christina Henning
Cover Design: Nick Newcomer

Published in the United States of America by
Engineering Science Reference (an imprint of IGI Global)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@igi-global.com
Web site: http://www.igi-global.com

Library of Congress Cataloging-in-Publication Data

Swarm intelligence for electric and electronic energy / Girolamo Fornarelli and Luciano Mescia, editors.
p. cm.
Includes bibliographical references and index.
Summary: "This book provides an exchange of knowledge on the advances, discoveries and improvements of swarm intelligence in electric and electronic engineering, bringing together new swarm-based algorithms and their effect on complex problems and various real-world applications"--Provided by publisher.
ISBN 978-1-4666-2666-9 (hardcover : alk. paper) -- ISBN 978-1-4666-2697-3 (ebook : alk. paper) -- ISBN 978-1-4666-2728-4 (pr : alk. paper) 1. Electrical engineering--Mathematics. 2. Electronics--Mathematics. 3. Swarm intelligence. I. Fornarelli, Girolamo, 1970- II. Mescia, Luciano, 1973-
TK153.S953 2013
621.30285'63--dc23
2012029074

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The views expressed in this book are those of the authors, but not necessarily of the publisher.

Table of Contents

Section 2
Antenna and Optical

Section 3
Control Optimization

Chapter 6
Particle Swarm Optimization Algorithms Applied to Antenna and Microwave Design Problems

Sotirios K. Goudos
Aristotle University of Thessaloniki, Greece

Zaharias D. Zaharis
Aristotle University of Thessaloniki, Greece

Konstantinos B. Baltzis
Aristotle University of Thessaloniki, Greece

ABSTRACT

Particle Swarm Optimization (PSO) is an evolutionary optimization algorithm inspired by the social behavior of birds flocking and fish schooling. Numerous PSO variants have been proposed in the literature for addressing different problem types. In this chapter, the authors apply different PSO variants to common antenna and microwave design problems. The Inertia Weight PSO (IWPSO), the Constriction Factor PSO (CFPSO), and the Comprehensive Learning Particle Swarm Optimization (CLPSO) algorithms are applied to real-valued optimization problems. Correspondingly, discrete PSO optimizers such as the binary PSO (binPSO) and the Boolean PSO with velocity mutation (BPSO-vm) are used to solve discrete-valued optimization problems. In case of a multi-objective optimization problem, the authors apply two multi-objective PSO variants. Namely, these are the Multi-Objective PSO (MOPSO) and the Multi-Objective PSO with Fitness Sharing (MOPSO-fs) algorithms. The design examples presented here include microwave absorber design, linear array synthesis, patch antenna design, and dual-band base station antenna optimization. The conclusion and a discussion on future trends complete the chapter.

DOI: 10.4018/978-1-4666-2666-9.ch006

INTRODUCTION

In the past decade, several evolutionary algorithms (EAs) that mimic the behavior and evolution of biological entities emerged. Among others, Particle Swarm Optimization (PSO) is a popular evolutionary algorithm which is based on the intelligence and movement of swarms (birds, fishes, bees, etc.) and resembles their behavior (Kennedy & Eberhart, 1995).

Many similarities exist between PSO and other evolutionary computation techniques such as Genetic Algorithms (GAs). In general, PSO does not have any evolution operators like crossover and mutation. Compared to Genetic Algorithms, PSO has fewer parameters to adjust and is easier to implement in any programming language. PSO is also computationally more efficient than a GA with the same population size. The algorithm has been successfully applied in many engineering disciplines: function optimization, artificial neural network training, fuzzy system control and other areas where GAs are also applied. The fact that particle swarm optimizers can handle efficiently arbitrary optimization problems has also made them popular for solving problems in electromagnetics, especially in electromagnetic design ones (Baskar, Alphones, Suganthan, & Liang, 2005; Deligkaris et al., 2009; Goudos, Moysiadou, Samaras, Siakavara, & Sahalos, 2010; Goudos, Rekanos, & Sahalos, 2008; Goudos & Sahalos, 2006; Goudos, Zaharis, Kampitaki, Rekanos, & Hilas, 2009; Khodier & Christodoulou, 2005; Robinson & Rahmat-Samii, 2004; Zaharis, 2008; Zaharis, Kampitaki, Lazaridis, Papastergiou, & Gallion, 2007). Numerous different PSO variants exist in the literature. The most common algorithms include the classical Inertia Weight PSO (IWPSO) and the Constriction Factor PSO (CFPSO) (Clerc, 1999). However, in order to further improve the performance of PSO on complex multimodal problems, a PSO variant was proposed (Liang, Qin, Suganthan, & Baskar, 2006). This variant is the Comprehensive Learning Particle Swarm Optimizer (CLPSO) which utilizes a new learning strategy. The CLPSO algorithm accelerates the convergence of the classical PSO. It has been applied successfully to Yagi-Uda antenna design by Baskar et al. (2005) and to linear array synthesis by Goudos et al. (2010).

The PSO algorithm is inherently used only for real-valued problems but can easily expand to discrete-valued problems. A simple modification of the real-valued PSO called binary PSO (binPSO) has been presented by Kennedy & Eberhart (1997). In (Marandi, Afshinmanesh, Shahabadi, & Bahrami, 2006) the Boolean PSO is introduced and applied to dual-band planar antenna design. The Boolean PSO is based on the idea of using exclusively Boolean update expressions in the binary space. An extension to Boolean PSO, that improves the algorithm performance, is the Boolean PSO with velocity mutation (BPSO-vm) which has been applied successfully to patch antenna design (Deligkaris et al., 2009).

Multi-objective extensions of PSO include the Multi-objective Particle Swarm Optimization (MOPSO) (Coello Coello, Pulido, & Lechuga, 2004) and the Multi-objective Particle Swarm Optimization with fitness sharing (MOPSO-fs) (Salazar-Lechuga & Rowe, 2005). The above algorithms have also been used in antenna and microwave design problems (Goudos & Sahalos, 2006; Goudos et al., 2009). The purpose of this chapter is to briefly describe the aforementioned algorithms and present their application to antenna and microwave design problems. Within this context, we present results from design cases using the IWPSO, CFPSO, CLPSO, binPSO, BPSO-vm and the multi-objective variants MOPSO and MOPSO-fs. The examples comprise the design of high-absorption planar multilayer coatings, the synthesis of unequally spaced linear arrays with sidelobe level (*SLL*) suppression under mainlobe beamwidth and null control constraints, the design of thinned planar microstrip arrays under constraints of impedance matching, low *SLL* and null control, and finally the optimization of dual-band base station antennas for wireless networks. Comparisons with optimization methods in the literature validate the efficacy of PSO. Our study completes with a brief discussion of future directions and perspectives in the area and it is supported with an adequate number of references.

The chapter is subdivided into four sections. First, we present the different PSO algorithms. In the next Section, we describe the design cases and present the numerical results. An outline of future research directions is provided in the following Section while in the "Conclusion" Section we conclude the chapter and discuss the advantages of using a PSO-based approach in the design and optimization of microwave systems and antennas. Finally, an "Additional Reading Section" gives a list of readings to provide the interested reader with useful sources in the field.

PARTICLE SWARM OPTIMIZATION: CLASSICAL ALGORITHMS AND VARIANTS

Particle swarm optimization is based on the intelligence and movement of swarms. In the nature, the swarm intelligence helps the swarm to find the place where there is more food than anywhere else. For example, let us consider a swarm of bees. The bees search for the place with the highest density of flowers. In general, the direction and speed of their motion are random quantities. However, each bee remembers the places where it found the highest concentration of flowers and it also knows the places where the rest of the bees have found plenty of flowers. As a result, and despite the random characteristics of their flights, the bees take into account the best positions encountering by them and the rest of the bees and thus make an attempt to balance exploration and exploitation. This behavior finally leads the bees to the place with the highest density of flowers. A similar approach is followed by the PSO method.

In the PSO algorithm, each individual in the swarm is called "particle" or "agent" and moves in an *N*-dimensional space trying to find an even better position by adjusting its direction. It learns from its own experience and the experiences of the neighboring particles and updates its velocity based on them. The position and velocity of the *i*th particle is respectively, $\overline{x}_i = \left(x_{i1}, x_{i2}, \ldots, x_{iN}\right)$ and $\overline{v}_i = \left(v_{i1}, v_{i2}, \ldots, v_{iN}\right)$, $i = 1, \ldots, M$ and $n = 1, \ldots, N$, where M is the "population size", i.e., the number of the particles that compose the swarm. The particle positions may be limited in the respective (*n*th) dimension between an upper and a lower boundary (U_n and L_n, respectively), i.e., $x_{in} \in \left[L_n, U_n\right]$, restricting the search space within these limits. The difference $R_n = U_n - L_n$ is the "dynamic range" of the *n*th dimension.

In general, the system is initialized and searches for optima by updating the particles positions in any iteration. Each particle position is updated by finding two optimum values, the best position, $\overline{p}_i = \left(p_{i1}, \ldots, p_{iN}\right)$, achieved so far by the particle (*pbest* position) and the best position, $\overline{g} = \left(g_1, \ldots, g_N\right)$, obtained so far by any particle of the swarm (*gbest* position). By taking into account these two values respectively as cognitive and social information, each particle updates its position and velocity. The algorithm is executed repeatedly until a specified number of iterations is reached or the velocity updates are close to zero. The particle quality is measured according to a predefined mathematical function $F\left(\overline{x}_i\right)$ called "fitness function", which is related to the problem to be solved and reflects the optimality of a particular solution. In each iteration, the change of $\overline{x}_i$ is $\Delta\overline{x}_i = \overline{v}_i \Delta t$, where Δt is the time interval. Thus, by setting $\Delta t = 1$ the new position of the *i*th particle is

$$\overline{x}_i^t = \overline{x}_i^{t-1} + \overline{v}_i^t \tag{1}$$

where $\overline{x}_i^t$ and $\overline{x}_i^{t-1}$ denote the particle position at the current and the previous iteration, respectively, and $\overline{v}_i^t$ is the velocity at the current iteration.

At this point, we have to notice that $\overline{g}$ corresponds to the maximum fitness value found so far by the swarm. The *gbest* neighborhood is equivalent to a fully connected social network in which every individual is able to compare the performance of every other member of the population, imitating the very best. In this case, every particle is attracted to the *gbest* position. In a similar way, we may consider that each (*i*th) particle is affected by the best performance of K_i neighbors. The last are particles near the individual in a topological rather than in the parameter space. In this case, the best position is called the "$\ell best$ position" and it is represented as $\overline{\ell}_i = \left(\ell_{i1}, \ldots, \ell_{iN}\right)$. Equivalently, in the first approach, the neighborhood of a particle is the whole swarm, while in the second one it consists of a specific number of certain particles. The optimal connectivity pattern between individuals depends on the problem to be solved. In general, the *gbest* approach convergences faster but it is more susceptible to convergence on local optima. This behavior is similar to GAs remaining stagnant to local optima.

Classical PSO Algorithms

As mentioned above, in particle swarm optimization, individuals are influenced by both their own previous behavior and the successes of their neighbors. In the IWPSO algorithm, the particle's velocity depends on its previous velocity value, on the distance between the particle's position and the position *pbest*, and on the distance between the particle's position and *gbest* or $\ell best$ position depending on the neighborhood model applied by the user. In the following analysis, we will refer only to the *gbest* case; the extension to the $\ell best$ case is straightforward.

In the IWPSO algorithm, the *i*th particle's velocity in the *n*th dimension is updated as in Equation 2 (see Box 1) where *w* is a positive parameter called "inertia weight", c_1 and c_2 are positive parameters known respectively as "cognitive coefficient" and "social coefficient", and rnd_{in1}^t, rnd_{in2}^t are random numbers uniformly distributed in the interval from 0 to 1. The index *t* denotes the current iteration, while *t*–1 denotes the previous one. The parameter *w* controls the impact of previous velocity values on the current velocity and usually has fixed values between 0.0 and 1.0. Large values of *w* facilitate global exploration, while a smaller w tends to facilitate local exploration to fine-tune the current search area. A proper choice of *w* achieves a balance between global and local exploration and results in faster convergence (Shi & Eberhart, 1998). A linear decrease of *w* from 0.9 to 0.4 during the simulation is usually employed (Shi & Eberhart, 1999). It has to be noticed, that the same value of *w* is used for all dimensions of all particles in a given population. The parameter c_1 represents the impact of the particle's distance from its best position, while c_2 determines the influence of the swarm. Usually, the two parameters are both set equal to 2.0 (Kennedy & Eberhart, 1995) or 1.49 (Robinson & Rahmat-Samii, 2004).

The stochastic changes in the velocity of the particles may result in an expansion of a particle's trajectory into wider cycles through the problem space. A solution to this problem is to set a

Box 1.

$$v_{in}^t = w \cdot v_{in}^{t-1} + c_1 \cdot rnd_{in1}^t \cdot \left(p_{in}^t - x_{in}^{t-1}\right) + c_2 \cdot rnd_{in2}^t \cdot \left(g_n^t - x_{in}^{t-1}\right) \tag{2}$$

Box 2.

$$v_{in}^{t} = K \cdot \left[v_{in}^{t-1} + \varphi_1 \cdot rnd_{in1}^{t} \cdot \left(p_{in}^{t} - x_{in}^{t-1} \right) + \varphi_2 \cdot rnd_{in2}^{t} \cdot \left(g_{n}^{t} - x_{in}^{t-1} \right) \right] \tag{3}$$

maximum allowed velocity $\overline{v}_{\max} = \left(v_{\max,1}, \ldots, v_{\max,N} \right)$. Thus, we set $v_{in} \equiv v_{\max,n}$, $\forall i,n: \; v_{in} > v_{\max,n}$ and $v_{in} \equiv -v_{\max,n}$ $\forall i,n: \; v_{in} < -v_{\max,n}$. As a result, we prevent explosion and scale the exploration of the particle. The choice of $\overline{v}_{\max}$ differs from problem to problem. In general, a small maximum velocity limits the global exploration. On the contrary, large $v_{\max,n}$ values may reduce the local exploration ability (Shi & Eberhart, 1998). If a step larger than $\overline{v}_{\max}$ is required to escape a local optimum then the particle will be trapped; on the other hand, it is better to take smaller steps as the solution approaches to an optimum. A good choice is to set each $v_{\max,n}$ coordinate around 10-20% of the dynamic range of the respective dimension when $w = 1$ and equal to this range if $w < 1$ (Eberhart & Shi, 2001). Another option as suggested in (Jin & Rahmat-Samii, 2007) is to set its value equal to the dynamic range in each dimension of the particle. For example, if a variable is allowed to be optimized within $\left(L_n, U_n \right)$, the maximum velocity in this dimension is $\left| L_n - U_n \right|$ in both directions.

In another classical PSO algorithm, the CFPSO, a different velocity update rule is proposed. In that case, the velocity components are updated as in Equation 3 (Box 2) where *K* is the "constriction coefficient" defined as:

$$K \equiv \frac{2}{\varphi - 2 + \sqrt{\varphi^2 - 4\varphi}}, \quad \varphi \geq 4 \tag{4}$$

The parameters $\varphi_{1,2}$ have a similar meaning to $c_{1,2}$ and the parameter φ, known as "acceleration constant", is their sum. A good choice for both φ_1 and φ_2 is 2.05 (Eberhart & Shi, 2001). The introduction of the constriction coefficient eliminates the need of $\overline{v}_{\max}$. However, it has been concluded from (Eberhart & Shi, 2000) that it is still better to use $\overline{v}_{\max}$ and to set it equal to the dynamic range of each variable on each dimension.

Nevertheless, the parameters *w*, *K* and $\overline{v}_{\max}$ are not always able to confine the particles within the search space. In order to solve this problem, three boundary conditions have been suggested: (i) The absorbing walls condition: When x_{in} becomes greater than U_n or less than L_n, the respective velocity component becomes zero and the *i*th particle is pulled back toward the search space, i.e., if $x_{in} > U_n$ then $x_{in} = U_n$ and $v_{in} = 0$, and also if $x_{in} < L_n$ then $x_{in} = L_n$ and $v_{in} = 0$. (ii) The reflecting walls condition: When x_{in} becomes greater than U_n or less than L_n, v_{in} is set to $-v_{in}$, and the particle is reflected back towards the search space. (c) The invisible walls condition: The particles are allowed to move outside the search space without any restriction but a large predefined fitness value is assigned to them. This condition saves computational time, because the fitness function is calculated only for the particles inside the search space.

Comprehensive Learning Particle Swarm Optimizer (CLPSO)

In order to improve the convergence speed of the classical PSO algorithms, a different learning strategy has been proposed. This strategy is involved in the CLPSO algorithm and ensures that the diversity of the swarm is preserved in order to discourage premature convergence. To achieve this, the algorithm adopts a velocity update rule that considers all particles' previous experiences, i.e., best solutions, as a potential during the calculation of the particle's new velocity. As a result, a particle can learn from a different exemplar in each dimension.

In the CLPSO algorithm, the velocity update equation is given below:

$$v_{in}^{t} = w \cdot v_{in}^{t-1} + c \cdot rnd_{in}^{t} \cdot \left(p_{f_i(n)n}^{t} - x_{in}^{t-1}\right) \quad (5)$$

where the $f_i = \left[f_i(1), \ldots, f_i(N)\right]$ defines the particle's *pbest* that should by followed by the *i*th particle and the $p_{f_i(n)n}^{t}$ is the corresponding dimension of all particles' *pbest*. In practice, a random number is generated for each dimension; this number is compared with a parameter called "learning probability" that takes different values for different particles. On the basis of the previous comparison, a decision whether the particle will learn in the corresponding dimension from its own *pbest* (if the random number is larger than the learning probability) or from another particle's *pbest* is made. In the second case, we first choose two particles of the population using a uniform random distribution (obviously, we exclude the particle whose velocity is to be updated). Next, we compare the fitness values of these two particles *pbest's* and select the best one. The selected particle's *pbest* is used as the exemplar to learn for that dimension. If all the exemplars of a particle are its own *pbest*, then we randomly choose one dimension, in order to learn from another particle's *pbest*. The updating strategy used in CLPSO provides a larger potential to explore more of the search space (it searches more promising regions to find the global optimum) than classical PSO algorithms. Obviously, the diversity is also increased with the particles' potential search. As it is reported in Liang et al. (2006) the improved learning strategy has enabled CLPSO to make use of the information in the swarm more effectively and to generate frequently better quality solutions when compared to eight PSO variants on numerical test problems.

The CLPSO algorithm can be easily implemented. However, its drawbacks compared to the classical particle swarm optimizers are the increased complexity and the higher computational load.

Discrete PSO Variants: The binPSO and BPSO-vm Algorithms

PSO algorithm is inherently used only for real-valued problems. However, in many real-world applications the variables to be optimized take integer values. Several PSO variants for discrete-valued problems exist in the literature (Chen et al., 2010; Modiri & Kiasaleh, 2011a; Sharaf & El-Gammal 2009; Tao, Chang, Yi, Gu, & Li, 2010). In the following paragraphs, we present two options to expand PSO for discrete-valued problems. These popular discrete PSO variants are the binary PSO (binPSO) and the Boolean PSO with velocity mutation (BPSO-vm). The binPSO algorithm is a simple modification of IWPSO that allows the algorithm to operate using binary representations. In binPSO, the location of each particle is a binary string with length *D* while the particle velocities remain real-valued and are updated as in IWPSO. The *i*th coordinate of each particle's position is a bit updated as

$$x_i^{t+1} = \begin{cases} 1, & \text{if } \rho < s(v_i^{t+1}) \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where ρ is random uniformly distributed number within the interval [0, 1] and $s(\cdot)$ is a sigmoid function that maps all the real values of velocity to the range $[0,1]$. Such a function is given below:

$$s(x) = \frac{1}{1+e^{-x}} \tag{7}$$

Another discrete PSO algorithm is the Boolean PSO (BPSO) (Marandi et al., 2006). The BPSO is based on the idea of using exclusively Boolean expressions to update both the velocity and the position of a particle. These expressions are given below:

$$v_{id}^{t} = w \bullet v_{id}^{t-1} + c_1 \bullet \left(p_{id}^{t} \oplus x_{id}^{t-1}\right) + c_2 \bullet \left(g_d^{t} \oplus x_{id}^{t-1}\right) \tag{8}$$

$$x_{id}^{t} = x_{id}^{t-1} \oplus v_{id}^{t} \tag{9}$$

where $d = 1,\ldots,D$, g_d is the dth bit of the position $\overline{g}$, while the parameters w, c_1 and c_2 are random bits chosen with probability of being "1" respectively equal to W, C_1 and C_2, which are user-selected parameters. Also the notations (●), (Å) and (+) represent respectively the *and*, *xor* and *or* Boolean operators.

In order to control the convergence speed of the optimization process, the algorithm sets a maximum allowed velocity $v_{\max}$ defined as the maximum allowed number of '1's in the velocity vectors $\overline{v}_i = [v_{i1},\ldots,v_{iD}]$. The number of '1's in the binary vector $\overline{v}_i$ is the "velocity length" L_i. This parameter is controlled using "negative selection" (NS), a biological immunity process responsible for eliminating the T-cells that recognize self antigens in the thymus (Afshinmanesh, Marandi, & Rahimi-Kian, 2005). According to NS, if $L_i \le v_{\max}$, the $\overline{v}_i$ is a non-self antigen and remains the same. If $L_i > v_{\max}$, $\overline{v}_i$ is considered as a self antigen and thus the NS is applied to it by changing randomly chosen '1's into '0's until the condition $L_i = v_{\max}$ is met.

The BPSO-vm extends the BPSO by using a mutation operator applied to the particle velocities. In particular, the bits of $\overline{v}_i$ are allowed to change with probability m_r (mutation probability) from '0' to '1', but are not allowed to change from '1' to '0'. This mutation scheme diversifies the population and thus results in an increase of the exploration ability of the particles. It should be mentioned that the mutation probability must be relatively small to avoid producing a pure random-search algorithm.

The structure of the BPSO-vm algorithm is described by the following steps:

1. The population size, the dimension of the binary solution space, the maximum allowed velocity, the mutation probability and the total number of iterations are set.
2. The initial particle positions $\overline{x}_i$ and their corresponding velocities $\overline{v}_i$ are randomly generated.
3. The NS process is applied to correct the particle velocities $\overline{v}_i$.
4. The cost functions, $F(\overline{x}_i)$, and the minimum one among all, $F_{\min}$, are evaluated. The particle with the minimum cost function is considered as the global best, $\overline{g}$.
5. The initial positions of the particles are set as the individual *pbest* ones, i.e., $\overline{p}_i = \overline{x}_i$.
6. The particles velocities are updated using Equation 8.
7. The NS process is applied to correct the particle velocities.
8. Velocity mutation is applied to every '0' bit of $\overline{v}_i$.
9. The positions $\overline{x}_i$ are updated using Equation 9.
10. The cost functions, $F(\overline{x}_i)$, are evaluated.
11. The *pbest* and *gbest* positions, respectively $\overline{p}_i$ and $\overline{g}$, are updated.
12. The algorithm is repeated from step 6 until the total number of iterations is reached.

Multi-Objective PSO Extensions: The MOPSO and MOPSO-fs Methods

PSO seems to be suitable for multi-objective optimization due to its speedy convergence in single objective problems. Over the last few years, several Multi-Objective PSO algorithms have been proposed.

The MOPSO algorithm proposed in Coello Coello et al. (2004) has been validated against highly competitive evolutionary multi-objective algorithms like the Non-dominated Sorting Genetic Algorithm II (NSGA-II) (Deb, Pratap, Agarwal, & Meyarivan, 2002). The key elements of MOPSO are:

1. The historical record of the best solutions found by a particle is used to store non-dominated solutions generated in the past. Therefore, a repository is introduced and the positions of the particles that represent non-dominated solutions are stored. The parameter that has to be adjusted is the "repository size".
2. A new mutation operator is also introduced. This operator intends to produce a highly explorative behavior of the algorithm. The effect of mutation decreases as the number of iterations increase. The parameter of mutation probability is therefore used.
3. The idea of an adaptive grid is introduced. An external archive is used to store all the solutions that are non-dominated with respect to the contents of the archive. Into the archive the objective function space is divided into regions. The adaptive grid is a space formed by hypercubes. Each hypercube can be interpreted as a geographical region that contains a number of particles. The adaptive grid is used to distribute in a uniform way the largest possible amount of hypercubes. It is necessary therefore to provide the parameter of grid subdivisions.

As reported in Coello Coello et al. (2004), the MOPSO algorithm is relatively easy to implement. The exploratory capabilities of PSO are improved using the mutation operator. Additionally, the authors have found that MOPSO requires low computational cost, which is a key issue in cases of electromagnetic optimization.

An improvement of the previous variant, the MOPSO-fs utilizes both particle swarm optimization and fitness sharing. The last aims to spread the solutions along the Pareto front. Fitness sharing is used in the objective space and enables the algorithm to maintain diversity between solutions. This means that particles within highly populated areas in the objective space are less likely to be followed.

An external repository stores the non-dominated particles found. The best particles found in each iteration are inserted into the repository. The role of the repository is twofold: it helps the search for the next generations and it maintains a set of non-dominated solutions until the end of the process. This set of solutions forms the Pareto front. The structure of the MOPSO-fs algorithm follows.

We consider an M-size population of N-dimensional particles $\overline{x}^i,\ \ i = 1,\ldots,M$ (N is the number of design parameters). First, the algorithm initializes with random population from a uniform distribution. The external repository is filled with all the non-dominated particles. A fitness sharing value is calculated for each particle in the repository. A high value of f_{sh}^i suggests that the vicinity of the ith particle is not highly populated. The fitness sharing value for the ith particle is

$$f_{sh}^i = \frac{10}{\sum_{j=1}^{R_s} sharing^{i,j}} \tag{10}$$

where R_s is the number of the particles in the repository and

$$sharing^{i,j} = \begin{cases} 1 - \left(d^{i,j} / \sigma_{share} \right)^2, & \text{if } d^{i,j} < \sigma_{share} \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

with $d^{i,j}$ the Euclidean distance between the *i*th and the *j*th particle and σ_{share} the radius of the vicinity area of a particle. According to the fitness sharing principle, the particles that have more particles in their vicinity are less fit than those with fewer particles around it.

Provided that fitness sharing is assigned to each particle in the repository, some particles are chosen as leaders according to roulette wheel selection, i.e., the particles with higher levels of fitness are more likely to be selected. In the next iteration, the leaders are going to be followed by the rest of the particles. The velocity update rule for the *i*th particle shows similarities it is given by

$$\begin{aligned} v_{in}^{t} &= w \cdot v_{in}^{t-1} + c_1 \cdot rnd_{in1}^{t} \cdot \left(p_{in}^{t} - x_{in}^{t-1} \right) \\ &\quad + c_2 \cdot rnd_{in2}^{t} \cdot \left(g_{hn}^{t} - x_{in}^{t-1} \right) \end{aligned} \tag{12}$$

where p_{in}^{t} is the best position found by the *i*th particle so far, g_{hn}^{t} is the leader particle position along the *n*th dimension, and

$$x_{in} = \begin{cases} L_n + r_3 \cdot (U_n - L_n), & \text{if } \quad x_{in} < L_n \\ U_n - r_4 \cdot (U_n - L_n), & \text{if } \quad x_{in} > U_n \end{cases},$$

rnd_{in2}^{t} are uniformly distributed numbers within 0 and 1. The position of each particle is updated using Equation 1.

Next, the repository is updated with the current solutions found by the particles according to the dominance and fitness sharing value criteria. In particular, the particles that dominate those inside the repository are inserted while all dominated solutions are deleted. This operation allows the repository to be maintained as the Pareto front found so far. In case of a repository full of non-dominated particles, if a particle non-dominated by any in the repository is found then fitness sharing values are compared; if its value is better than the worst fitness sharing in the repository, it replaces the respective particle. The fitness sharing of all particles is updated when a particle is inserted in or deleted from the repository. Finally, the memory of each particle is updated according to the criterion of dominance, i.e., if the current particle position dominates the previous one in the particle's memory it replaces it.

This algorithm can be improved by adding a constraint handling mechanism that assigns to each particle a constraint violation number (CVN) which carries the sum of the constraint violations produced by the solution. Particles with smaller CVN dominate the ones with larger CVN while feasible solutions have zero CVN. Also, if the *i*th particle position in the *n*th dimension is found to be out of bounds, then it is updated as

$$x_{in} = \begin{cases} L_n + rnd_{in3} \cdot (U_n - L_n), & \text{if } \quad x_{in} < L_n \\ U_n - rnd_{in4} \cdot (U_n - L_n), & \text{if } \quad x_{in} > U_n \end{cases} \tag{13}$$

where rnd_{in3}, rnd_{in4} are uniformly distributed numbers within 0 and 1.

Control Parameter Selection

It must be pointed out that several PSO variants exist in the literature. In order to select, the best algorithm for every problem one has to consider the problem characteristics. For example, micro-PSO performs very well for microwave image reconstruction (Huang & Mohan, 2007). Another key issue is the selection of the algorithm control parameters, which is also in most cases problem-dependent. The control parameters selected here for these algorithms are those that commonly perform well regardless of the characteristics of the problem to be solved.

For real-coded GAs typical values are 0.9 for the crossover probability and 1/N for the mutation probability. The same values apply also for NSGA-II (Deb et al., 2002). For the binary-coded GA we set the crossover and mutation probabilities equal to 0.8 and 0.25, respectively.

In the PSO algorithms c_1 and c_2 are set equal to 2.05. For CFPSO, these values result in $K = 0.7298$. For IWPSO, it is common practice to linearly decrease the inertia weight starting from 0.95 to 0.4. The velocity is updated asynchronously, which means that the global best position is updated at the moment it is found. In CLPSO the parameter *c* in Equation 5 is set equal to one as it is suggested in Liang et al. (2006).

In BPSO, it is $W = 0.1$, $C_1 = C_2 = 0.5$ and $v_{\max} = 2$. In BPSO-vm, we additionally set $m_r = 0.02$.

The main characteristics of the MOPSO algorithm are; the repository size, the mutation operator and the grid subdivisions. The repository is the archive where the positions of the particles that represent non-dominated solutions are stored. Therefore the parameter that has to be adjusted is the repository size. This is usually set equal to the swarm size. MOPSO introduces a mutation operator that intends to produce a highly explorative behavior of the algorithm. The effect of mutation decreases as the number of iterations increase. A setting of 0.5 has been found suitable after experiments for mutation probability; similarly, a value of 30 has been found suitable for the parameter of grid divisions (Coello Coello et al., 2006). The above parameters are those selected for our problem given below for the MOPSO algorithm.

MOPSO-fs also uses a repository to store all the all the non-dominated solutions (Salazar-Lechuga & Rowe, 2005). As in MOPSO the repository size parameter is set equal to swarm size. Another parameter that has to be set in MOPSO-fs is the sigma share value. This is set empirically after several trials to 2.0.

It must be pointed out that several modifications were made to MOPSO and MOPSO-fs algorithms. Constraint handling was added to both algorithms. Furthermore in case of discrete valued variables like the material number the velocity update rules given by the Binary PSO were used (Kennedy & Eberhart, 1997). More details about these modifications can be found in Goudos and Sahalos (2006) and in Goudos et al. (2009).

Finally, another issue for all the above algorithms is the definition of the stopping criterion. Usually, this is the iteration number or the number of objective-function evaluations. In all the design cases presented here the algorithms stop when the maximum iteration number is reached.

APPLICATIONS, RESULTS, AND DISCUSSION

In this Section, we provide representative microwave and antenna design problems that are solved with the aforementioned optimizers. First, we present an example case of a multilayer planar microwave absorber. The design method is a hybrid of the IWPSO algorithm that uses both binary and real variables. In the second example, we address the common problem of linear array synthesis. In this case, we apply the Comprehensive Learning PSO variant, the classical IWPSO and CFPSO algorithms and real-coded GA. The design of a thinned planar microstrip array under constraints on the radiation pattern and the impedance matching between the array elements is the subject of the third example. We show that the BPSO-vm variant is the most suitable optimizer for the problem. The obtained solutions from other discrete PSO variants are also presented. In the last example, we apply the MOPSO-fs variant in the design of a dual band base station antenna array for mobile communications under constraints on return loss, sidelobe level and half-power beamwidth.

Multilayer Planar Microwave Absorber Design

The microwave absorbers are used in several applications, which range from electromagnetic interference mitigation to anechoic chambers. Usually, microwave absorbers are constructed by tiling material in a periodic way to simplify construction. Multilayer microwave absorbers are frequently combined with other devices to reduce the radar cross section of a wide range of objects. Such absorbers need not only to suppress reflection over a wide frequency band but also to be thin, practical, and economical.

The problem of a planar microwave absorber design lies in the minimization of the reflection coefficient of an incident plane wave in a multilayer structure for a desired range of frequencies and angles of incidence. The reflection coefficient depends on the thickness and the electric and magnetic properties of each layer. Several studies, which address this problem, exist in the literature. For example in (Cui & Weile, 2005) a parallel PSO algorithm is applied.

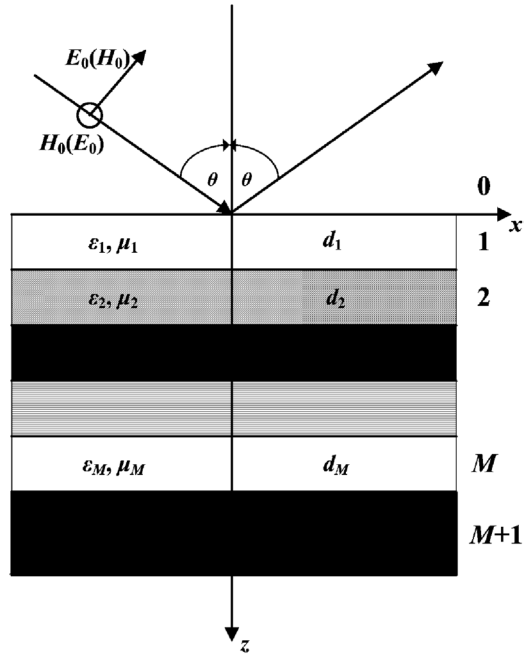
The multilayer planar microwave absorber, see Figure 1, comprises M layers of different materials backed by a perfect electric conductor (PEC) such as the ground plane (this layer's number is M+1). In this structure, the unknown parameters are the number of layers and their physical characteristics. The last are the thickness d_i and the frequency dependent permittivity and permeability of each layer given by

$$\varepsilon_i(f) = \varepsilon_0 \left(\varepsilon_i'(f) - j\varepsilon_i''(f)\right) \tag{14}$$

$$\mu_i(f) = \mu_0 \left(\mu_i'(f) - j\mu_i''(f)\right) \tag{15}$$

where ε_0 and μ_0 are the free space permittivity and permeability and $i = 1,\dots,M$. All layers are assumed infinite. Obviously, the total thickness of the absorber is

Figure 1. Multilayer planar absorber



$$d_{tot} = \sum_{i=1}^{M} d_i \tag{16}$$

The incident plane wave is either TE or TM and it is normal to the absorber. The incident media is the free space. In this multilayer structure, the general expression of the reflection coefficient at the interface between layers i and $i+1$ for an incident plane wave, $R_{i,i+1}{}^{TE}$ and $R_{i,i+1}{}^{TM}$, can be calculated (Chew, 1995) from the recursive formula

$$R_{i,i+1}{}^{TE/TM} = \frac{r_{i,i+1}^{TE/TM} + R_{i+1,i+2}^{TE/TM} e^{-2jk_{i+1}d_{i+1}}}{1 + r_{i,i+1}^{TE/TM} R_{i+1,i+2}^{TE/TM} e^{-2jk_{i+1}d_{i+1}}} \tag{17}$$

with

$$r_{i,i+1}^{TE} = \begin{cases} \dfrac{\mu_{i+1}k_i - \mu_i k_{i+1}}{\mu_{i+1}k_i + \mu_i k_{i+1}}, & i \le M \\ -1, & i = M+1 \end{cases}$$

and

$$r_{i,i+1}^{TM} = \begin{cases} \dfrac{\varepsilon_{i+1}k_i - \varepsilon_i k_{i+1}}{\varepsilon_{i+1}k_i + \varepsilon_i k_{i+1}}, & i \leq M \\ 1, & i = M+1 \end{cases} \tag{18}$$

where k_i is the wavenumber of the ith layer, that is,

$$k_i = 2\pi f \sqrt{\varepsilon_i \mu_i} \tag{19}$$

The design of the absorber is defined as the minimization problem of $R_1^{TE/TM}$ (in dB) given below:

$$R_{0,1}^{TE/TM} = 20\log\left(\max\left|R(f,\theta)\right| \;\; f \in \mathrm{B},\;\; \theta \in \mathrm{A}\right) \tag{20}$$

where $\max\left|R(f,\theta)\right|$ is the maximum reflection coefficient of the first layer over the desired frequency and angle range for a given polarization and A, B are the desired sets of angles of incidence and frequencies, respectively. Free space is assumed to be layer 0.

Several design approaches exist for the above problem. An effective way to do so is to use an exact penalty method and to combine the two objective functions in a single one. The objective function can be expressed as

$$F(f,\theta) = R(f,\theta) + \Xi \cdot \max\left\{0, d_{tot} - d_{des}\right\} \tag{21}$$

where Ξ is a very large number and d_{des} is the desired maximum total thickness. Parameter Ξ is chosen large enough to ensure that solutions which do not fulfill the constraint result in large fitness values. By minimizing the above formula with a global optimizer a solution can be found. The same algorithm can run for different values of d_{des} and different designs for different total thicknesses can be found.

We solve the above problem using a slightly modified hybrid variant of IWPSO. This uses the IWPSO position and velocity update rules for the real variables and binPSO for the discrete variables. We select a swarm size of 100 particles and set the number of iterations up to 3000.

A five-layer broadband absorber in the frequency range from 200MHz to 2GHz for normal incidence and TE polarization is considered. The design parameters are the material number ID_i and the layer thickness d_i. Here, we use a materials database in the literature (Michielssen, Sajer, Ranjithan, & Mittra, 1993). This database comprises several types of materials such as lossless and lossy dielectrics, lossy magnetics and relaxation type magnetic ones. The maximum layer thickness is 2mm. The calculated results are compared against the solutions obtained from a binary coded GA (Michielssen et al., 1993). Two design examples are considered. Table 1 presents the optimal design parameters found by the hybrid PSO variant and those reported in the literature using the GA. The characteristics of the materials are presented in Table 2; for brevity, we give the characteristics of the finally selected ones only. Finally, Figure 2 illustrates the frequency response of the structures. We notice that the two algorithms obtain absorber structures with similar performance. However, the PSO gives slightly thinner geometries.

Linear Antenna Array Synthesis

Several applications of modern wireless communications systems require radiation characteristics such as higher directivity that cannot be achieved by using a single antenna. The use of antenna arrays is the obvious solution for such a case. Among others, antenna arrays are employed for several radar and wireless communications applications in space and on earth. Their advantages include the possibility of fast scanning and precise control of the radiation pattern.

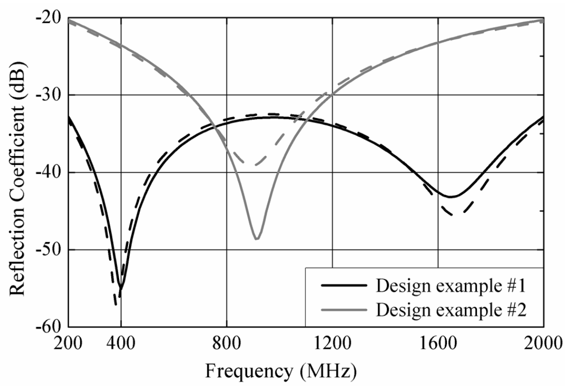
Table 1. Optimal design parameters (the distances are in mm)

Layer	Design example #1				Design example #2			
	GA		PSO		GA		PSO	
	ID_i	d_i	ID_i	d_i	ID_i	d_i	ID_i	d_i
1	*14*	0.966	*14*	0.96180	*16*	0.516	*16*	0.02335
2	*8*	1.002	*8*	0.94795	*4*	1.092	*16*	0.45153
3	*4*	1.182	*6*	0.04390	*4*	1.440	*4*	1.99725
4	*4*	0.984	*4*	1.69495	*4*	0.306	*4*	1.01406
5	*4*	1.380	*4*	1.85884	*4*	0.234	*4*	0.01202
6	Ground plane (PEC)							
d_{tot}		5.514		**5.494**		3.588		**3.498**

Table 2. Materials characteristics

Lossy Magnetic Materials ($\varepsilon' = 15,\ \varepsilon'' = 0$) $\mu = \mu' - j\mu''$ $\mu'(f) = \mu'(1\text{GHz})f^{-a}$ $\mu''(f) = \mu''(1\text{GHz})f^{-b}$				
#	$\mu'(1GHz)$	*a*	$\mu''(1GHz)$	*b*
4	3	1.000	15	0.957
Lossy Dielectric Materials ($\mu' = 1,\ \mu'' = 0$) $\varepsilon = \varepsilon' - j\varepsilon''$ $\mu'(f) = \mu'(1\text{GHz})f^{-a}$ $\varepsilon''(f) = \varepsilon''(1\text{GHz})f^{-b}$				
#	$\varepsilon'(1GHz)$	*a*	$\varepsilon''(1GHz)$	*b*
6	5	0.861	8	0.569
8	10	0.778	6	0.861
Relaxation-type Magnetic Materials ($\varepsilon' = 15,\ \varepsilon'' = 0$) $\mu = \mu' - j\mu''$ $\mu'(f) = \dfrac{\mu_m f_m^2}{f^2 + f_m^2}$ $\mu''(f) = \dfrac{\mu_m f_m f}{f^2 + f_m^2}$				
#	μ_m		f_m	
14	30		2.5	
16	25		3.5	
f is measured in GHz				

Figure 2. Plots of absorbers' frequency response; solid (dashed) lines refer to the PSO- (GA-) based solutions



Synthesis of linear antenna arrays has been extensively studied in the last decades using several analytical or stochastic methods (Bayraktar, Werner, & Werner, 2006; Benedetti, Azaro, Franceschini, & Massa, 2006; Boeringer & Werner, 2004; Ismail, & Hamici, 2010; Modiri & Kiasaleh, 2011b). The linear array design problem is multimodal and therefore requires the use of an optimization method that does not easily get trapped in local minima. Common optimization goals in array synthesis are the sidelobe level suppression (while preserving the main lobe gain) and the null control to reduce interference effects. For a uniformly excited linear array the above goals can be achieved by finding the optimum element positions.

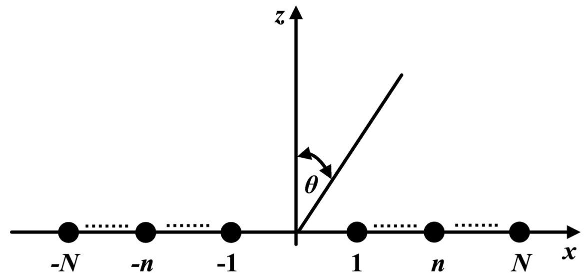
We assume a 2*N*-element linear array which is symmetrically placed along the *x*-axis, see Figure 3. The array elements are isotropic sources.

The antenna array factor in the *xz*-plane can be written as

$$AF(\theta) = 2\sum_{n=1}^{N} I_n \cos\left(\frac{2\pi}{\lambda} x_n \sin\theta + \varphi_n\right) \tag{22}$$

where I_n and φ_n are the excitation amplitude and phase, respectively, of the *n*th element, x_n is its position and λ is the wavelength. In a uniform excited array, it is $I_n = 1$ and $\varphi_n = 0$, $n = 1,\dots,N$. As a result, Equation 22 is simplified into

Figure 3. Geometry of the antenna array



$$AF(\theta) = 2\sum_{n=1}^{N} \cos\left(\frac{2\pi}{\lambda} x_n \sin\theta\right) \tag{23}$$

Our objective is to find the optimum element positions that minimize *SLL* while setting the mainlobe to a desired beamwidth BW_d within $\pm\Delta\theta$ deg. The use of an exact penalty method provides an effective way to combine the above objectives into a single cost function. In this case, the design problem is defined by the minimization of the objective function

$$\begin{aligned} F(\overline{x}) = {} & \max_{\theta\in S}\left\{AF_{dB}^{\,\overline{x}}(\theta)\right\} + \\ & \Xi \cdot \max\left\{0, \left|BW_c - BW_d\right| - \Delta\theta\right\} \end{aligned} \tag{24}$$

with $\overline{x}$ the vector of the element positions, S the space spanned by the angle θ excluding the main lobe and BW_c the calculated beamwidth.

The additional requirement of desired null level at specific directions modifies the objective function as:

$$F'(\overline{x}) = F(\overline{x}) + \Xi \cdot \left(\sum_{k=1}^{K} \max\left\{0, AF_{dB}^{\,\overline{x}}(\theta_k) - C_{dB}\right\}\right) \tag{25}$$

Figure 4. Convergence rate plot for the 28-element array case

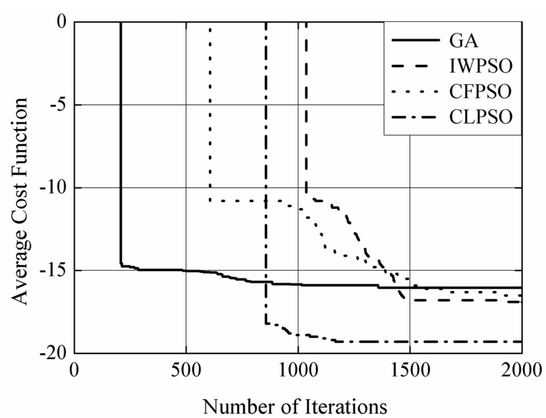
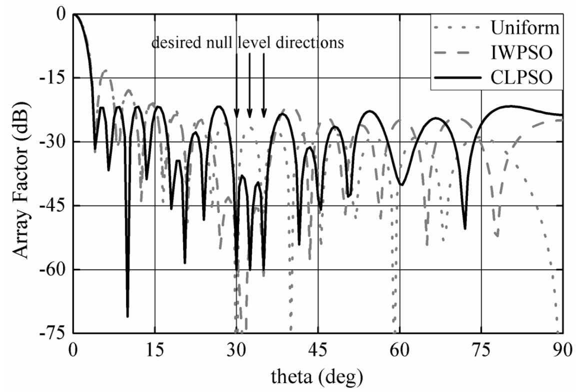


Figure 5. Optimized and uniform arrays patterns



where K denotes the number of the null directions, C_{dB} gives the desired null level (in dB) and θ_k is the direction of the kth null.

In a representative example taken from Khodier and Christodoulou (2005), we design a 28-element array with desired nulls at 30, 32.5, and 35 degrees. The desired null level and main-lobe beamwidth are –60dB and 8.35deg, respectively. The beamwidth tolerance is set to $\pm 12\%$. We apply three different PSO variants and a real-coded GA to the above-mentioned problem. The PSO algorithms are the IWPSO, the CFPSO and the CLPSO. In all the cases, the population size is 40 and the number of iterations is 2000. Each algorithm is initialized by using random values and it is executed twenty times. The best results are compared. Parameter Ξ is set to 10^6.

The convergence rate plot is shown in Figure 4. Clearly, CLPSO outperforms IWPSO and CFPSO methods in convergence speed and cost function value. The GA algorithm convergences faster than the PSO variants, but at the expense of a higher cost function value. Table 3 holds the optimal element positions in the array resulted from the CLPSO. The peak *SLL* is –21.63dB and the null levels are –60.04, –60.01 and –60dB at 30, 32.5 and 35deg, respectively. Khodier and Christodoulou (2005) solved the same problem with IWPSO. In that case, the optimized antenna's peak *SLL* was –13.27dB and the respective null levels were –52.74dB, –51.66dB and –61.46dB. The obtained radiation pattern is plotted in Figure 5. Notice that the array resulted from CLPSO gives a lower sidelobe level at least for the first two sidelobes closer to the main lobe. For comparison reasons, the radiation pattern of a uniform linear array with 28 elements is also drawn.

Table 3. Elements positions (normalized with respect to $\lambda/2$)

Element #	1	2	3	4	5	6	7
Position	0.470	1.322	2.263	3.178	4.142	5.369	6.212
Element #	8	9	10	11	12	13	14
Position	7.135	8.313	9.794	11.192	12.792	14.360	15.960

Thinned Planar Microstrip Array Design

Aperiodic antenna arrays have received great attention with the advances in both radio astronomy and radar techniques. Thinned arrays are a special type of aperiodic arrays in which a fraction of the elements in a uniformly spaced array are turned off to reduce the grating lobes resulted from the periodic grid. Thinned arrays are used in modern wireless communications ranging from cellular systems like UMTS to satellite communications.

In the third example, we focus on the constrained design of thinned planar microstrip arrays. Let us consider a $M \times M$ planar array with elements rectangular microstrip patches placed on a substrate of relative dielectric constant ε_r and thickness h. The array elements are uniformly excited with equal phase, simplifying the structure of the feeding network and providing a broadside radiation pattern.

In order to describe the array geometry, we consider a binary string of $D = M \times M$ bits in which "1" denotes the existence and "0" the absence of element. This string can be considered as the position of a particle in a binary PSO variant, a chromosome in a binary GA method, etc. The binary string that corresponds to the minimum cost function provides the optimum antenna array geometry. Next, we apply the method-of-moments (MoM) (Gibson, 2008) to calculate the far-field gain, $g(\theta, \varphi)$, and SLL of the array and the input impedance Z_{in} of each element. The last is used for the calculation of the return loss RL at the input of each element that is equal to

$$RL = 20 \log \left| \frac{Z_{in} - Z_0}{Z_{in} + Z_0} \right| \quad (\text{in dB}) \tag{26}$$

with Z_0 the characteristic impedance of the feeding network (a typical value is 50 Ohm). The RL calculation for all the elements allows the derivation of the average return loss among them, RL^{avg}. Low RL^{avg} values ensure that the impedance matching condition is well-satisfied for all the array elements.

Here, we desire a maximum far-field gain g_m at $\theta = 0^\circ$, a side lobe level threshold SLL_{des} and a maximum average return loss RL_{des}^{avg}. Therefore, the cost function may be formed as a weighted sum of three terms, i.e., it is

$$F = W_g\, g_m + W_s\, F_s + W_r\, F_r \tag{27}$$

with

$$F_s = R\left(SLL - SLL_{des}\right) \tag{28}$$

$$F_r = R\left(RL^{avg} - RL_{des}^{avg}\right) \tag{29}$$

where $R(\cdot)$ is the ramp function.

In case we require null control, i.e., a gain below a threshold value g_{des} at a specific direction (θ, φ), Equation 27 is modified into

$$F = W_g\, g_m + W_s\, F_s + W_r\, F_r + W_n\, F_n \tag{30}$$

where

$$F_n = R\left(g\left(\theta, \varphi\right) - g_{des}\right) \tag{31}$$

Here, we solve this design problem with the BPSO-vm variant. In order to validate our approach, comparisons with the BPSO, the binPSO, and a binGA variant (Dorica & Giannacopoulos, 2007a, 2007b) are also performed. In all cases, the population size is 30 and the maximum number of iterations in each run is 200. The algorithms are executed 20 times.

Two antenna design examples are studied. In both, it is $\varepsilon_r = 4$ and $h = 0.3$cm while the side length of each element is 3.4cm. Thus, the resonant

Table 4. Cost function characteristics (the best values are in bold)

Optimization Method	Minimum Value		Maximum Value		Mean Value		Standard Deviation	
	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2	Case 1	Case 2
BPSO-vm	**234.1**	**509.9**	**365.4**	**721.7**	**304.9**	**619.0**	**17.3**	**35.8**
binPSO	526.7	625.1	911.9	1017.5	728.2	820.8	53.9	66.4
BPSO	458.5	622.7	731.6	988.4	609.3	805.9	36.4	60.1
binGA	290.2	548.2	468.7	841.4	385.1	695.5	27.1	47.7

frequency of each single array element placed on the substrate is 2.14GHz (UMTS downlink). In the first example, we consider a 12×12 thinned planar array composed of elements that are symmetrical arranged with respect to the *x*- and *y*-axis. This symmetry simplifies the problem because the optimization is applied on a quarter of the antenna surface only. Thus, the array geometry is described by a 36-bit string. The distance between the centers of the elements along the *x*- and *y*-axis is 6.8cm. The cost function is described from Equation 27, where $W_g = -1$ and $W_s = W_r = 100$. The side lobe level and impedance matching requirements are $SLL_{des} = -20\text{dB}$ and $RL_{des}^{avg} = -20\text{dB}$, respectively. In the second example, we consider an 8×8 thinned planar microstrip array. Now, the array geometry is described from a 64-bit string. The distance between the centers of the elements along the *x*- and *y*-axis is 5.4cm. In this design case, we set (again) $SLL_{des} = RL_{des}^{avg} = -20\text{dB}$ but we further desire a gain threshold equal to –25dBi at $\theta = 45\,\text{deg}$ and $\varphi = 50\,\text{deg}$. Therefore, the cost function is the (30) in which we set $W_n = 100$ (the rest of the parameters are kept the same).

In Table 4, the minimum, maximum, mean and standard deviation of the cost function are given for all the optimization algorithms and both design examples. The convergence rate of the algorithms is shown in Figures 6 and 7. It is obvious that the BPSO-vm outperforms the other methods in terms of cost values and standard deviation but has a slightly slower convergence rate. A comparison

Figure 6. Average cost function versus number of iterations; design example 1

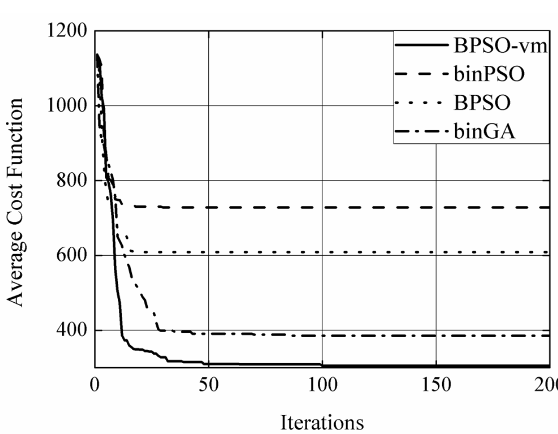


Figure 7. Average cost function versus number of iterations; design example 2

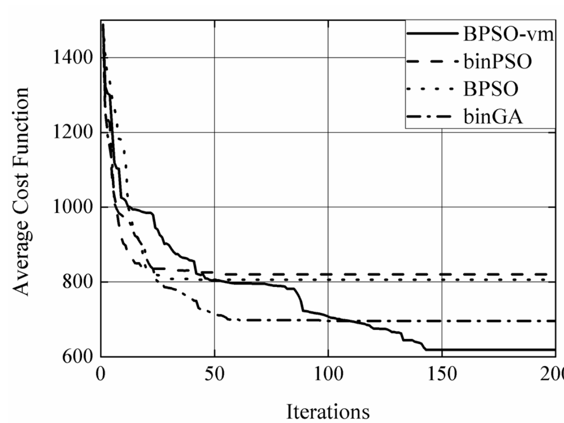


Figure 8. Optimized array geometry: Design example 1 (a) and 2 (b)

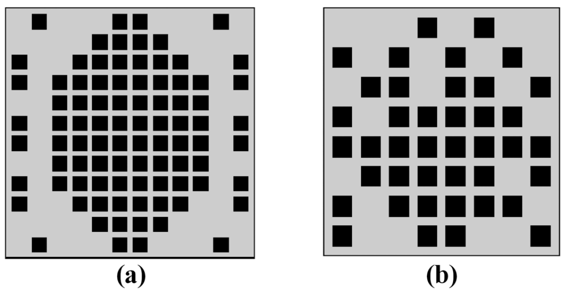
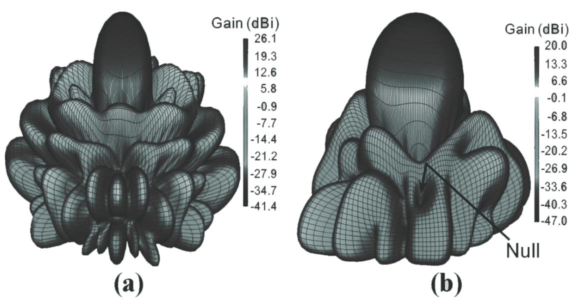


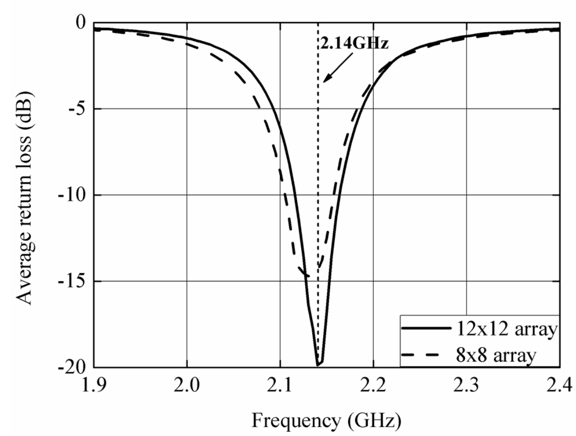
Figure 9. Radiation patterns of the optimized arrays: Design example 1 (a) and 2 (b)



between binGA and the other two PSO variants shows the improved performance of the first.

The optimized geometries for the two design examples resulted from the BPSO-vm are shown in Figure 8 while the three-dimensional radiation patterns of these arrays are presented in Figure 9. The maximum far-field gain and sidelobe level values of the first optimized array (design example 1) are $g_m = 26.1\text{dBi}$ and $SLL = -20\text{dB}$, respectively; the return loss of its elements at 2.14GHz varies from –11.3dB to –37dB with $RL_{avg} = -19.7$ dB. The second optimized array (design example 2) has $g_m = 20\text{dBi}$ and $SLL = -20\text{dB}$ while the return loss of its elements at 2.14MHz varies from –10.1dB to –27.3dB with $RL_{avg} = -14.7\text{dB}$. Its gain at $\theta = 45\,\text{deg}$ and $\varphi = 50\,\text{deg}$ is –25dBi, i.e., its radiation level is 45dB below the maximum. It obviously

Figure 10. Frequency response of the thinned planar microstrip arrays resulted from the BPSO-vm



comes that both arrays show an excellent frequency response. Finally, Figure 10 illustrates the frequency response of the two arrays.

Optimal Design of a Dual-Band Base Station Antenna

The simultaneous transmission of radio signals in different frequency bands is a common solution in order to provide a wide variety of services through a wireless network. Among different techniques, dual-band antennas are commonly used. Base stations play a crucial role in any cellular network. The wide spread of 3G networks and wireless LANs imposes new requirements for base station antennas.In the last example of this chapter, we study the design of such a structure. In particular, our objective is the optimal design of a dual-band base station K-element array that operates in the UMTS/WLAN frequency bands. Each array element comprises an active dipole covered by a cylindrical dielectric coating. Adjacent to the coating surface a second (parasitic) dipole is placed, see Figure 11. The system also includes a corner reflector. The array excitation is uniform with equal phases. The geometry parameters to be optimized are the active and the parasitic dipole lengths L_a and L_p, respectively,

Figure 11. Array element geometry

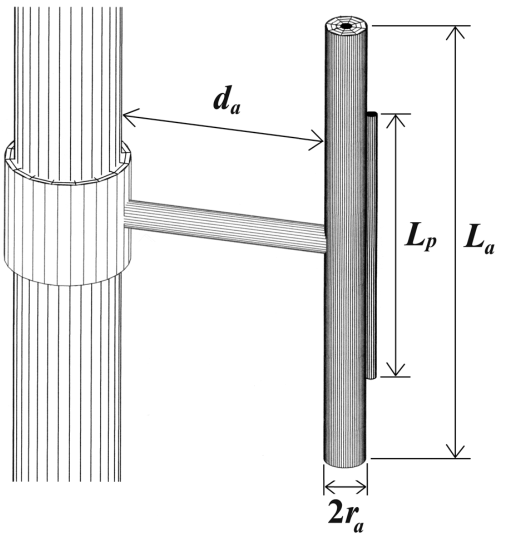
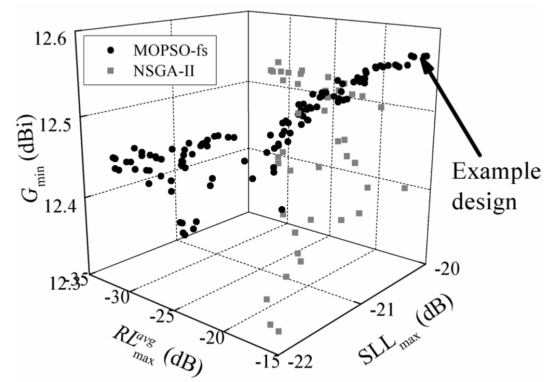


Figure 12. Pareto fronts found by the MOPSO-fs and NSGA-II



the reflector length L_r, the radius of the dielectric coating r_a, the vertical distance between the closest end points of any two adjacent active dipoles z_a and the distance d_a between an active dipole and the surface of the mast.

In general, this problem requires the optimization of $N = 4K + 1$ design parameters. However, the problem complexity reduces significantly in symmetric structures. For example, if the axis of symmetry is in the middle element normal to the element axis then $N = 2K + 2$ if K is even and $N = 2K + 3$ if it is odd. The problem is defined by three objectives which are subject to six constraints. The objectives are the minimization of the average return loss between all elements RL_f^{avg}, the minimization of the side lobe levels SLL_f, and the third is the gain, G_f, maximization, where the subscript f denotes the operating frequency ($f = 1,2$).

In order to solve this optimization problem, we apply the MOPSO-fs and the NSGA-II (Deb, Pratap, Agarwal, & Meyarivan, 2002) algorithms. In both cases, the population size is 140 and the number of iterations in an execution run is 2000. Each algorithm is executed 10 times.

The array concerns a six-element symmetrical array design (14 design parameters). The operating frequencies are 2.14GHz and 2.442GHz for UMTS and WiFi, respectively, transmission. The dielectric cover has relative permittivity 2.5 which remains constant in the operating frequency range. The diameter of the mast d_m is 2.54 cm. The imposed constraints are $RL_f^{avg} \leq -15\text{dB}$ and $SLL_f \leq -20\text{dB}$. Also, the horizontal half-power beamwidth for both frequencies is limited to $120\,\text{deg}$ with tolerance set to one degree.

Each point of the Pareto front is a feasible design solution with coordinates:

$$RL_{\max}^{avg} = \max_{f=1,2}\{RL_f^{avg}\},$$

$$SLL_{\max} = \max_{f=1,2}\{SLL_f\} \text{ and}$$

$$G_{\min} = \min_{f=1,2}\{G_f\}.$$

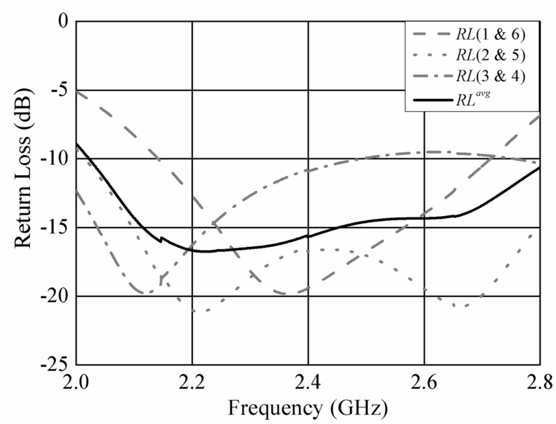
The Pareto fronts produced by the two algorithms are shown in Figure 12. In particular, MOPSO-fs has found 140 points of the Pareto front while NSGA-II has found only 88. It must be pointed out the MOPSO-fs required 1345 iterations to find 140 feasible solutions while NSGA-II required 1896 iterations to find 88 feasible solutions. The points on the Pareto front found by the

Table 5. Antenna geometric parameters (units in cm)

*i*th element	$L_a(i)$	$L_p(i)$	$d_a(i)$	$z_a(i, i+1)$
1	4.74	4.49	3.63	3.90
2	4.74	4.24	3.31	1.76
3	6.23	3.76	3.18	2.20
4	5.95	3.76	3.18	1.76
5	4.74	4.24	3.31	3.90
6	4.74	4.49	3.63	
$r_a = 0.4$, $L_r = 3.07$				

MOPSO-fs have larger dispersion. We select a point (shown with an arrow in Figure 12) of the Pareto front found by MOPSO-fs. This point represents a feasible antenna design with $RL_{\max}^{avg} = -15.1\text{dB}$, $SLL_{\max} = -20\text{dB}$ and $G_{\min} = 12.57\text{dBi}$. The geometric parameters of this antenna are reported in Table 5. We illustrate the return loss of each active dipole and the average return loss versus frequency in Figure 13. The antenna's horizontal and vertical radiation patterns are plotted in Figure 14. At this point, it should be mentioned that MOPSO-fs not only generates a larger number of feasible solutions compared

Figure 13. Return loss versus frequency

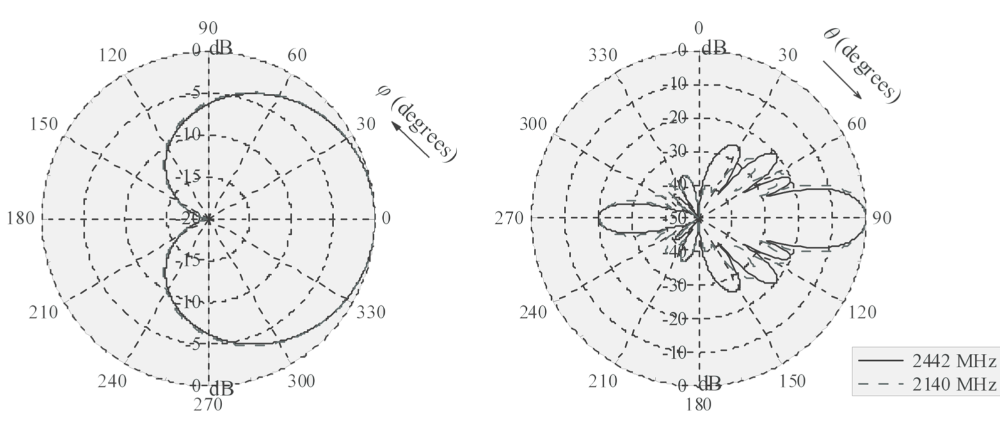


to the NSGA-II but, in average, provides a faster convergence too.

FUTURE RESEARCH DIRECTIONS

The research domain of evolutionary algorithms and especially swarm algorithms is growing rapidly. Specific PSO optimizers that work well for a given problem are introduced. A current and growing research trend in evolutionary algorithms is their hybridization with local optimizers. These algorithms are called Memetic Algorithms (MAs). MAs are inspired by Dawkins' notion of meme. The advantage of such an approach is that the use

Figure 14. Horizontal and vertical antenna radiation patterns



of a local search optimizer ensures that specific regions of the search space can be explored using fewer evaluations and good quality solutions can be generated early during the search. Furthermore, the global search algorithm generates good initial solutions for the local search. MAs can be highly efficient due to this combination of global exploration and local exploitation. So far several PSO-based MAs have been proposed in order to overcome some of the drawbacks and to improve the performance of common PSO optimizers. Recent developments include applications to real world problems like DNA sequence compression (Zhu, Zhou, Ji, & Shi, 2011) and face recognition algorithms (Zhou, Ji, Shen, Zhu, & Chen, 2011). Among others recent Memetic PSO algorithms include multimodal (Wang, Wang, & Wang, 2011) and multi-objective (Wei & Shang, 2011) optimization problems.

Parallelization techniques that use several CPUs or multicore programming can also be used in order to reduce computational cost. The Graphics Processing Unit (GPU) has also been used in electromagnetics combined with a numerical method in order to speed up calculation time (Gao, 2012; Junkin, 2011; Lee, Ahmed, Goh, Khoo, Li, & Hung, 2011; Peng & Nie, 2008; Tao, Lin, & Bao, 2010). PSO-based optimizers can be easily combined with such techniques to reduce computation time.

The application of multi-objective PSO algorithms to antenna and microwave design problems introduces new challenges regarding performance and computational cost. Recently a novel two local bests (lbest) based MOPSO (2LB-MOPSO) version is proposed (Zhao & Suganthan, 2010). This new MOPSO version enables the search around small regions in the parameter space in the vicinity of the best existing fronts. Therefore, new research directions have to be explored.

CONCLUSION

In this chapter, we discussed and evaluated the application of particle swarm optimizers to antenna and microwave design problems. We have presented common and state-of-the-art PSO methods found in the literature. These PSO methods were compared with other evolutionary algorithms on common design problems in electromagnetics. The obtained results exhibit the applicability and efficiency of particle swarm optimizers. The emerging trends and future research directions complete our study.

Within this context, we presented the classical IWPSO and CFPSO algorithms. We discussed about the CLPSO variant, which is a powerful optimizer for solving multimodal optimization problems. Two PSO versions for discrete-valued problems, the binPSO and the BPSO-vm were presented. A brief description of MOPSO and MOPSO-fs algorithms, which are suitable for solving multi-objective optimization problems subject to constraints, was given.

We applied the IWPSO algorithm for the design of a multilayer planar microwave absorber. The problem is inherently bi-objective and the two objective functions were combined into a single-objective one. The results reveal that this algorithm produces at a lower computational cost, slightly thinner geometries compared to GAs. The CLPSO, the common IWPSO and CFPSO algorithms and a real-coded GA were used for solving a constrained linear array synthesis problem. The derived results showed that CLPSO outperforms the two common PSO algorithms and the GA in terms of convergence rate and cost function values. Its major advantage is its updating strategy which results to a larger potential search space. Therefore, CLPSO is best suited for solving complex multimodal problems that are common in electromagnetics. We studied the problem of thinned planar microstrip array design and showed that the BPSO-vm outperforms other discrete PSO variants and a binary-coded GA in terms of best fitness value and standard deviation. We applied the MOPSO-fs to the design of a dual-band base station array. Our study showed that the MOPSO-fs algorithm performs better in finding the Pareto front than the popular NSGA-II algorithm.

Recent PSO algorithms and their application to electromagnetic design will be part of our future work. Furthermore, we plan to explore the applicability of other state-of-the-art algorithms to antenna and microwave design problems such as the recently proposed new MOPSO (Zhao & Suganthan, 2010).

PSO-based optimizers can be used in cases where the computational time plays an important role. Comparisons against popular evolutionary optimization methods in the literature showed that the PSO variants outperform or produce similar results in terms of solution accuracy and convergence speed. The PSO algorithms combined with a numerical method can be valuable tools for constrained single or multi-objective optimization problems in antennas and microwaves.

REFERENCES

Afshinmanesh, F., Marandi, A., & Rahimi-Kian, A. (2005, November 21-24). *A novel binary particle swarm optimization method using artificial immune system*. Paper presented at the EUROCON 2005 – The International Conference on “Computer as a Tool”, Belgrade, Serbia and Montenegro.

Baskar, S., Alphones, A., Suganthan, P. N., & Liang, J. J. (2005). Design of Yagi-Uda antennas using comprehensive learning particle swarm optimisation. *IEE Proceedings. Microwaves, Antennas and Propagation*, *152*(5), 340–346. doi:10.1049/ip-map:20045087

Bayraktar, Z., Werner, P. L., & Werner, D. H. (2006). The design of miniature three-element stochastic Yagi-Uda arrays using particle swarm optimization. *IEEE Antennas and Wireless Propagation Letters*, *5*(1), 22–26. doi:10.1109/LAWP.2005.863618

Benedetti, M., Azaro, R., Franceschini, D., & Massa, A. (2006). PSO-based real-time control of planar uniform circular arrays. *IEEE Antennas and Wireless Propagation Letters*, *5*(1), 545–548. doi:10.1109/LAWP.2006.887553

Boeringer, D. W., & Werner, D. H. (2004). Particle swarm optimization versus genetic algorithms for phased array synthesis. *IEEE Transactions on Antennas and Propagation*, *52*(3), 771–779. doi:10.1109/TAP.2004.825102

Chen, W.-N., Zhang, Z., Chung, H. S. H., Zhong, W.-L., Wu, W.-G., & Shi, Y.-H. (2010). A novel set-based particle swarm optimization method for discrete optimization problems. *IEEE Transactions on Evolutionary Computation*, *14*(2), 278–300. doi:10.1109/TEVC.2009.2030331

Chew, W. C. (1995). *Waves and fields in inhomogeneous media*. New York, NY: IEEE Press.

Clerc, M. (1999, July 6-9). *The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization*. Paper presented at the 1999 Congress on Evolutionary Computation, Washington, USA.

Coello Coello, C. A., Pulido, G. T., & Lechuga, M. S. (2004). Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, *8*(3), 256–279. doi:10.1109/TEVC.2004.826067

Cui, S., & Weile, D. S. (2005). Application of a parallel particle swarm optimization scheme to the design of electromagnetic absorbers. *IEEE Transactions on Antennas and Propagation*, *53*(11), 3616–3624. doi:10.1109/TAP.2005.858866

Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, *6*(2), 182–197. doi:10.1109/4235.996017

Deligkaris, K. V., Zaharis, Z. D., Kampitaki, D. G., Goudos, S. K., Rekanos, I. T., & Spasos, M. N. (2009). Thinned planar array design using Boolean PSO with velocity mutation. *IEEE Transactions on Magnetics*, *45*(3), 1490–1493. doi:10.1109/TMAG.2009.2012687

Dorica, M., & Giannacopoulos, D. D. (2007a). Evolution of wire antennas in three dimensions using a novel growth process. *IEEE Transactions on Magnetics*, *43*(4), 1581–1584. doi:10.1109/TMAG.2006.892105

Dorica, M., & Giannacopoulos, D. D. (2007b). Evolution of two-dimensional electromagnetic devices using a novel genome structure. *IEEE Transactions on Magnetics*, *43*(4), 1585–1588. doi:10.1109/TMAG.2006.892106

Eberhart, R. C., & Shi, Y. (2000, July 16-19). *Comparing inertia weights and constriction factors in particle swarm optimization*. Paper presented at the 2000 IEEE Conference on Evolutionary Computation, California, CA, USA.

Eberhart, R. C., & Shi, Y. (2001, May 27-30). *Particle swarm optimization: Developments, applications and resources*. Paper presented at the 2001 Congress on Evolutionary Computation, Seoul, Korea.

Gao, P. C., Tao, Y. B., Bai, Z. H., & Lin, H. (2012). Mapping the SBR and TS-ILDCs to heterogeneous CPU-GPU architecture for fast computation of electromagnetic scattering. *Progress in Electromagnetics Research*, *122*, 137–154. doi:10.2528/PIER11092303

Gibson, W. C. (2008). *The method of moments in electromagnetics*. Boca Raton, FL: CRC Press.

Goudos, S. K., Moysiadou, V., Samaras, T., Siakavara, K., & Sahalos, J. N. (2010). Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control. *IEEE Antennas and Wireless Propagation Letters*, *9*, 125–129. doi:10.1109/LAWP.2010.2044552

Goudos, S. K., Rekanos, I. T., & Sahalos, J. N. (2008). EMI reduction and ICs optimal arrangement inside high-speed networking equipment using particle swarm optimization. *IEEE Transactions on Electromagnetic Compatibility*, *50*(3), 586–596. doi:10.1109/TEMC.2008.924389

Goudos, S. K., & Sahalos, J. N. (2006). Microwave absorber optimal design using multi-objective particle swarm optimization. *Microwave and Optical Technology Letters*, *48*(8), 1553–1558. doi:10.1002/mop.21727

Goudos, S. K., Zaharis, Z. D., Kampitaki, D. G., Rekanos, I. T., & Hilas, C. S. (2009). Pareto optimal design of dual-band base station antenna arrays using multi-objective particle swarm optimization with fitness sharing. *IEEE Transactions on Magnetics*, *45*(3), 1522–1525. doi:10.1109/TMAG.2009.2012695

Huang, T., & Mohan, A. S. (2007). A microparticle swarm optimizer for the reconstruction of microwave images. *IEEE Transactions on Antennas and Propagation*, *55*(3), 568–576. doi:10.1109/TAP.2007.891545

Ismail, T. H., & Hamici, Z. M. (2010). Array pattern synthesis using digital phase control by quantized particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, *58*(6), 2142–2145. doi:10.1109/TAP.2010.2046853

Jin, N., & Rahmat-Samii, Y. (2007). Advances in particle swarm optimization for antenna designs: Real number, binary, single-objective and multiobjective implementations. *IEEE Transactions on Antennas and Propagation*, *55*(3), 556–567. doi:10.1109/TAP.2007.891552

Junkin, G. (2011). Conformal FDTD modeling of imperfect conductors at millimeter wave bands. *IEEE Transactions on Antennas and Propagation*, *59*(1), 199–205. doi:10.1109/TAP.2010.2090490

Kennedy, J., & Eberhart, R. (1995, November 27 - December 1). *Particle swarm optimization*. Paper presented at the 1995 IEEE International Conference on Neural Networks, Perth, Australia.

Kennedy, J., & Eberhart, R. (1997, October 12-15). *A discrete binary version of the particle swarm algorithm*. Paper presented at the 1997 IEEE International Conference on Systems, Man and Cybernetics, Orlando, USA.

Khodier, M. M., & Christodoulou, C. G. (2005). Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, *53*(8), 2674–2679. doi:10.1109/TAP.2005.851762

Lee, K. H., Ahmed, I., Goh, R. S. M., Khoo, E. H., Li, E. P., & Hung, T. G. G. (2011). Implementation of the FDTD method based on Lorentz-Drude dispersive model on GPU for plasmonics applications. *Progress in Electromagnetics Research*, *116*, 441–456.

Liang, J. J., Qin, A. K., Suganthan, P. N., & Baskar, S. (2006). Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. *IEEE Transactions on Evolutionary Computation*, *10*(3), 281–295. doi:10.1109/TEVC.2005.857610

Marandi, A., Afshinmanesh, F., Shahabadi, M., & Bahrami, F. (2006, July 16-21). *Boolean particle swarm optimization and its application to the design of a dual-band dual-polarized planar antenna*. Paper presented at the 2006 IEEE Congress on Evolutionary Computation, Vancouver, Canada.

Michielssen, E., Sajer, J.-M., Ranjithan, S., & Mittra, R. (1993). Design of lightweight, broad-band microwave absorbers using genetic algorithms. *IEEE Transactions on Microwave Theory and Techniques*, *41*(6), 1024–1031. doi:10.1109/22.238519

Modiri, A., & Kiasaleh, K. (2011a, August 30 - September 3). *Permittivity estimation for breast cancer detection using particle swarm optimization algorithm*. Paper presented at the 2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Boston, USA.

Modiri, A., & Kiasaleh, K. (2011b). Modification of real-number and binary PSO algorithms for accelerated convergence. *IEEE Transactions on Antennas and Propagation*, *59*(1), 214–224. doi:10.1109/TAP.2010.2090460

Peng, S., & Nie, Z. (2008). Acceleration of the method of moments calculations by using graphics processing units. *IEEE Transactions on Antennas and Propagation*, *56*(7), 2130–2133. doi:10.1109/TAP.2008.924768

Robinson, J., & Rahmat-Samii, Y. (2004). Particle swarm optimization in electromagnetics. *IEEE Transactions on Antennas and Propagation*, *52*(2), 397–407. doi:10.1109/TAP.2004.823969

Salazar-Lechuga, M., & Rowe, J. E. (2005, September 2-5). *Particle swarm optimization and fitness sharing to solve multi-objective optimization problems*. Paper presented at the 2005 IEEE Congress on Evolutionary Computation, Edinburgh, UK.

Sharaf, A. M., & El-Gammal, A. A. A. (2009, November 15-17). *A discrete particle swarm optimization technique (DPSO) for power filter design*. Paper presented at the 2009 4th International Design and Test Workshop, Riyadh, Saudi Arabia.

Shi, Y., & Eberhart, R. C. (1998). Parameter selection in particle swarm optimization . In Porto, V. W., Saravanan, N., Waagen, D., & Eiben, A. E. (Eds.), *Evolutionary Programming VII* (*Vol. 1447*, pp. 591–600). Lecture Notes in Computer Science Berlin, Germany: Springer. doi:10.1007/BFb0040810

Shi, Y., & Eberhart, R. C. (1999, July 6-9). *Empirical study of particle swarm optimization*. Paper presented at the 1999 Congress on Evolutionary Computation, Washington DC, USA.

Tao, Q., Chang, H.-Y., Yi, Y., Gu, C.-Q., & Li, W.-J. (2010, July 9-11). *An analysis for particle trajectories of a discrete particle swarm optimization*. Paper presented at the 2010 3rd IEEE International Conference on Computer Science and Information Technology, Chengdu, China.

Tao, Y., Lin, H., & Bao, H. (2010). GPU-based shooting and bouncing ray method for fast RCS prediction. *IEEE Transactions on Antennas and Propagation*, *58*(2), 494–502. doi:10.1109/TAP.2009.2037694

Wang, H., Wang, N., & Wang, D. (2011, May 23-25). *A memetic particle swarm optimization algorithm for multimodal optimization problems*. Paper presented at the 2011 Chinese Control and Decision Conference, Mianyang, China.

Wei, J., & Zhang, M. (2011, June 5-8). *A memetic particle swarm optimization for constrained multi-objective optimization problems*. Paper presented at the 2011 IEEE Congress on Evolutionary Computation, New Orleans, USA.

Zaharis, Z. D. (2008). Radiation pattern shaping of a mobile base station antenna array using a particle swarm optimization based technique. *Electrical Engineering*, *90*(4), 301–311. doi:10.1007/s00202-007-0078-y

Zaharis, Z. D., Kampitaki, D. G., Lazaridis, P. I., Papastergiou, A. I., & Gallion, P. B. (2007). On the design of multifrequency dividers suitable for GSM/DCS/PCS/UMTS applications by using a particle swarm optimization-based technique. *Microwave and Optical Technology Letters*, *49*(9), 2138–2144. doi:10.1002/mop.22658

Zhao, S.-Z., & Suganthan, P. N. (2011). Two-*lbests* based multi-objective particle swarm optimizer. *Engineering Optimization*, *43*(1), 1–17. doi:10.1080/03052151003686716

Zhou, J., Ji, Z., Shen, L., Zhu, Z., & Chen, S. (2011, April 11-15). *PSO based memetic algorithm for face recognition Gabor filters selection*. Paper presented at the 2011 IEEE Workshop on Memetic Computing, Paris, France.

Zhu, Z., Zhou, J., Ji, Z., & Shi, Y.-S. (2011). DNA sequence compression using adaptive particle swarm optimization-based memetic algorithm. *IEEE Transactions on Evolutionary Computation*, *15*(5), 643–658. doi:10.1109/TEVC.2011.2160399

ADDITIONAL READING

Carro, P. L., De Mingo, J., & Ducar, P. G. (2010). Radiation pattern synthesis for maximum mean effective gain with spherical wave expansions and particle swarm techniques. *Progress in Electromagnetics Research*, *103*, 355–370. doi:10.2528/PIER10031808

Chamaani, S., Abrishamian, M. S., & Mirtaheri, S. A. (2010). Time-domain design of UWB Vivaldi antenna array using multiobjective particle swarm optimization. *IEEE Antennas and Wireless Propagation Letters*, *9*, 666–669. doi:10.1109/LAWP.2010.2053691

Chamaani, S., & Mirtaheri, S. A. (2010). Planar UWB monopole antenna optimization to enhance time-domain characteristics using PSO. *AEÜ. International Journal of Electronics and Communications*, *64*(4), 351–359. doi:10.1016/j.aeue.2008.11.017

Chamaani, S., Mirtaheri, S. A., & Abrishamian, M. S. (2011). Improvement of time and frequency domain performance of antipodal Vivaldi antenna using multi-objective particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, *59*(5), 1738–1742. doi:10.1109/TAP.2011.2122290

Das, S., Abraham, A., & Konar, A. (2008). Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. In Y. Liu, A. Sun, E. H. T. Loh, W. F. Lu, & E.-P. Lim (Eds.), *Studies in Computational Intelligence: Vol. 116 - Advances of Computational Intelligence in Industrial Systems* (pp. 1-38). Berlin, Germany: Springer.

Demarcke, P., Rogier, H., Goossens, R., & De Jaeger, P. (2009). Beamforming in the presence of mutual coupling based on constrained particle swarm optimization. *IEEE Transactions on Antennas and Propagation*, *57*(6), 1655–1666. doi:10.1109/TAP.2009.2019923

Donelli, M., Martini, A., & Massa, A. (2009). A hybrid approach based on PSO and Hadamard difference sets for the synthesis of square thinned arrays. *IEEE Transactions on Antennas and Propagation*, *57*(8), 2491–2495. doi:10.1109/TAP.2009.2024570

Goudos, S. K., Baltzis, K. B., Bachtsevanidis, C., & Sahalos, J. N. (2010). Cell-to-switch assignment in cellular networks using barebones particle swarm optimization. *IEICE Electronics Express*, *7*(4), 254–260. doi:10.1587/elex.7.254

Goudos, S. K., Zaharis, Z., Baltzis, K. B., Hilas, C., & Sahalos, J. N. (2009, June 11-12). *A comparative study of particle swarm optimization and differential evolution on radar absorbing materials for EMC applications*. Paper presented at the EMC Europe Workshop 2009, Athens, Greece.

Khodier, M., & Al-Aqeel, M. (2009). Linear and circular array optimization: A study using particle swarm intelligence. *Progress in Electromagnetics Research B*, *15*, 347–373. doi:10.2528/PIERB09033101

Lanza, M., Pérez, J. R., & Basterrechea, J. (2009). Synthesis of planar arrays using a modified particle swarm optimization algorithm by introducing a selection operator and elitism. *Progress in Electromagnetics Research*, *93*, 145–160. doi:10.2528/PIER09041303

Li, W., Hei, Y., Shi, X., Liu, S., & Lv, Z. (2010). An extended particle swarm optimization algorithm for pattern synthesis of conformal phased arrays. *International Journal of RF and Microwave Computer-Aided Engineering*, *20*(2), 190–199.

Marler, R. T., & Arora, J. S. (2004). Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, *26*(6), 369–395. doi:10.1007/s00158-003-0368-6

Pan, F., Hu, X., Eberhart, R., & Chen, Y. (2008, September 21-23). *An analysis of bare bones particle swarm*. Paper presented at the 2008 IEEE Swarm Intelligence Symposium, Saint Louis, USA.

Panduro, M. A., Brizuela, C. A., Balderas, L. I., & Acosta, D. A. (2009). A comparison of genetic algorithms, particle swarm optimization and the differential evolution method for the design of scannable circular antenna arrays. *Progress in Electromagnetics Research B*, *13*, 171–186. doi:10.2528/PIERB09011308

Pérez, J. R., & Basterrechea, J. (2009). Hybrid particle swarm-based algorithms and their application to linear array synthesis. *Progress in Electromagnetics Research*, *90*, 63–74. doi:10.2528/PIER08122212

Shihab, M., Najjar, Y., Dib, N., & Khodier, M. (2008). Design of non-uniform circular antenna arrays using particle swarm optimization. *Journal of Electrical Engineering*, *59*(4), 216–220.

Wu, H., Geng, J., Jin, R., Qiu, J., Liu, W., Chen, J., & Liu, S. (2009). An improved comprehensive learning particle swarm optimization and its application to the semiautomatic design of antennas. *IEEE Transactions on Antennas and Propagation*, *57*(10), 3018–3028. doi:10.1109/TAP.2009.2028608

Yisu, J., Knowles, J., Hongmei, L., Yizeng, L., & Kell, D. B. (2008). The landscape adaptive particle swarm optimizer. *Applied Soft Computing*, *8*(1), 295–304. doi:10.1016/j.asoc.2007.01.009

Zhang, L., Yang, F., & Elsherbeni, A. Z. (2009). On the use of random variables in particle swarm optimizations: A comparative study of Gaussian and uniform distributions. *Journal of Electromagnetic Waves and Applications*, *23*(5), 711–721. doi:10.1163/156939309788019787

Zhang, S., Gong, S.-X., Guan, Y., Zhang, P.-F., & Gong, Q. (2009). A novel IGA-edsPSO hybrid algorithm for the synthesis of sparse arrays. *Progress in Electromagnetics Research*, *89*, 121–134. doi:10.2528/PIER08120806

Zhang, S., Gong, S.-X., & Zhang, P.-F. (2009). A modified PSO for low sidelobe concentric ring arrays synthesis with multiple constraints. *Journal of Electromagnetic Waves and Applications*, *23*(11-12), 1535–1544. doi:10.1163/156939309789476239

KEY TERMS AND DEFINITIONS

Genetic Algorithms: A stochastic population-based global optimization technique that mimics the process of natural evolution.

Method of Moments (MoM): A method for solving electromagnetic field problems using a full wave solution of Maxwell's integral equations in the frequency domain. The MoM is applicable to problems involving currents on metallic and dielectric structures and radiation in free space.

Non-Dominated Sorting Genetic Algorithm II (NSGA-II): A fast and elitist multi-objective evolutionary genetic algorithm. Its main features include a non-dominated sorting procedure and the implementation of elitism for multiobjective search, using an elitism-preserving approach.

Sidelobe Level (SLL): The ratio, usually expressed in decibels (dB), of the amplitude at the peak of the main lobe to the amplitude at the peak of a side lobe.

Universal Mobile Telecommunications System (UMTS): A third generation mobile cellular technology for networks developed by the 3GPP (3rd Generation Partnership Project).

Wireless Local Area Network (WLAN): A network in which a mobile user can connect to a local area network (LAN) through a wireless (radio) connection. Most modern WLANs are based on IEEE 802.11 standards, marketed under the Wi-Fi brand name.