

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

5,800

Open access books available

142,000

International authors and editors

180M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

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

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

Numbers displayed above are based on latest data collected.

For more information visit www.intechopen.com



Particle Swarm Optimization Algorithms with Applications to Wave Scattering Problems

Alkmini Michaloglou and Nikolaos L. Tsitsas

Abstract

Particle Swarm Optimization (PSO) algorithms are widely used in a plethora of optimization problems. In this chapter, we focus on applications of PSO algorithms to optimization problems arising in the theory of wave scattering by inhomogeneous media. More precisely, we consider scattering problems concerning the excitation of a layered spherical medium by an external dipole. The goal is to optimize the physical and geometrical parameters of the medium's internal composition for varying numbers of layers (spherical shells) so that the core of the medium is substantially cloaked. For the solution of the associated optimization problem, PSO algorithms have been specifically applied to effectively search for optimal solutions corresponding to realizable parameters values. We performed rounds of simulations for the the basic version of the original PSO algorithm, as well as a newer variant of the Accelerated PSO (known as "Chaos Enhanced APSO"/ "Chaotic APSO"). Feasible solutions were found leading to significantly reduced values of the employed objective function, which is the normalized total scattering cross section of the layered medium. Remarks regarding the differences and particularities among the different PSO algorithms as well as the fine-tuning of their parameters are also pointed out.

Keywords: Swarm Intelligence, optimization, particle swarm optimization (PSO), accelerated particle swarm optimization (APSO), chaos-enhanced APSO, chaotic APSO (CAPSO), wave scattering, cloaking

1. Introduction

Particle Swarm Optimization (PSO) is a population-based, stochastic optimization algorithm. It is modelled after the intelligent behavior patterns found in swarms of animals when they manage their biological needs. It was first introduced in 1995 [1], and since then many enhancements and new versions of the algorithm have appeared. The model originates from the behavior of flocks (swarms) of birds when in search of food sources. It was inspired by research carried out by Heppner and Grenander [2], in order to experiment on a "cornfield model". Exploiting these studies, Kennedy and Eberhart developed the PSO algorithm, in which the members of the swarm, called *particles* have some form of memory and common knowledge and are motivated by a common goal; in the mathematical framework this goal is the global optimum of the objective function of the optimization problem. The particles' positions represent the solutions, and depending on the method, they can

also have velocity or other characteristics, or even a societal structure. The swarm acts in alliance, aims to be effective, and there exists enough individuality to achieve diversity in possible solutions. By design, particle swarm optimization is inseparable from Swarm Intelligence. The swarm, as defined in literature, is designed to follow the basic principles of Swarm Intelligence, namely *proximity, quality, diverse response, stability* and *adaptability*.

In this chapter, two PSO algorithms are presented. First, the original PSO, which utilizes a global best position \mathbf{g}^* and an individual best position \mathbf{x}^* for the particles, which are described by both their position and velocity. This is considered to be the basic PSO algorithm, and the version chosen [3] also utilizes an inertia mechanism to describe the particles' movement. The second algorithm is an enhancement of the Accelerated Particle Swarm Optimization (APSO) algorithm, referred to as the Chaotic APSO (CAPSO) [4]. In this algorithm, the particles update their position in a single step and are only described by position, not velocity vectors. Additionally, they only use the global best position \mathbf{g}^* as an attraction to the optimum. Specified parameters get updated to fine tune the process, and precisely, the attraction parameter β updates through the use of chaotic maps.

Both aforementioned algorithms have been applied to wave scattering problems, and results of numerical implementations alongside with conclusions are provided. Precisely, we consider the cloaking problem concerning the excitation of a layered spherical medium with perfect electric conducting (PEC) core by an external dipole. The main purpose is to determine suitable parameters of the magneto-dielectric layers covering the PEC core so that the scattered far-field is significantly reduced for a wide range of observation angles. Obtained optimal designs demonstrating efficient cloaking performance are presented exhibiting reduced values of the bistatic scattering cross section for realizable coatings parameters. It is particularly stressed that the CAPSO determines optimal values of the scattering problem's variables, which yield highly-efficient cloaking designs by employing ordinary coatings materials.

PSO algorithms in computational methodologies and engineering applications involving electromagnetic waves were initially developed in [5, 6], where implementations in antenna design were also proposed. A quantum PSO algorithm, based on Quantum Mechanics rather than the Newtonian rules considered in the original versions of the algorithm, was developed in [7] and applied for finding a set of infinitesimal dipoles producing the same near and far fields of a circular dielectric resonator antenna. A molecular dynamics formulation of the PSO algorithm leading to a physical theory for the swarm environment was presented in [8] and applied to problems of synthesis of linear array antennas. Variants of PSO algorithms with relevant applications in electromagnetic design problems, like microwave absorbers and base station antenna optimization for mobile communications were analyzed in [9]. Specifically, concerning the cloaking behavior of layered media, related optimization problems were investigated in [10–16]. Optimization techniques for meta-devices design are overviewed in [17].

2. Particle Swarm Optimization (PSO)

In this section, the basic principles of Particle Swarm Optimization (PSO) are presented and an in depth description of the algorithms that have been developed and applied for the considered cloaking problems is given. After discussing the theoretical basis of the swarm optimization method and its ties to Swarm Intelligence, the PSO algorithm and the chaotic-enhanced version (CAPSO) of the accelerated particle swarm optimization (APSO) algorithm are described.

2.1 Introduction to PSO

PSO is a population-based stochastic optimization algorithm, modelled after the behavior of swarms of animals, like flocks of birds, swarms of various types of insects or ants or school of fish [18]. In literature, it is also categorized as a metaheuristic algorithm. Usually, the population is referred to as a swarm. These types of methods are also considered to be and referred to as behaviorally-inspired, opposed to evolutionary-based methods like genetic algorithms, although some parallels can be drawn between them, with regards to their inner workings. Another similar research field is artificial life. The term, as well as the algorithm, was originally proposed in 1995 [1] and although PSO's precursor was the study and simulation of animal behavior (even in the hopes of studying human social behavior), it grew into an optimizer, with a simple, yet well-defined description. By definition, PSO is indissolubly linked to Swarm Intelligence.

The appeal of swarm optimizers is due to numerous reasons. There exist many types of biological swarms, so one can safely assume that they constitute a promising pool of inspiration and resources to draw methods and conclusions from. The global adaptive behavior of the swarm, and its co-operational behavior and decision making, is practical but not strictly utilitarian, since a swarm behaves with fluid and elegant coordination. Additionally, the way a biological swarm acts can be clearly and directly perceived by humans. Thus, we have a better understanding of the animals' purpose, goals, communication and utility unlike other natural phenomena, which can be way more abstract, complicating the creation of a well-structured model or method.

Since the initial introduction of PSO, several variations of the method have been introduced. A plethora of algorithms have been and are still being designed with different parameters and applications in mind, in order to adjust to specific problems. These numerous variants are widely used and examined, and, thus, PSO has grown to be a very effective technique. In the following subsection, a more generic description of the swarm and its behavior is presented, while detailed descriptions of specific algorithms are given in the sequel.

2.1.1 The Particle Swarm

The term "particle" refers to the points in the n -dimensional space (where n is the number of variables of the objective function) which represent the biological entities of the swarm. Let us assume that the representative animal species is birds. The swarm consists of the entirety of the particles, making up the population. The particles have neither mass, nor volume and although they could be considered points in space, the term particle has been chosen as a good compromise, due to its more active usage in literature [1].

Each particle maintains information about two characteristics; its position \mathbf{x} and velocity \mathbf{u} . The position is strictly the most important characteristic, since it represents the solutions to the objective function of the optimization problem. The particles also have some common memory of useful information, since they share information regarding the best position the swarm has achieved (based on the objective function), referred to as the "global best" \mathbf{g}^* . In nature, this knowledge could refer to food, shelter or destination. Depending on the variant or type of PSO algorithm being used, they can also remember their individual best position \mathbf{x}^* , or a set of best positions if they follow a different type of structure, or even a best position that represents their social clan and/or leader.

According to [19], the biological swarm has three specific qualities. First, *cohesiveness*: the members are not unrelated to each other and all of them are part of the

same group, thus to an extent, they “stick together”. Second, there is *separation*, the members actively try to not collide with each other and move with some respect to the average distance between them. Last, there is *alignment*, the whole population actively tries to move towards the same direction as a group effort. In Biology, this is the source of food, while in Optimization it is the optimum of the problem. Of course, since particles are designed to be without mass and volume, separation is not a physical quality the swarm is forced to have. When converging to a solution, all particles end at or near to the specific position representing this solution. However, separation exists as a principle, since particle “collision” does not hinder their movement in any way, shape or form. Particles are separate entities to each other to a certain degree since they are created with their individual attributes (e.g. initial positions, individual best, clan leader and more, depending on the algorithm) and act accordingly, having a degree of autonomy, while searching in unison with respect to the swarm.

2.1.2 Basic Principles of Swarm Intelligence

In order to clearly establish the link between PSO and Swarm Intelligence, we present a comprehensible list of Swarm Intelligence principles, in reference to Millonas’ categorization [1, 18, 20]. Let us refer to a group of entities that collectively act and behave. This group has Swarm Intelligence if these principles are true.

1. **Proximity principle.** The members of the group should be able to handle and do elementary space and time computations. This means that the group can behaviorally respond to environmental stimuli and changes. Also, they should be able to do so in order to better conduct their main utilities and functions which are specific to this group. Such activities vary, depending on the group, for example a swarm of ants could have a main utility of food foraging.
2. **Quality principle.** The group should not only react to time and space stimuli, but also check for quality factors and parameters, e.g. safety.
3. **Principle of diverse response.** The group should not respond to its environment in an absolutely ordered manner. There should be safety locks, and insurance policies for it to survive in case of unpredicted changes and fluctuations in the environment. Resources should not all rely to a single point of focus. Therefore, the swarm must be prepared to act and respond with diverse and alternative solutions.
4. **Principle of stability.** The group as a whole, should not reform its behavior patterns into a completely alternate mode every time a change happens, since such an intense structural and behavioral change wastes too much energy, and might eliminate the possibility of reaching good results.
5. **Principle of adaptability.** However, the group should also be able to switch its behavioral mode, provided this change is a positive one and the group has ways of knowing so.

One can observe that stability and adaptability are principles that go hand-in-hand and the best strategy to approach, is to safely explore a viable middle ground. Some level of randomness or noise should exist in the group, to a degree that diverse response is allowed to happen. That is the reason why such parameters are usually very important to the algorithms and can dramatically change their results.

PSO dictates that the swarm acts in a way which is complicit with the aforementioned principles. In the original PSO publication, Kennedy and Eberhart do confirm that the PSO algorithm has been designed to function in this manner. Similar explanations and proofs were provided in literature [1, 18]. As it has been briefly mentioned, in PSO, particles maintain their position and velocity, and have the ability to react to environmental time and space stimuli in order to update them. They do so in time steps-iterations, thus following the *proximity principle*. The swarm reacts to the global best value \mathbf{g}^* alongside with other quality factors when doing said updates, so it enforces the *quality principle*. Said quality factors, do not prevent the *diverse response*, because the swarm avoids behaving in an excessively restricted manner. This is encouraged by diversity and noise existing within the swarm. Lastly, the swarm bases its behavioral change(s) on a well-defined criterion (which includes the global best position \mathbf{g}^*), thus providing *adaptability* without jeopardizing the swarm's *stability*. The mode of behavior changes when it is beneficial and cost-effective.

2.2 The PSO algorithm

In this section, we refer to the original PSO algorithm [1], alongside with the upgrade proposed in 1998 [3] which utilizes an inertia mechanism.

2.2.1 Description

The PSO algorithm follows all the principles and characteristics mentioned so far. By default, a maximization optimizer is considered due to the way the model works, but there exist methods to effectively utilize the algorithm in order to find minima as well.

The behavior of the flock was heavily inspired by and based upon Heppner's [2] simulation of a bird flock, referred to as a *cornfield model* or *cornfield vector*. Heppner wanted to simulate the way a flock of birds moves while searching for food (namely "cornfield" in the simulation). The birds' behavior in real life, hints to the existence of what we refer to as a common sense or knowledge, meaning that members of the flock have the ability to share knowledge originating from their peers without having experienced it themselves. This serves as both a cognitive function and a means of communication. Very often, we do witness this phenomenon; flocks of birds can discover a new bird feeder in their area in a matter of few hours, and an increasing number of them will systematically start visiting it. This behavior was modelled in the simulation, in which the birds were given two types of memory. For the flock's memory of food sources they were given what we previously referred to as the global best \mathbf{g}^* and for their individual memory, they kept information of the best position they have individually visited, their \mathbf{x}^* . There were also extra parameters to adjust how effectively each memory spot affects the birds' movement and behavior.

Kennedy and Eberhart [1] utilized Heppner's simulation model, and designed the PSO algorithm in order to use these advantageous observations. So, in the PSO algorithm, the model is as follows.

1. When particles locate a good solution to the optimization problem, this knowledge is transmitted to the whole swarm, meaning that the \mathbf{g}^* value is known to each member.
2. All particles do gravitate towards good solutions, but not in an absolute forced way, because,

3. all particles maintain their personal memory spot for their own value \mathbf{x}^* , thus preserving some ability for independent thinking.

The particles move with respect to Newton's laws of motion, while there exist parameters to insert some randomness. There exist also learning rates that the particles adhere to.

In 1998, Shi and Eberhart [3] proposed strategies on how to fine-tune the parameters of the original PSO algorithm. Particularly, they suggested the use of an inertia weight mechanism θ applied to the particles' movement because it was found in experimentation that the particle velocities built up too fast and the maximum of the objective function can be skipped. Usually, the inertia decreases in a linear manner while the iterations of the algorithm run, and it gets updated once per iteration i . For the inertia, the values $\theta_{\max} = 0.9$ and $\theta_{\min} = 0.4$ are commonly used [19].

$$\theta_i = \theta_{\max} - \frac{\theta_{\max} - \theta_{\min}}{i_{\max}} i \quad (1)$$

Therefore, the velocity and position updates are described, respectively, in the following formulae, with respect to iteration i :

$$\mathbf{u}_i = \theta_i \mathbf{u}_{i-1} + c_1 r_1 [\mathbf{x}^* - \mathbf{x}_{i-1}] + c_2 r_2 [\mathbf{g}^* - \mathbf{x}_{i-1}] \quad (2)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{u}_i, \quad (3)$$

where the parameters c_1 and c_2 are the cognitive (individual) and social (group) learning rates and are usually assumed to both be 2, so that the particle overflies the target approximately half of the time. It is interesting to note that if c_1 and c_2 are different to each other, then the particles will in time favor one type of best position (or *behavior*) over the other. In a way, this would conceptually translate to the particles choosing to be more selfish than social and vice versa. This could lead to less optimal solutions than the ones expected. The parameters r_1 and r_2 are uniformly distributed random numbers in the range from 0 to 1.

2.2.2 Algorithm

After describing the model of the algorithm, a concrete and defined algorithm can be presented for the computational implementation. The algorithm is depicted in pseudo code form in **Figure 1**.

Regarding the various parameters, we make the following remarks. Usually a size of 20 to 30 for N is assumed, but these numbers can vary depending on the optimization problem. The bigger the swarm, the more evaluations of the objective function f are made during each iteration, thus due to the computations, the algorithm becomes more time consuming. From a programmer's point of view, f does not necessarily need to be an input, however, it is depicted in this manner for reasons of clarity.

2.3 The CAPSO algorithm

As we have previously mentioned, in the original version of the PSO algorithm, both a global (\mathbf{g}^*) and an individual best (\mathbf{x}^*) are used, with the particles' position being greatly affected by them. The accelerated particle swarm optimization algorithm (APSO) however, introduced by Yang [21], follows a different approach. The

Algorithm 1: Particle Swarm Optimization (PSO)

Input: N , \mathbf{x}_l , \mathbf{x}_u , c_1 , c_2 , i_{\max} , f
Output: A swarm S of size N (N position vectors)
 Initialize S , randomly generate the position \mathbf{x} of each particle w.r.t. the bounds \mathbf{x}_l , \mathbf{x}_u of the objective function;
 Initialize all velocities \mathbf{u} to zero;
 Initialize best positions \mathbf{x}^* (and respective values) for individual particles and find \mathbf{g}^* ;
 Choose randomly two values in $[0, 1]$ for r_1 and r_2 ;
 Iteration $i = 0$;
 Initialize $\theta_{\min}, \theta_{\max}$;
while $i < i_{\max}$ **do**
 Calculate inertia: $\theta = \theta_{\max} - \frac{\theta_{\max} - \theta_{\min}}{i_{\max}} i$;
 For each particle in S , the values for iteration i are:
 1. Update velocity: $\mathbf{u}_i = \theta \mathbf{u}_{i-1} + c_1 r_1 [\mathbf{x}^* - \mathbf{x}_{i-1}] + c_2 r_2 [\mathbf{g}^* - \mathbf{x}(i-1)]$;
 2. Update position: $\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{u}_i$;
 3. Compute the value of the new position according to f ;
 4. Check/Update: $\mathbf{x}^*, \mathbf{g}^*$
 (Optional) Check for convergence;
 Upgrade iteration: $i = i + 1$;
end
return S ;

Figure 1.
 The PSO algorithm pseudo code.

chaos-enhanced particle swarm optimization, or chaotic APSO (CAPSO) is a variation of the APSO algorithm.

2.3.1 Accelerated Particle Swarm Optimization

It is noted that the individual best \mathbf{x}^* in PSO, acts as a creator of diversity in the swarm. That is not necessarily the only purpose of the individual best, but it is a very prominent one. Thus, this diversity could be recreated by utilizing randomness to bypass the use of the individual best. There exist some algorithms that belong in this more “simplistic” philosophy, and try to use only the most necessary parameters and formulae. The accelerated particle swarm optimization algorithm (APSO), follows this route. APSO has been applied in many optimization problems and is a solid method with good results. One can safely develop and use APSO, and similar methods or variants, while keeping in mind that PSO, or even more its standard versions, is still *in general* a better option if the optimization problem of interest is highly nonlinear and multimodal [21].

Ergo, the APSO algorithm only uses the global best \mathbf{g}^* to generate the velocity vector \mathbf{u} , resulting to using a simpler mathematical formula. For a specific particle, during the i – *th* iteration, the velocity is:

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \alpha(r - 1/2) + \beta(\mathbf{g}^* - \mathbf{x}_{i-1}) \quad (4)$$

where r is a random variable with values from 0 to 1, and the $1/2$ is used as a means of convenience. It is suggested [21], that a normal distribution αr_i is used, where r is drawn from $N(0,1)$. Thus, velocity and positions updates are given, respectively, by

$$\mathbf{u}_i = \mathbf{u}_{i-1} + \beta(\mathbf{g}^* - \mathbf{x}_{i-1}) + \alpha r_{i-1}, \quad (5)$$

$$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{u}_i \quad (6)$$

In [21], the following simplified formula is also suggested for the particle location update in a single step:

$$\mathbf{x}_i = (1 - \beta)\mathbf{x}_{i-1} + \beta\mathbf{g}^* + \alpha r_{i-1}, \quad (7)$$

hence there is no need of utilizing structs or vectors for the velocity, while separate initializations and updates are also avoided.

The typical parameter values for this accelerated PSO are $\alpha \in [0.1, 0.4]$ and $\beta \in [0.1, 0.7]$. More generally, we must keep in mind that these parameters should scale with respect to the scales of the problem variables. A further improvement to APSO [21] is to reduce the randomness as iterations proceed. This means that we can use a monotonically decreasing function specifically for the parameter α , e.g.

$$\alpha = \alpha_0 \gamma^t, \quad (0 < \gamma < 1) \quad (8)$$

or

$$\alpha = \alpha_0 e^{-\gamma t}. \quad (9)$$

Other non-increasing functions $\alpha(t)$ can be used like the example provided in code in [21].

2.3.2 Chaos-Enhanced APSO

Gandomi et al. proposed a variation of the APSO algorithm, the chaotic APSO (CAPSO) [4]. According to the study, the attraction parameter β in (Eq. (7)) is crucially important in determining the speed of the convergence and how the algorithm behaves, since this parameter characterizes the variations of the global best attraction. A well tuned β is of great importance. After parametric investigations, it is suggested that β should be in $[0.2, 0.7]$ for most problems solved by APSO. Additionally, it is noted that the parameter β has no practical reason of remaining a constant. On the contrary, a varying β can offer an advantage in terms of convergence speed and algorithm behavior.

The method suggested for tuning the parameter β is *chaotic maps*. In Mathematics, chaotic maps are evolutionary functions that exhibit some sort of chaotic behavior [22]. Chaotic maps often occur in the study of dynamical systems. Also, they are used to generate fractals. They can change in time in a continuous or discrete manner, but usually chaotic maps are discrete ones. Therefore, they take the form of iterated functions. Chaotic maps are normalized, their variations are always between $[0, 1]$, so they can safely be used for tuning the parameter β .

In the original proposal of CAPSO [4], many chaotic maps were tested in terms of convergence and effectiveness. The results were listed in detail, and it was noted that the *Sinusoidal* map was the best performing one, and the *Singer* map was the

second best. Consequently, the Sinusoidal map is the best choice for applications. It was noted that chaotic maps with a unimode centered around their middle tend to produce better results, and Sinusoidal and Singer maps fall into this category. They are as follows:

Sinusoidal Map:

$$x_{k+1} = ax_k^2 \sin(\pi x_k) \quad (10)$$

As an alternative, the following simplified form has also been suggested and applied [4, 23]:

$$x_{k+1} = \sin(\pi x_k) \quad (11)$$

Singer Map:

$$x_{k+1} = \mu(7.86x_k - 23.31x_k^2 + 28.75x_k^3 - 13.302875x_k^4), \quad (12)$$

where $\mu \in [0.9, 1.08]$.

2.3.3 The CAPSO Algorithm

Having described the basis of the APSO algorithm, as well as the improvements added from chaotic maps, the CAPSO algorithm is now presented in pseudo code form in **Figure 2**.

The following information is provided for the various parameters. Usually a size of 40 for N is considered sufficient, but these numbers can vary depending on the optimization problem. The parameter α gets updated through a chosen $\alpha(t)$ (which

Algorithm 2: Chaotic APSO (CAPSO)

Input: l, u, it_{max}, n

Output: A swarm S of n particles and their final positions

Initialize S of size n with random positions w.r.t the upper and lower bounds u and l ;

Initialize α and β parameters;

Initialize iteration, $it = 1$;

Set convergence as *false*;

while $it \leq it_{max}$ AND convergence == *false* **do**

 Find global best g^* for iteration it ;

 Update α ;

 Update β through a chaotic map;

for Each particle in S **do**

 | Update locations using (Eq. (7));

end

 Update the iteration counter, $it = it + 1$;

 Update convergence;

end

Figure 2.
 The CAPSO algorithm pseudo code.

is a monotonically decreasing function or a non-increasing function in general). For α , the initial value depends on the scale of the problem variables and on $\alpha(t)$. One can apply the values proposed for APSO, or alternatively $\alpha = 10$ can be chosen for an initial value as a starting point. Testing with different initial values is encouraged. The parameter β is updated through a chaotic map, preferably the Sinusoidal map. In the original paper [4], the maximum iteration number is suggested to be 250. One must keep in mind that depending on the problem, these values might have to be re-evaluated and re-adjusted.

2.4 Development suggestions

Many suggestions can be made regarding the robustness of algorithms, as well as the speed, effectiveness and organization of the code. All these highly depend on the programming language, development technique, programmer expertise, computational load of the optimization problem and numerous more parameters. When developing these algorithms, we must take into consideration all of the above, and more, since applications can greatly diversify from one another.

Below, two suggestions are made regarding the PSO and APSO/CAPSO algorithms, which, when applied, improved the testing process on a complicated wave scattering optimization problem detailed below. However, they are not heavily dependent on the nature of said optimization problem, and they could be proven to be helpful regardless.

1. Application of constraints/bounds. A method that reassures that the variables remain in their allowed bounds is vital. This is very common in optimization. If a variable crosses a bound, the lower or higher permitted value can be enforced, with respect to which bound was crossed. This reassures that the swarm will not go out of bounds if it gets driven to do so by a nearby invalid optimum. Additionally, it ensures that the final output of the algorithm is a valid and applicable one, even if it is not the best optimum. For complex optimization problems, constraint/bound checking can be complicated, if for example the variables have to follow specific rules, or have specific characteristics in relation to each other. We can see this technique being applied in APSO's code [21].

2. Convergence checking. By default, in most PSO related algorithms, it is implied that the algorithm stops when it reaches a pre-defined maximum of iterations. However, many times, the swarm can find a solution faster than that. Thus, if there is a convergence criterion (representing the degree in which the population agrees on a solution), it can be applied as an end condition for the algorithm. For example, a very common convergence criterion is standard deviation.

3. Particle swarm optimization in wave scattering problems

In this section, PSO optimizations to representative applications of wave scattering theory are presented. Precisely, we investigate the electromagnetic cloaking of spherically layered media excited by an external source. The optimizations concern the determinations of the physical (material) and geometrical characteristics of the layered medium so that the scattered far field generated by the layered medium is significantly reduced.

The scattering geometry is depicted in **Figure 3**. It consists of a layered spherical medium V with external radius a_1 . The interior of V is divided by $P - 1$ concentric spherical interfaces $r = a_p$ ($p = 2, \dots, P$) into $P - 1$ homogeneous magneto-dielectric layers V_p ($p = 1, \dots, P - 1$), consisting of materials with real relative dielectric permittivities ε_p and magnetic permeabilities μ_p , and surrounding a perfect electric conducting (PEC) core (layer V_P). The exterior V_0 of V has permittivity ε_0 , permeability μ_0 , and wavenumber k_0 . Medium V is excited by an external magnetic dipole, with position vector \mathbf{r}_0 on the z -axis and dipole moment along the direction $\hat{\mathbf{y}}$.

The exact solution of the considered scattering problem was determined in [24–26] by means of a combined Sommerfeld and T-matrix methodology in conjunction with suitable eigenfunctions expansions. Specifically, the electric fields in each spherical shell are decomposed into primary and secondary components, which are then expressed as series of the spherical vector wave functions. The unknown coefficients in the expansions of the secondary fields are determined analytically by imposing the transmission boundary conditions on the interfaces of the spherical shells and applying a T-matrix method. It is emphasized that the exact solution of the scattering problem (here this is obtained in the form of a Mie series) is crucial for the fast and efficient implementation of the PSO algorithm in the present setting.

By applying the above-described methodology, we obtain the following expression of the total scattering cross section

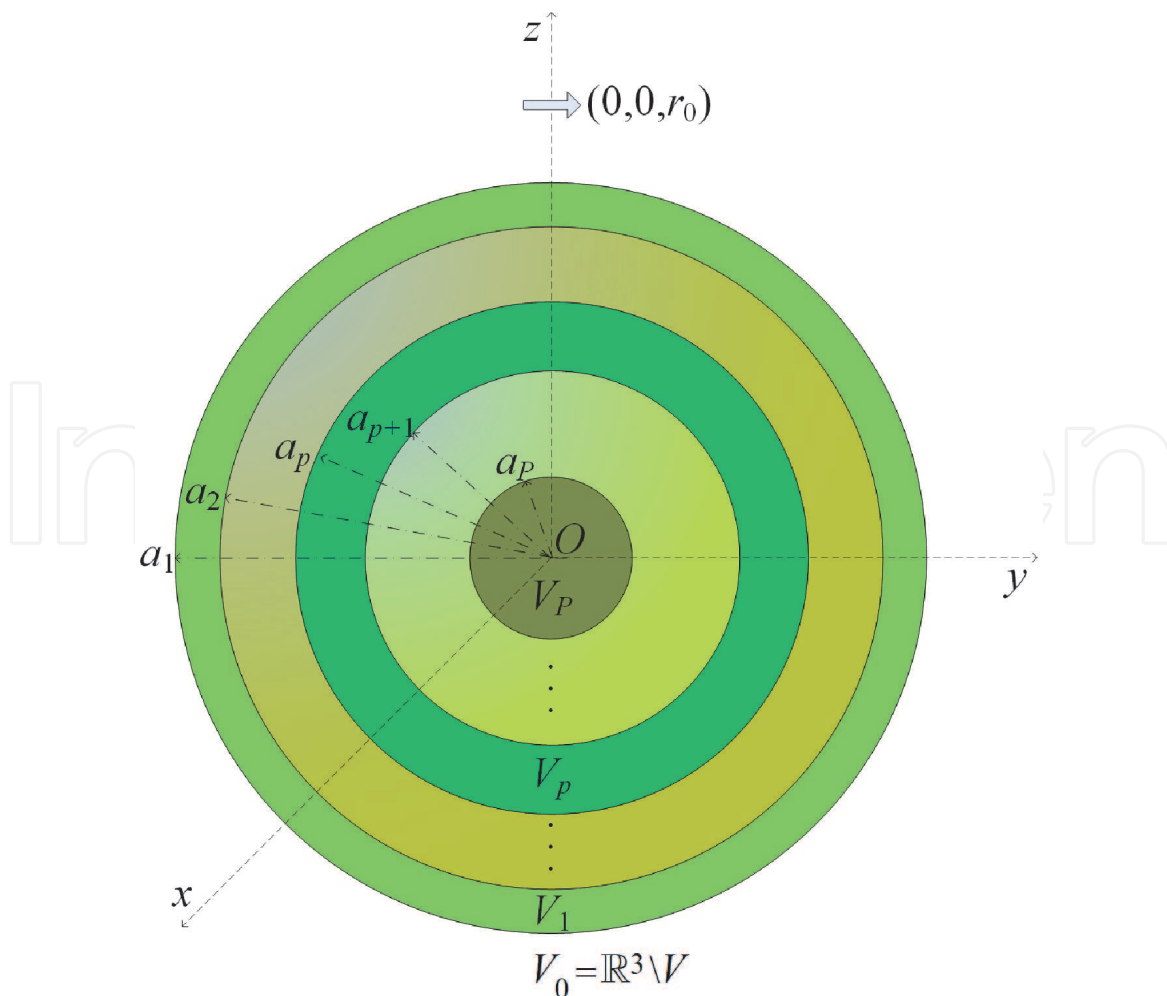


Figure 3. Geometrical configuration of the considered spherically-layered medium excited by an external dipole.

$$\begin{aligned}\sigma^{\dagger}(\mathbf{r}_0) &= \frac{1}{4\pi} \int_{S^2} \sigma(\theta, \phi; \mathbf{r}_0) \mathrm{d}s(\hat{\mathbf{r}}) \\ &= \frac{2\pi}{k_0^2} \sum_{n=1}^{\infty} (2n+1) \left[|\gamma_n|^2 + |\delta_n|^2 \right],\end{aligned}\quad (13)$$

where S^2 denotes the unit sphere in \mathbb{R}^3 , and $\sigma(\theta, \phi; \mathbf{r}_0)$ is the bistatic (differential) scattering cross section given by

$$\sigma(\theta, \phi; \mathbf{r}_0) = \frac{4\pi}{k_0^2} \left[|S_{\theta}(\theta; \mathbf{r}_0)|^2 \cos^2 \phi + |S_{\phi}(\theta; \mathbf{r}_0)|^2 \sin^2 \phi \right], \quad (14)$$

while functions $S_{\theta}(\theta; \mathbf{r}_0)$ and $S_{\phi}(\theta; \mathbf{r}_0)$ are defined by

$$S_{\theta}(\theta; \mathbf{r}_0) = \sum_{n=1}^{\infty} \frac{(-1)^n (2n+1)}{\sqrt{n(n+1)}} \left[\delta_n \frac{P_n^1(\cos \theta)}{\sin \theta} - \gamma_n \frac{\partial P_n^1(\cos \theta)}{\partial \theta} \right], \quad (15)$$

$$S_{\phi}(\theta; \mathbf{r}_0) = \sum_{n=1}^{\infty} \frac{(-1)^n (2n+1)}{\sqrt{n(n+1)}} \left[\gamma_n \frac{P_n^1(\cos \theta)}{\sin \theta} - \delta_n \frac{\partial P_n^1(\cos \theta)}{\partial \theta} \right], \quad (16)$$

with P_n^1 the first-order Legendre function of degree n , and

$$\gamma_n = \frac{h_n(k_0 r_0)}{h_0(k_0 r_0)} i^n \alpha_n, \quad \delta_n = \frac{\hat{h}'_n(k_0 r_0)}{\hat{h}_0(k_0 r_0)} i^{n-1} \beta_n, \quad (17)$$

where h_n is the spherical Hankel function of order n , and $\hat{h}_n(z) = zh_n(z)$. The coefficients α_n and β_n are defined in [24].

The objective function we consider in the optimization schemes is the *normalized total scattering cross section* $\sigma^{\dagger}(\mathbf{r}_0) / (\pi a_{\text{PEC}}^2)$, where a_{PEC} is the radius of the PEC sphere to be cloaked when covered by suitable coating magneto-dielectric layers. Achieving small values of this objective function provides efficient designs in terms of significant reductions in the scattered far-field. In [27], the backscattering cross section $\sigma(\theta, 0; \mathbf{r}_0)$ was used as the objective function. The latter can yield efficient designs only in traditional monostatic scenarios, while the present consideration of the total scattering cross section as the objective function shows the actual scattered far-field's characteristics for all observation angles.

For the numerical solution of the scattering problem, we used the code developed in [24], which is valid for an arbitrary number P of layers. The above-described PSO algorithms were implemented in MATLAB[®]. The swarms were MATLAB structs or arrays for which we followed the steps of Algorithms 1 or 2 presented above. The components of the position vector consisted of the optimization variables a_p of the radii, ε_p of the dielectric permittivities, and μ_p of the magnetic permeabilities of the first $P-1$ dielectric layers. The radius a_p of the PEC core was chosen constant at $k_0 a_p = k_0 a_{\text{PEC}} = 2\pi$ (one free-space wavelength). In this way, for a medium with P layers, the number of optimization variables for the particles position is $3(P-1)$.

The conducted experiments focused on small values of P in order to obtain designs with a relatively small number of coating layers, which also facilitate the fabrication procedure. For the variations of the variables of the optimization problem, different ranges were considered. Particularly, the differences $k_0(a_{p+1} - a_p)$ between two consecutive layers radii were considered in $[\frac{\pi}{10}, \pi]$ or $[\frac{\pi}{10}, \frac{\pi}{2}]$, while the values of the permittivities ε_p and permeabilities μ_p in $[0.5, 10]$, $[0.4, 5]$ or $[0.5, 5]$.

The external magnetic dipole was taken at $r_0 = 5a_{\text{PEC}}$. The two above-described particle swarm optimization algorithms were developed to minimize the normalized total scattering cross section for a spherical medium with $P = 3$ or 4 total number of layers. The actual reductions in the far-field with respect to the angles of observation are demonstrated in **Figures 4** and **5**, depicting the normalized bistatic scattering cross sections $\sigma(\theta, \phi; \mathbf{r}_0) / (\pi a_{\text{PEC}}^2)$ versus the angle θ in the xOz and yOz planes, respectively. In these figures, the corresponding cross section curves for a bare (containing no coating layers) PEC sphere are also shown, for comparison purposes.

Significant reductions in the far-field contributions with respect to the bare PEC sphere are observed for large ranges of the observation angles. Particularly, the CAPSO algorithm determines optimal variables corresponding to notably smaller objective function's values for a wide range of observation angles than the classic PSO algorithm. Moreover, the improved performance of the CAPSO algorithm is exhibited by the fact that the attained solutions yield reduced scattered far-field's values for all angles in the yOz plane and for nearly all angles in the xOz plane (apart from a resonance region of the bare PEC cross sections curves around $\theta = 140^\circ$). Another interesting conclusion is that the optimal solutions for $P = 3$ (two covering layers) generate smaller-in-general-far field's values for a wider angular range than the optimal solutions for $P = 4$ (three covering layers).

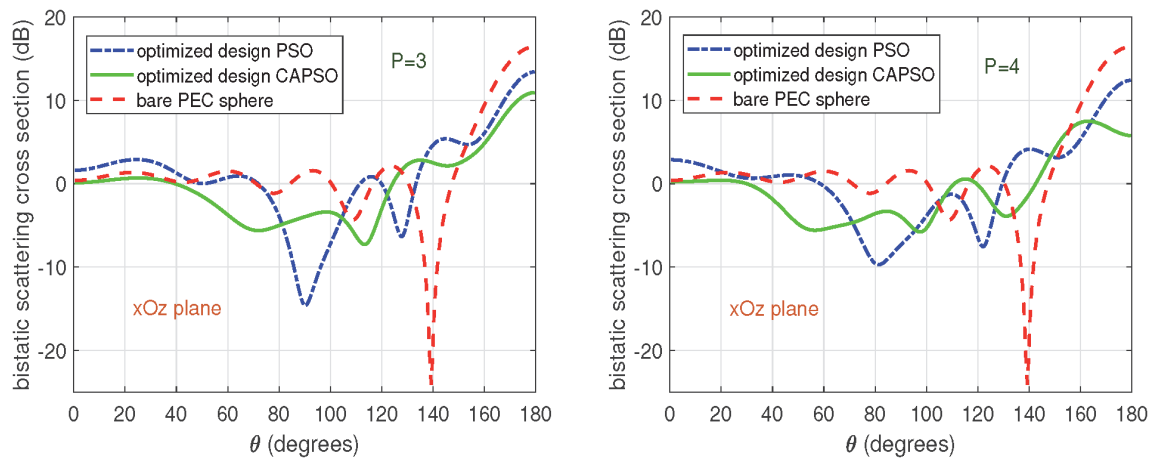


Figure 4. Normalized bistatic cross section in the xOz plane versus the angle θ for $P = 3$ (left panel) and $P = 4$ (right panel) optimized layers with parameters computed by the classic PSO and the CAPSO algorithms.

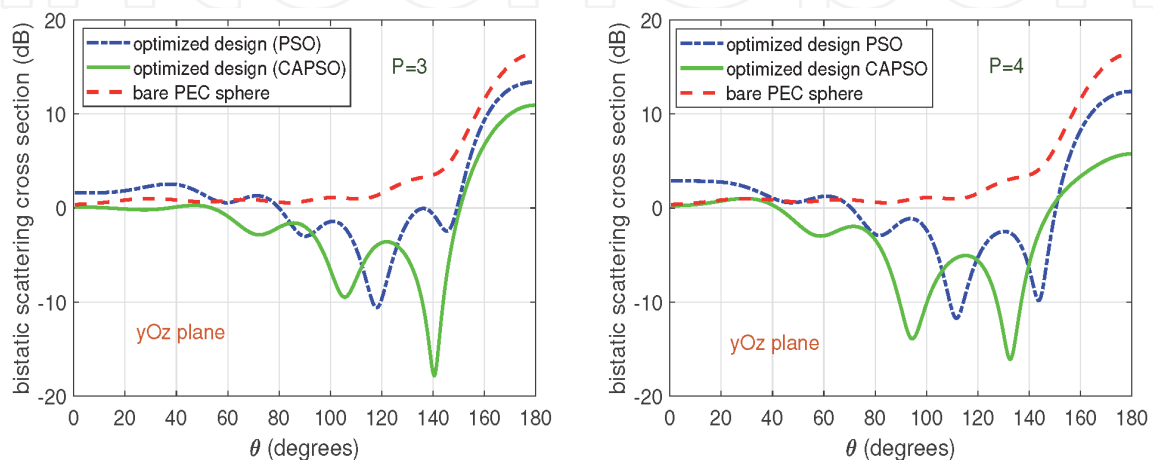


Figure 5. As in **Figure 4**, but for the normalized bistatic cross section in the yOz plane.

Besides, the effectiveness of the cloaking performance of the layered medium with respect to variations of the dipole's distance from the external boundary $r = a_1$ of the medium as well as the sensitivity of the results versus inevitable fabrications imperfections are also important to be examined. Some preliminary numerical results to this direction were presented in [28] by applying the classic PSO algorithm. Extensions to spherical antennas [29] and inhomogeneous media [30] can also be considered by modifying and extending the algorithms presented in this work.

4. Conclusions

Since its introduction to the scientific community, particle swarm optimization (PSO) has gone through many enhancements and variants, and has been applied to numerous diverse problems. The particles that compose the swarm's population act in a manner that follows the basic principles of Swarm Intelligence, as presented in literature. The algorithms utilize the intelligent swarm in order to discover the optima of objective functions. In this chapter, two algorithms were described. The PSO algorithm (1998 version), and the CAPSO algorithm which is a variant of the APSO algorithm. In the PSO, particles move with respect to Newton's laws of motion, and they are described by both position and velocity. Particles' position and velocity updates are affected by the global best \mathbf{g}^* at the time and the individual best \mathbf{x}^* . The algorithm includes their learning rates, adjusted in a manner that ensures equal weights to social and individual learning. An inertia mechanism is added to prevent the particles from moving too quickly, thus missing the discovery of optimal solutions. In contrast, the CAPSO algorithm particles do not keep memory of an individual best. They follow a more simplistic approach and update their position in a single step, affected only by the global best at the time. However, there are two parameters, α and β to fine-tune the swarms movement and insert necessary randomness. In CAPSO, the very crucial attraction parameter β , updates through chaotic maps. Specifically, in this work, the Sinusoidal map and the Singer map were considered and applied. It is noted that these maps have a unimode centered around their middle, and have provided the best results in relative research and testing. Both of the discussed algorithms were also provided in pseudocode format.

The PSO and CAPSO algorithms were developed and tested for cloaking problems concerning the covering of a perfectly conducting core by a number of coating layers with optimal parameters so that the total scattered field is significantly reduced. The resulting scattering performance of the medium was examined and it was demonstrated that both PSO and CAPSO algorithms are effective in achieving the goal of the scattered field reduction. Particularly, the CAPSO was shown to be successful in determining optimal solutions yielding enhanced cloaking behavior for a notably large range of the observation angles.

It is noted that the developed algorithms do not utilize a population topology mechanism since the global best is well known to all particles. Thus, in future research, alternative variants of these algorithms could be explored, for example the SPSO 2011 [31] or the Adaptive Clan PSO [32].

Conflict of interest

The authors declare no conflict of interest.

Abbreviations

PSO	Particle Swarm Optimization
APSO	Accelerated Particle Swarm Optimization
CAPSO	Chaotic Accelerated Particle Swarm Optimization
PEC	Perfect Electric Conducting

IntechOpen

IntechOpen

Author details

Alkmini Michaloglou[†] and Nikolaos L. Tsitsas^{*†}
School of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

*Address all correspondence to: ntsitsas@csd.auth.gr

[†] These authors contributed equally.

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Kennedy J, Eberhart R. Particle swarm optimization. In: Proceedings of ICNN'95-international conference on neural networks 1995 Nov 27 (Vol. 4, pp. 1942-1948). IEEE.
- [2] Heppner F, Grenander U. A stochastic nonlinear model for coordinated bird flocks. *The ubiquity of chaos*. 1990;233:238.
- [3] Shi Y, Eberhart RC. Parameter selection in particle swarm optimization. In: International conference on evolutionary programming 1998 Mar 25 (pp. 591-600). Springer, Berlin, Heidelberg.
- [4] Gandomi AH, Yun GJ, Yang XS, Talatahari S. Chaos-enhanced accelerated particle swarm optimization. *Communications in Nonlinear Science and Numerical Simulation*. 2013 Feb 1;18(2):327-40.
- [5] Robinson J, Rahmat-Samii Y. Particle swarm optimization in electromagnetics. *IEEE Transactions on Antennas and Propagation*. 2004 Apr 5; 52(2):397-407.
- [6] Jin N, Rahmat-Samii Y. Particle swarm optimization for antenna designs in engineering electromagnetics. *Journal of Artificial evolution and applications*. 2008 Mar 30;2008.
- [7] Mikki SM, Kishk AA. Quantum particle swarm optimization for electromagnetics. *IEEE Transactions on Antennas and Propagation*. 2006 Oct 9; 54(10):2764-75.
- [8] Mikki SM, Kishk AA. Physical theory for particle swarm optimization. *Progress In Electromagnetics Research*. 2007;75:171-207.
- [9] Goudos SK, Zaharis ZD, Baltzis KB. Particle swarm optimization as applied to electromagnetic design problems. *International Journal of Swarm Intelligence Research (IJSIR)*. 2018 Apr 1;9(2):47-82.
- [10] Alù A, Engheta N. Achieving transparency with plasmonic and metamaterial coatings. *Physical Review E*. 2005 Jul 26;72(1):016623.
- [11] Alù A, Engheta N. Multifrequency optical invisibility cloak with layered plasmonic shells. *Physical Review Letters*. 2008 Mar 18;100(11):113901.
- [12] Qiu CW, Hu L, Zhang B, Wu BI, Johnson SG, Joannopoulos JD. Spherical cloaking using nonlinear transformations for improved segmentation into concentric isotropic coatings. *Optics Express*. 2009 Aug 3;17(16):13467-78.
- [13] Castaldi G, Gallina I, Galdi V, Alù A, Engheta N. Analytical study of spherical cloak/anti-cloak interactions. *Wave Motion*. 2011 Sep 1;48(6):455-67.
- [14] Martins TC, Dmitriev V. Spherical invisibility cloak with minimum number of layers of isotropic materials. *Microwave and Optical Technology Letters*. 2012 Sep;54(9):2217-20.
- [15] Wang X, Chen F, Semouchkina E. Spherical cloaking using multilayer shells of ordinary dielectrics. *AIP Advances*. 2013 Nov 12;3(11):112111.
- [16] Ladutenko K, Peña-Rodríguez O, Melchakova I, Yagupov I, Belov P. Reduction of scattering using thin all-dielectric shells designed by stochastic optimizer. *Journal of Applied Physics*. 2014 Nov 14;116(18):184508.
- [17] Campbell SD, Sell D, Jenkins RP, Whiting EB, Fan JA, Werner DH. Review of numerical optimization techniques for meta-device design.

- Optical Materials Express. 2019 Apr 1;9 (4):1842-63.
- [18] Wang D, Tan D, Liu L. Particle swarm optimization algorithm: an overview. *Soft Computing*. 2018 Jan;22 (2):387-408.
- [19] Rao SS. *Engineering optimization: theory and practice*. John Wiley & Sons; 2009 Jul 20.
- [20] Millonas MM. *Swarms, phase transitions, and collective intelligence*. arXiv preprint [adap-org/9306002](https://arxiv.org/abs/1906.0002). 1993 Jun 11.
- [21] Yang XS. *Nature-Inspired Optimization Algorithms*. Elsevier; 2014 Feb 17.
- [22] Sprott JC. *Chaos From Euler Solution of ODEs*. Oxford University Press; 2003, pp. 63-65.
- [23] Lu H, Wang X, Fei Z, Qiu M. The effects of using chaotic map on improving the performance of multiobjective evolutionary algorithms. *Mathematical Problems in Engineering*. 2014 Feb;2014.
- [24] Tsitsas NL, Athanasiadis C. On the scattering of spherical electromagnetic waves by a layered sphere. *The Quarterly Journal of Mechanics and Applied Mathematics*. 2006 Feb 1;59(1): 55-74.
- [25] Tsitsas NL. Direct and inverse dipole electromagnetic scattering by a piecewise homogeneous sphere. *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik*. 2009 Oct 1;89(10):833-49.
- [26] Prokopiou P, Tsitsas NL. Electromagnetic excitation of a spherical medium by an arbitrary dipole and related inverse problems. *Studies in Applied Mathematics*. 2018 May;140 (4):438-64.
- [27] Tsitsoglou Z, Prokopiou P, Tsitsas NL. Dipole-Scattering by Spherical Media and Related Optimization Problems. In: 2018 2nd URSI Atlantic Radio Science Meeting (AT-RASC) 2018 May 28 (pp. 1-4). IEEE.
- [28] Michaloglou A, Tsitsas NL. Particle Swarm Optimization of Layered Media Cloaking Performance. *URSI Radio Science Letters*. 2020; 2: (5 pages) DOI: 10.46620/20-0016.
- [29] Valagiannopoulos CA, Tsitsas NL. On the resonance and radiation characteristics of multi-layered spherical microstrip antennas. *Electromagnetics*. 2008 May 27;28(4): 243-64.
- [30] Valagiannopoulos CA, Tsitsas NL. Linearization of the T-matrix solution for quasi-homogeneous scatterers. *Journal of the Optical Society of America A*. 2009 Apr 1;26(4):870-81.
- [31] Zambrano-Bigiarini M, Clerc M, Rojas R. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In: 2013 IEEE Congress on Evolutionary Computation 2013 Jun 20 (pp. 2337-2344). IEEE.
- [32] Pontes MR, Neto FB, Bastos-Filho CJ. Adaptive clan particle swarm optimization. In: 2011 IEEE Symposium on Swarm Intelligence 2011 Apr 11 (pp. 1-6). IEEE.