A Brief Historical Review of Particle Swarm Optimization (PSO)

Article · June 2012

DOI: 10.1166/jbic.2012.1002

1 author:

Juan Luis Fernández-Martínez
University of Oviedo

Journal of Bioinformatics and Intelligent Control
Vol. 1, 3–16, 2012

REVIEW

A Brief Historical Review of Particle Swarm Optimization (PSO)

E. García-Gonzalo and J. L. Fernández-Martínez*

Mathematics Department, Universidad de Oviedo, 33007 Oviedo, Spain

Particle Swarm Optimization is an evolutionary algorithm that has been applied to many different engineering and technological problems with considerable success. Since its first publication in 1995, it has been continually modified trying to improve its convergence properties. Thus, many variants have been proposed in the literature. Some of these variants were related to a particular problem and had little application outside the field where they have been proposed. Others have been used for solving different kind of problems and have enjoyed a longer life. These PSO variants have been used to solve a wide range of optimization and inverse problems: continuous, discrete, dynamical, multioptima, combinatorial, with and without additional constraints. In this paper we briefly review the history of Particle Swarm Optimization, insisting in the importance of the stochastic stability analysis of the particle trajectories in order to achieve convergence.

Keywords: Particle Swarm Optimization, PSO Variants, Stability Analysis, Convergence.

CONTENTS

1. INTRODUCTION

Particle Swarm Optimization (PSO) was first proposed in 1995.[28] It was originated in social modeling, being one of its fathers, James Kennedy, a social psychologist. The first PSO algorithm was closely related to the graphic animation of flocks.[21,40] With time, PSO has been further developed, modified, and successfully applied for optimization in many engineering problems and technological fields.

Although the PSO algorithm seems very simple to implement, one of the most interesting questions from the beginning was understanding its convergence properties and avoiding its numerical instabilities. This fact has motivated the proposal of many PSO variants, most of them based on numerical experiments or heuristic modifications. In this paper we describe, the origin, the PSO development and the actual state of art. All these modifications tried to improve the PSO performance, one of the most important achievements was the stochastic stability analysis of the PSO trajectories[17,22,24,38,52] and the use of physical models to understand the PSO swarm dynamics.[4,15,33] Both approaches served to dramatically improve the knowledge about how PSO works. In conclusion, nowadays PSO should be considered as a stochastic algorithm with a well established theoretical background. Thus, PSO should not be considered a heuristic algorithm anymore.

1.1. Bio-Inspired Computing and Evolutionary Algorithms

Natural and biologically inspired computing relates to the algorithms that are based on certain natural phenomena, arising from the connection between biology, computer science, artificial intelligence and applied mathematics. In the field of optimization, these algorithms try to provide

*Author to whom correspondence should be addressed.

2326-7496/2012/1/003/014doi:10.1166/jbic.2012.1002

REVIEW

a solution to the difficulties that appear in certain technological problems, such as ill-posed inverse problems with noisy data, non-convex and non differentiable optimization, etc., that cannot be easily tackled by traditional optimization methods.[1] These algorithms are based in biological models, and using some simple rules, they generate a cooperative and/or competitive behavior, trying to reach some optimal solution (or solutions) for a given problem. A subgroup of the bio-inspired algorithms are the evolutionary algorithms. These are based on populations that evolve with time and have a stochastic component aimed at escaping from entrapment in local optima. Examples of such algorithms are Genetic Algorithms,[5] Differential Evolution,[46] Ant Colony System[8] and Particle Swarm Optimization.[28] In the case of nonlinear inverse problems these optima lie in flat elongated valleys of the cost function topography.[14] In this case, these algorithms can be also used to explore the cost function landscape, if the fitness (or data misfit) of the members of the population (particles) is fast to compute. In any case, *No Free Lunch* theorem[50] applies, since there is not a best universal algorithm over all the possible problems: the performance of the algorithms is highly dependent on the objective function landscape.[14]

1.2. The PSO Predecessor

Reynolds[40] developed a method to animate a flock of *boids*, bird like objects, flying. It was based on simple laws: each boid would be aware only of its immediate proximity or neighborhood; it would avoid collision with other boids; it would try to move according to an average velocity of its neighbors; and it would not leave the flock. Kennedy and Eberhart modified these rules and proposed the PSO algorithm.[28] For instance, they suppressed the no-collision rule in order to apply the new algorithm to optimization problems, using a mathematical function as fitness function for the each individual of the flock (particle). With these modifications, they estimated that the behavior of the group resembled more that of a swarm than of a flock, thus, the new algorithm was called Particle Swarm Optimization.

1.3. The Original PSO Version

Particle Swarm Optimization can be considered as an evolutionary computation technique inspired by the social behavior of individuals in groups in nature. The behavior of any individual in the swarm is influenced by its own experience and that of its neighbors.

In an optimization problem, an individual or particle, i, is represented by a vector whose length D is the dimension of the fitness function domain. As first step, a population of N particles is initialized with random positions $\mathbf{x}_i^0$ and velocities $\mathbf{v}_i^0$. The fitness function $f(\mathbf{x})$ is evaluated for each particle. In each iteration, the positions and velocities of each particle are updated using the value of the fitness function. At iteration $k+1$, the algorithm updates

Esperanza García-Gonzalo received the Bachelor's degree in mining engineering in 1990 and the Ph.D. degree in 2011, both from the University of Oviedo, Spain. She is Associate Professor with the Department of Mathematics, University of Oviedo. Her current research interests include global stochastic optimization methods, particularly particle swarm optimization and their application to inverse problems in geophysics and reservoir engineering.

Juan Luis Fernández-Martínez received his bachelor degree in 1986, and his Ph.D. degree in 1994, both in mining engineering from the University of Oviedo, Spain. He specialized as a Petroleum Engineer with École Nationale du Pétrole et des Moteurs, Paris, France (1988), and with the Royal School of Mines, Imperial College, London, U.K. (1989). After working as a computing and consulting engineer in France for a few years in the petroleum industry, he joined the Department of Mathematics, University of Oviedo, in 1994, where he is currently Associate Professor in Applied Mathematics. He works in inverse problems and stochastic optimization. From 2008 to 2010, he was a Visiting Professor with the Department of Civil and Environmental Engineering, University of California Berkeley, Berkeley, and with the Department of Energy Resources, Stanford University, Palo Alto, CA, where he worked on inverse problems and uncertainty quantification applied to reservoir optimization. He is also very interested in machine learning applied to the biomedical field (cancer, genomics and proteomics) and in digital imaging processing and biometric applications.

positions $\mathbf{x}_i^{k+1}$ and velocities $\mathbf{v}_i^{k+1}$ of the individuals as follows:

$$\begin{aligned}\mathbf{v}_i^{k+1} &= \mathbf{v}_i^k + \phi_1^k(\mathbf{g}^k - \mathbf{x}_i^k) + \phi_2^k(\mathbf{l}_i^k - \mathbf{x}_i^k),\\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1}\end{aligned} \tag{1}$$

with $i = 1, \ldots, N$ and $k = 0, \ldots, N_{\text{iter}}$, and

$$\phi_1^k = r_1^k a_g, \quad \phi_2^k = r_2^k a_l$$
$$a_l, a_g \in \mathbb{R}, \quad r_1^k, r_2^k \to U(0,1)$$

where a_g and a_l are real constants, called global and local accelerations, and r_1^k and r_2^k random variables uniformly distributed in $[0,1]$. These random variables cause the algorithm to be stochastic. The fact that

$$\phi_1^k \to U(0, a_g) \quad \text{and} \quad \phi_2^k \to U(0, a_l)$$

originated that the notation for these variables was simplified to ϕ_1 and ϕ_2, avoiding their implicit dependency on the iterations.

As we can see in relation (1), the velocity of any particle i of the swarm at iteration $k+1$ is a function of three major components:

- The previous velocity term, $\mathbf{v}_i^k$.
- The cognitive learning term, which is the difference between each particle's best position so far found (called $\mathbf{l}_i^k$) and the particle current position $\mathbf{x}_i^k$. Thus, in the case of a minimization problem, the local best position is defined as follows:

$$\mathbf{l}_i^{k+1} = \begin{cases}\mathbf{l}_i^k & \text{if } f(\mathbf{x}_i^{k+1}) > f(\mathbf{l}_i^k),\\ \mathbf{x}_i^{k+1} & \text{if } f(\mathbf{x}_i^{k+1}) \le f(\mathbf{l}_i^k),\end{cases} \quad i = 1, \ldots, N$$

- The social learning term, which is the difference between the global best position so far found in the entire swarm (called $\mathbf{g}^k$) and the particle's current position $\mathbf{x}_i^k$. The global best is defined as follows (for a minimization problem):

$$\mathbf{g}^{k+1} = \begin{cases}\mathbf{g}^k & \text{if } f(\mathbf{x}_i^{k+1}) > f(\mathbf{g}^k),\\ \mathbf{x}_i^{k+1} & \text{if } f(\mathbf{x}_i^{k+1}) \le f(\mathbf{g}^k),\end{cases} \quad \forall i = 1, \ldots, N$$

This last relationship implies elitism since the global best position is not updated unless it improves its fitness. Generally, $\mathbf{g}^k$ only coincides with a particle of the swarm when its position is updated. Otherwise, $\mathbf{g}^k$ is an attractor that is related to the particle swarm history.

Other approach would consist in updating $\mathbf{g}^k$ in each iteration, taking only into account the particles of the actual iteration.

$$\mathbf{g}^k = \mathbf{x}_{i_g}^k : \quad f(\mathbf{x}_{i_g}^k) = \min_{1 \le i \le N} f(\mathbf{x}_i^k)$$

In this case the global best always coincides with the particle in the swarm that has the lowest misfit in the current iteration. Thus, the algorithm is, generally, more exploratory since $\mathbf{g}^k$ is updated even if the algorithm worsens (fitness decreases). In any case, these two components (cognitive and social) are stochastically weighted by uniform random numbers in the interval $[0,1]$, r_1^k and r_2^k, affecting the local and global acceleration, a_l and a_g, and causing the particle trajectories to oscillate at each iteration around the position

$$\mathbf{o}_i^k = \frac{\phi_1^k \mathbf{g}^k + \phi_2^k \mathbf{l}_i^k}{\phi_1^k + \phi_2^k}, \quad i = 1, \ldots, N \tag{2}$$

which is named particle oscillation center.

The ratio between the stochastic accelerations ϕ_1^k and ϕ_2^k affect the balance between local and global search (exploration and exploitation). This algorithm can be rewritten only in terms of positions as second order stochastic difference equation, as follows:

$$\begin{aligned}\mathbf{v}_i^k &= \mathbf{x}_i^k - \mathbf{x}_i^{k-1}\\ \mathbf{v}_i^{k+1} &= \mathbf{v}_i^k + \phi_1^k(\mathbf{g}^k - \mathbf{x}_i^k) + \phi_2^k(\mathbf{l}_i^k - \mathbf{x}_i^k)\\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1}\end{aligned}$$

Then

$$\mathbf{x}_i^{k+1} + (\phi_1^k + \phi_2^k - 2)\mathbf{x}_i^k + \mathbf{x}_i^{k-1} = \phi_1^k \mathbf{g}^k + \phi_2^k \mathbf{l}_i^k \tag{3}$$

The stability condition for the first order moments of the trajectories (mean trajectories) implies that the roots of the characteristic equation

$$r^2 + (\overline{\phi} - 2)r + 1 = 0, \quad \overline{\phi} = \frac{a_g + a_l}{2}$$

lie within the unit circle. This provides the first order stability condition $0 < \overline{\phi} < 4$. Nevertheless, the mean trajectories are oscillatory waves with constant amplitude around (2) and the swarm might not converge (or collapse towards) to the global optimum. The stability analysis of the trajectories for a simple one-dimensional PSO system corresponding to (1) was performed in Ref. [36], under the suggestive title "PSO: Surfing the waves." This analysis did not take into account the introduction of the additional parameter, called the inertia weight, introduced by Shi and Eberhart.[43] The stability analysis of the PSO algorithm with inertia weight it will be shown in Section 4.1 and it will generalize the case $\omega = 1$.

2. THE PSO ALGORITHM EVOLUTION

Initially the PSO behavior was analyzed by means of numerical experiments. See for instance.[6,26,44] As pointed out before, in the initial PSO version without the inertia weight, the particles were stable if the mean total acceleration, $\overline{\phi}$, was chosen between 0 and 4. Nevertheless the trajectories were oscillatory. Mean trajectories of the particles were unstable for $\overline{\phi}$ values outside this interval. Thus,

REVIEW

REVIEW

the initial PSO modifications were aimed at maintaining the particles inside the search space, providing swarm stability and also convergence to the global optimum for different benchmark functions, expecting that these results will apply to real life problems.

When the stability problem was overcome, a fast convergence to a local optimum (or any other position in the search space) was pointed out as an additional issue for the PSO algorithm. This led to the phenomenon called premature convergence that, as we will show, it is in fact due to a poor understanding of the swarm dynamics: the PSO swarm can be physically interpreted as a damped mass-spring system; and choosing the PSO parameters (ω, a_g, a_l) inside the first order stability region implies that the mean trajectories of the swarm will collapse towards the oscillation center. Thus, the swarm collapse cannot be avoided if some additional mechanisms are not implemented to disperse the swarm.

2.1. Velocity Clamping

The first idea that was used to achieve stability, consisted in preventing the particles escaping from the search space by limiting their velocity. This mechanism was named velocity clamping[12] and it was very important in the original PSO version if $\overline{\phi} > 4$. In this case, the mean trajectories were not stable and the particles might escape the search space. Velocity clamping also served to avoid particles hitting the limits of the search space. This occurs when particles follow non stable dynamics or when the total position perturbation, v_{ij}^{k+1}, is not related to the size of the search space in the j direction. Velocity clamping works as follows: if the velocity of any of the N particles of the swarm exceeds a maximum value $v_{\max}$ (that is related to the length of the search space D in each coordinate j), then:

$$v_{ij}^{k+1} = \begin{cases} v_{\max,j} & \text{if } v_{ij}^{k+1} > v_{\max,j} \quad i = 1, \ldots, N \\ v_{ij}^{k+1} & \text{if } v_{ij}^{k+1} \le v_{\max,j} \quad j = 1, \ldots, D \end{cases}$$

The $v_{\max}$ parameter is sometimes defined as a constant that has to be tuned for every individual optimization problem. Other approach consists in defining every component of $v_{\max}$ as a percentage (for instance 5%) of the total length of the search space in each coordinate. In any case, the introduction of $v_{\max}$ implied one additional parameter to be tuned. This parameter was not really needed in posterior PSO versions, provided that a correct stochastic stability analysis of the swarm was performed.[17]

Velocity clamping is related with the position perturbation in each iteration and it does not guarantee that the particles remain inside the search space. This can be achieved by means of a projection operator: after updating the position

$$x_{ij}^{k+1} = x_{ij}^{k} + v_{ij}^{k+1}$$

if the particle is outside the search space, we can project it back, for instance, by using:

$$x_{ij}^{k+1} = \begin{cases} u_j & \text{if } x_{ij}^{k+1} > u_j \\ b_j & \text{if } x_{ij}^{k+1} < b_j \\ x_{ij}^{k+1} & \text{if } b_j \le x_{ij}^{k+1} \le u_j \end{cases}$$

where b_j and u_j are the lower and the upper bounds of the search space for the j component. Other ways of projecting can be also defined but are less intuitive.

It is also important to control the number of particles hitting the borders of the search space with iterations. It is possible to observe that at the exploration stage (first iterations), the number of particles in the interior of the search space might be very low until a good position is found (see Fig. 1).

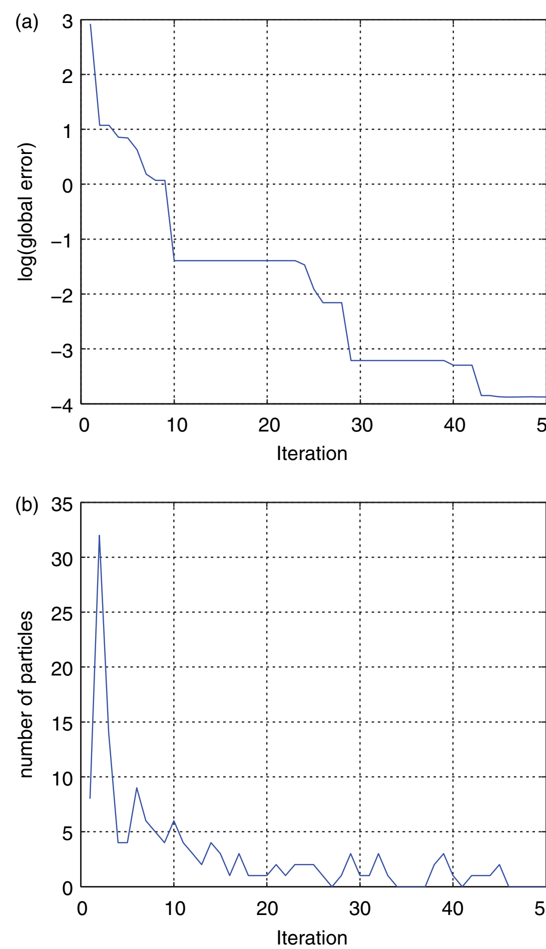

Fig. 1. (a) Global best error (minimization problem) in logarithmic scale. (b) Number of particles outside the search space for a PSO run using the Rosenbrock function in 2 dimensions, with 100 particles, $(\omega, a_g, a_l) = (0.6, 1.7, 1.7)$, search space $[-30, 30]^2$ and 50 iterations.

2.2. The Introduction of the Inertia Weight

The inertia weight ω was introduced[43] to modify the stability condition of the PSO algorithm. The formulation became:

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \omega \mathbf{v}_i^k + \phi_1^k(\mathbf{g}^k - \mathbf{x}_i^k) + \phi_2^k(\mathbf{l}_i^k - \mathbf{x}_i^k), \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{aligned} \tag{4}$$

Now, it is possible to observe that, if no other particles improves its fitness, the global best $\mathbf{g}^k$ follows:

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \omega \mathbf{v}_i^k \\ \mathbf{g}_i^{k+1} &= \mathbf{g}_i^k + \omega \mathbf{v}_i^k \end{aligned}$$

Thus, its movement will be only affected by the inertia constant ω.

The introduction of the inertia weight was aimed at achieving a balance between the local and global abilities of the PSO. Large inertia values facilitates global search (exploration of the search space). In the case $\omega = 1$ we reach the previous PSO version (1). With the introduction of ω the first order stability region became now,

$$\{(\omega, \overline{\phi}) : |\omega| < 1,\ 0 < \overline{\phi} < 2(\omega + 1)\} \tag{5}$$

This region has been deduced by different researchers.[13,48] Some analysis restrict ω to the interval (0, 1), which is, in fact, incorrect.

Choosing $(\omega, \overline{\phi})$ within this stability region implies that the mean trajectories converge asymptotically to a fixed point (the oscillation center mean):

$$E(\mathbf{x}_i^k) \underset{k\to\infty}{\longrightarrow} E(\mathbf{o}_i^k) \quad \forall i = 1, \ldots, N,$$

being

$$E(\mathbf{o}_i^k) = \frac{a_g E(\mathbf{g}^k) + a_l E(\mathbf{l}_i^k)}{a_g + a_l}$$

It is possible to analyze individually the effect of any of the terms in the velocity update. For instance, Kennedy and Eberhart[28] pointed out that the simplified algorithm where the term $\mathbf{v}_i^k$ has been suppressed did not perform as good as the complete version. Nevertheless it is important to remark that in this case the first order stability condition changes. Now, the difference equation (3) becomes

$$\mathbf{x}_i^{k+1} + (\phi_1^k + \phi_2^k - 1)\mathbf{x}_i^k = \phi_1^k \mathbf{g}^k + \phi_2^k \mathbf{l}_i^k$$

and the characteristic equation providing the stability condition for the mean trajectories is

$$r^2 + (\overline{\phi} - 1)r = 0$$

Imposing the unit circle condition we arrive at the stability condition for the mean trajectories, $0 < \overline{\phi} < 2$. The mean trajectories are oscillatory if $1 < \overline{\phi} < 2$ and exponential if $0 < \overline{\phi} < 1$ as seen in Figure 2. As a main conclusion, the results provided by this simplified PSO algorithm cannot be compared to the complete PSO version if $2 \le \overline{\phi} < 4$, because now the mean trajectories are not stable and the simplified algorithm might not converge (collapse towards the global best).

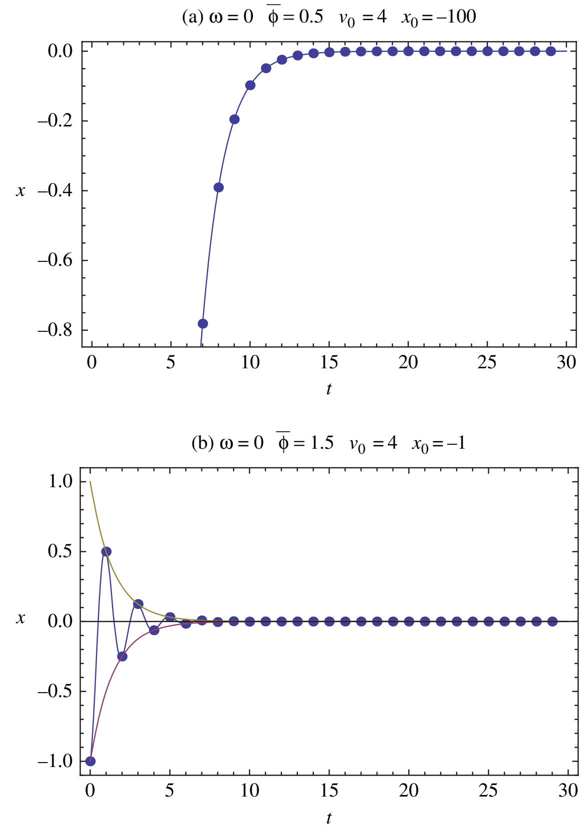

Fig. 2. Particle mean homogeneous trajectories (no attraction center) when $\omega = 0$. (a) Example for $0 < \overline{\phi} < 1$. (b) Example for $1 < \overline{\phi} < 2$.

Shi and Eberhart[43] observed that if the cognitive and social terms were suppressed, the particles would keep on flying in the same direction until they hit the boundary. The reason is that, in this case the mean trajectories are not stable because the greatest root of the characteristic equation of its difference equation

$$\mathbf{x}_i^{k+1} - (1 + \omega)\mathbf{x}_i^k + \omega \mathbf{x}_i^{k-1} = 0$$

is always equal to 1, and there is not attraction because the cognitive and social terms have been suppressed. In this case the algorithm might not converge.

In conclusion, as a simplified rule, the social and cognitive terms are responsible for attraction and convergence, while the velocity term is in charge of exploration.

Kennedy[25] also studied the influence of the social and cognitive terms alone combined with the velocity term. He defined the following cases:

- Cognition-only model:

$$\mathbf{v}_i^{k+1} = \omega \mathbf{v}_i^k + \phi_2^k(\mathbf{l}_i^k - \mathbf{x}_i^k)$$

REVIEW

- Social-only model:

$$\mathbf{v}_i^{k+1} = \omega \mathbf{v}_i^k + \phi_1^k (\mathbf{g}^k - \mathbf{x}_i^k)$$

The first order stability conditions for these two models is the same that for the complete PSO version (5), considering that $\overline{\phi} = \overline{\phi_i^k}$ where i is equal to 1 or 2 depending on the version. As a general rule the cognition-only model is more exploratory than the social one.

2.3. Clerc and Kennedy's Constriction Factor Model

Clerc and Kennedy[7] were one of the first in analyzing the stability and convergence of a simplified PSO version. In their analysis they introduced a constriction factor PSO version. The constriction factor $\chi \in \mathbb{R}$ allows to write the velocity and position updates as follows:

$$\begin{aligned} \mathbf{v}_i^{k+1} &= \chi(v_i^k + \varphi_1 r_1^k (\mathbf{g}^k - \mathbf{x}_i^k) + \varphi_2 r_2^k (\mathbf{l}_i^k - \mathbf{x}_i^k)), \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k + \mathbf{v}_i^{k+1} \end{aligned} \tag{6}$$

with

$$\varphi_1, \varphi_2 \in \mathbb{R}, \quad r_1^k, r_2^k \rightarrow U(0,1)$$

This constriction factor PSO version (6) can be seen as a particular case of (4) with

$$\omega = \chi, \quad a_g = \chi\varphi_1, \quad a_l = \chi\varphi_2 \tag{7}$$

Clerc and Kennedy presented in their paper several constriction factors to achieve the swarm stability. Nevertheless, the following relationship between $\chi = \omega$ and $\varphi = \varphi_1 + \varphi_2 \in \mathbb{R}$

$$\chi(\varphi) = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad \text{with } \varphi > 4 \tag{8}$$

became the most popular. Relationship (8) defines a curve in the plane $(\omega, \overline{\phi})$

$$\begin{cases} \omega(\varphi) = \dfrac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \\ \overline{\phi}(\varphi) = \dfrac{\varphi}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}, \end{cases} \quad \varphi > 4$$

This curve lies within within the complex zone of the first order stability region of the particle trajectories where the exploration capabilities of PSO are very high. Also, based on this relationship (8) good parameter sets for (ω, a_g, a_l) were found,[6,11] and the clamping of the velocities was rendered unnecessary.

2.4. The Binary and Discrete PSO Versions

The original PSO version was designed to work with continuous variables having their range over $\mathbb{R}$. Thus it was not suitable to solve optimization problems where the control variables took values over $\mathbb{Z}$ (discrete) or even in the set $\{0, 1\}$ (binary variables).

In 1997 Kennedy and Eberhart[29] presented the binary PSO. To transform the continuous variables provided by PSO to the $[0, 1]$ interval they used a sigmoid function (see Fig. 3):

$$s(x_i) = \frac{1}{1 + e^{-x_i}}$$

Finally, a thresholding operator was used to project $s(x_i)$ to $z_i \in \{0, 1\}$ as follows:

$$z_i = \begin{cases} 1 & \text{if } s(x_i) > p_i, \\ 0 & \text{if } s(x_i) \leq p_i, \end{cases} \quad p_i \rightarrow U(0,1)$$

In 2002 Laskari et al.[31] showed that PSO with a slight modification can be used to solve discrete optimization problems, by working with continuous variables and rounding off the real optima to the nearest integer position.

2.5. PSO Numerical Experiments

Different papers have been devoted to analyze the role of the different PSO parameters by means of numerical experiments:

- *Swarm size*: different sizes have been proposed for different dimensions and problems. Nevertheless, there does not exist a standard rule to chose the number of particles depending on the dimension. Generally the number of particles increases with the dimension and is typically lower than 100. For example Trelea[48] used population sizes of $N = 15$, 30 and 60 particles for benchmark functions in dimension 30. The number of particles also depend on the computational cost needed to perform any individual fitness calculation. Thus, there exists a trade off between the maximum number of particles in the swarm and the maximum number of iterations if the computational cost is very high.

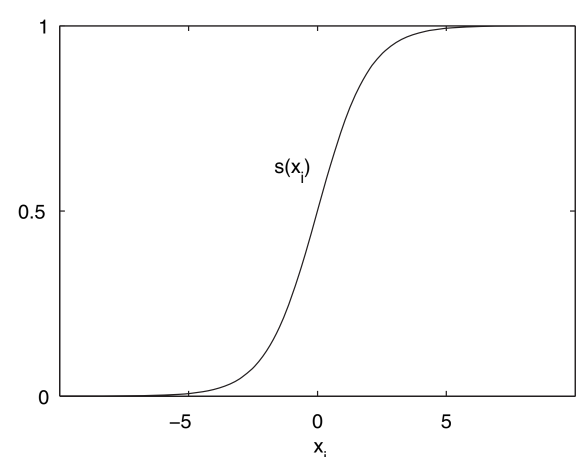

Fig. 3. Sigmoid curve.

• *Initial position*. Generally, the particles are randomly distributed in the search space using a uniform distribution. Nevertheless, particles could be distributed following other kind of statistical distributions or being drawn in smaller parts of the search space.[44]

• *Initial velocities and velocity update*. Different strategies have been used to initialize the velocities: random initial velocities, null initial velocities, or using the initial position as in standard PSO (3.2):

$$v_{ij}^0 = U(b_j - x_{ij}^0, u_j - x_{ij}^0)$$

Also, as it has been already shown, different velocity updates have been proposed: clamping, social-only model, cognition-only model, etc.

• *Selection of the updated particles*. Generally all the particle positions are updated in each iteration. Nevertheless, some numerical experiments only updated a part of the entire swarm. The criteria for the selection of the particles are variable: randomly selected, selecting only the best particles (trying to improve the convergence to the global optimum), or selecting only the particles with lowest misfit (trying to avoid premature convergence).

• *The law of variation for ω, a_g and a_l*. Once the stability analysis of the PSO trajectories has been performed, different criteria have been provided to choose the (ω, a_g, a_l) parameters inside the first order stability region (5). In the simplest version these parameters are constant for all the particles in the swarm and do not depend on the iterations. Nevertheless, in some approaches ω can vary linearly[44,51] or nonlinearly[20,49] with iterations. Typically, the inertia is initially set to a maximum value close to the upper border of first order stability region, $\omega = 1$, and is decreased (linearly or nonlinearly) with iterations till a minimum value. This approach is similar to the temperature cooling in simulated annealing: when the temperature (or the inertia) is high the algorithm is more exploratory; conversely when the temperature (or the inertia) decreases the global best has a lower probability to be changed. Different inertia approaches have been proposed in the literature.[3,32] Also the parameters a_g and a_l can vary with iterations. See, for instance.[35] Using different values of $a_g$and a_l for the same $\overline{\phi}$ serves to increase the exploitation if $a_g > a_l$, and conversely, increasing the exploration if $a_l > a_g$.[6] Also, the PSO cloud versions[19] has been proposed where each particle (or coordinate component) have a different set of (ω, a_g, a_l) that are located close to the upper border of the second order stability region where the exploration is very high. Usually the best (ω, a_g, a_l) points belong to the complex zone of the first order stability region where the particle mean trajectories are highly oscillatory.

• *The swarm topology*. The information links between particles, that is, connections between particles for the interchange of the information with the global best is called swarm topology. The first versions of Particle Swarm Optimization used a completely informed network where the global best was the particle that had the highest fitness from the entire swarm (or the lowest misfit in the case of a minimization). Kennedy[26,30] studied different particle configurations: *circles*, *wheels*, *stars*, *random edges*, etc. In these studies he concluded that using moderately connected networks performs better than completely informed networks. The number of informed particles can vary. For instance, in the present Standard PSO each particle informs only K particles, usually three.

Although all these numerical experiments are interesting, our opinion is that the most important point is to perform a correct stochastic stability analysis of the PSO trajectories.

REVIEW

2.6. The PSO Parameter Tuning

All the numerical experiments and theoretical analysis on PSO have shown that its success depended greatly on the fitness function but also on the parameters (ω, a_g, a_l) that have been selected. The first tentatively-given values for a_g and a_l were 2 for both,[28] being ω fixed to 1. This point is located on the right border of the first order stability region. Thus, the particle trajectories do not collapse towards the oscillation center (2) and (almost) perform a random search strategy. Nevertheless for low dimensional problems, PSO can find the global optimum due to its stochastic character. As an American proverb says: "even a blind pig can find an acorn every once in a while." In Figure 4 the position of the local best in the last iteration can be seen when $\omega = 1$ and for a $(\omega, \overline{\phi})$ point inside the first order stability region. It can be observed that the dispersion of the local best positions is higher in the first case, showing more exploratory character of the $\omega = 1$ version. Also the number of particles located on the boundaries of the search space is higher in this case due to the non stable dynamics of the swarm.

Later on when the inertia weight was introduced,[43] Shi and Eberhart proposed to adopt $0.8 < \omega < 1.2$ for the same values of a_g and a_l used by Kennedy and Eberhart[28] ($a_g = a_l = 2$). Similar comments as in the previous case about the stability of the mean trajectories can be made, particularly for $1 \le \omega < 1.2$. Nevertheless, the question of which were the best (ω, a_g, a_l) values was still left opened.

The introduction of the PSO with constriction factor[7] popularized the following parameter sets:[11]

• Adopting

$$\varphi = 4.1 \Longrightarrow \varphi_1 = \varphi_2 = 2.05$$

then

$$\begin{cases} \omega = \chi(\varphi) = 0.729 \\ a_g = a_l = \chi\varphi_1 = 1.494 \end{cases}$$

• Carlisle and Dozier[6] used a variant of the previous case with $\varphi_1 = 1.3$ and $\varphi_2 = 2.8$. This corresponds with $\omega = 0.729$ and $a_g = 0.9488$ and $a_l = 2.0436$.

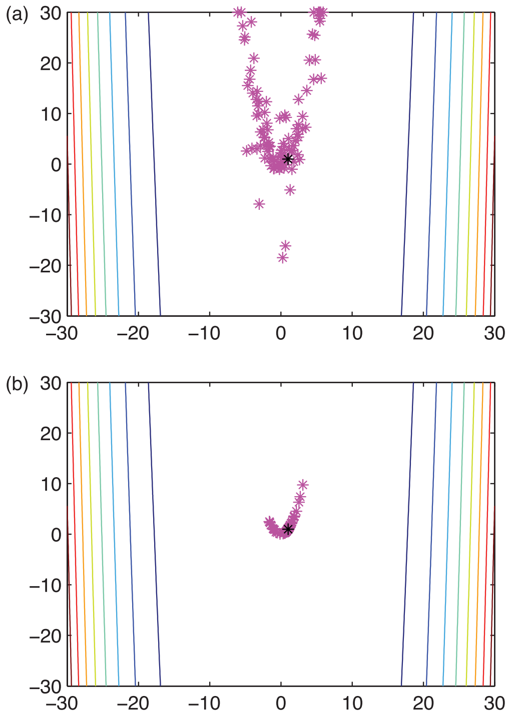

Fig. 4. Positions of the local best particles in the last iteration for a PSO run using the Rosenbrock function in 2 dimensions, with 100 particles, $(\omega, a_g, a_l) = (0.6, 1.7, 1.7)$, search space $[-30, 30]^2$ and 50 iterations. (a) $\omega = 1$ and $\overline{\phi} = 2$; (b) $\omega = 0.6$ and $\overline{\phi} = 1.7$. The black asterisk denotes the global optimum.

• Using again the constriction factor coefficient, Clerc proposed a new set of values:

$$\chi = \frac{1}{2\ln 2}$$

then

$$\varphi = 4.107 \Longrightarrow \varphi_1 = \varphi_2 = 2.0535$$

and

$$\begin{cases} \omega = \chi(\varphi) = 0.72 \\ a_g = a_l = \chi\varphi_1 = 1.48 \end{cases}$$

This parameter set is used in the actual Standard PSO version.

Based on a large number of simulation experiments Trelea[48] selected the point $\omega = 0.6$ and $a_g = a_l = 1.7$.

It has been shown numerically, for a wide range of benchmark functions, that there is not only one magic point but a whole $(\omega, \overline{\phi})$ region where PSO performs equally well.[13] This region usually includes the previous points. An example can be seen in Figure 5(c), being $\overline{\phi} = (a_g + a_l)/2$. As we will show later on, this region is close to the upper border of the second order stability region where the variance of the trajectories is very high or becomes unbounded (unstable). Thus, the region where these points lie is characterized by very high exploratory capabilities, that is, exploration and convergence are intimately related.

3. PSO VARIANTS

3.1. PSO Taxonomy

As always happens with newborns, there have been so many PSO variants that have been proposed since its birth. Some of these variants are applied to a particular problem an had little impact. Other versions try to provide a general design for any optimization problem.

New algorithms have been designed from the hybridization of the PSO with other heuristic optimization algorithms, obviously trying to improve its convergence, if not for a general use, at least, for a particular problem. PSO has been hybridized mostly with evolutionary algorithms such as genetic algorithms[23,49] or differential evolution.[46] Nevertheless, these hybridizations are not really needed if the PSO parameters (ω, a_g, a_l) are tuned in a correct way. As we will show in the section devoted to the PSO physical models and the stability analysis, PSO can be physically interpreted as a damped mass spring system. Thus, if the PSO parameters (ω, a_g, a_l) are chosen within the first order stability region, the mean trajectories of the particles will collapse towards the oscillation center (2). Nevertheless, particle trajectories should be considered as stochastic processes. Thus, the variance of the trajectories and the temporal covariance between consecutive positions depends on the (ω, a_g, a_l) point that has been selected. This knowledge is very important to perform a correct tuning of the PSO parameters.[15,17]

Sedighizadeh and Masehian[42] enumerate and describe briefly 102 algorithms that are based on PSO. These authors classified the PSO algorithms according to the type of control variables: *continuous*, *binary* and *discrete*. They also used different criteria to cluster these algorithms, such as, *attraction* (there can be attraction, repulsion or both), *swarm topology*, *activity* (a particle can be in passive state and have no social behavior), *sign of particle trajectories* (particles can follow the worst particle instead of the best one), *hierarchy* (higher-level particles have more effect on the evolution of the swarm), *restriction* (particles move in a constricted space or not), *hybridized with other heuristic algorithms* such as Simulated Annealing, Ants Colony Optimization, Genetic Algorithms, Differential evolution, etc., *optimization objective* (single-objective or multi-objective optimization), etc.

Poli et al.[39] used other PSO classification according to the following groups: *binary particle swarms*;[29] *PSO for dynamic problems*, with optima changing with time; *PSO for noisy objective functions*, that are very common in real applications; *hybrids and adaptive particle swarms*, that try to modify the parameters of the algorithm according to the information from the objective function and use strategies taken from evolutionary algorithms; *PSO with diversity control* where the tendency of the swarm to collapse is prevented; and *bare-bones PSO*[27] where the velocity update is eliminated and particles move according to a normal probability distribution.

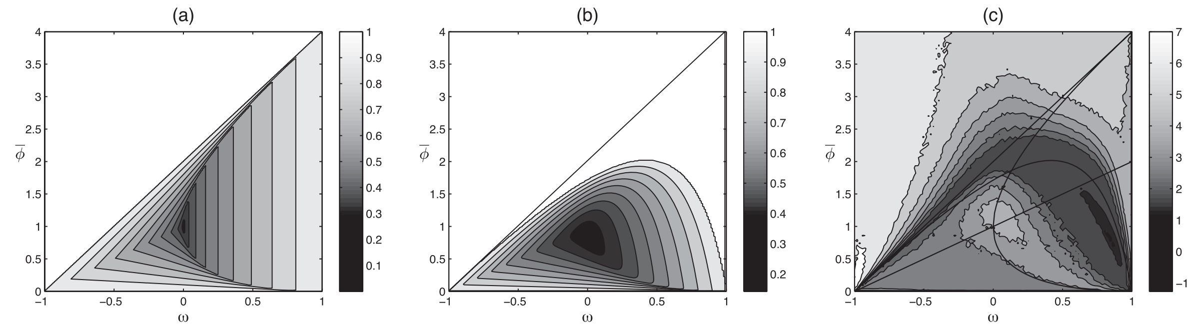

Fig. 5. (a) First order stability region, (b) second order stability region for $\alpha = 1$, (c) PSO error in logarithmic scale for Rosenbrock function for 10 dimensions, 50 runs, 100 particles and 500 iterations. Darker regions mean lower errors.

Rini et al.[41] classifies PSO algorithms in two main classes: *basic variants of PSO* and *modification variants of PSO*. In the first class they include the velocity clamping strategy, the inertia weight PSO version, the constriction factor, and the synchronous versus asynchronous (optimization in parallel processing). In the second class they took into account the kind of problem that is solved: single solution, niching (locate multiple solutions), constraint optimization, multiobjective optimization, dynamic environment and discrete problems.

Other authors[2,10] have provided their own classification. This fact provides an idea of the variety of PSO versions that have been proposed.

3.2. The Standard PSO: An Attempt for Uniformization

The creators of the PSO algorithm Russell Eberhart and James Kennedy have been involved with others authors in the elaboration of a C code to implement the so-called *Standard PSO*. This project is published in *Particle Swarm Central* (http://www.particleswarm.info) and constitutes an attempt for uniformization of all the PSO variants that have been proposed in the literature. As Clerc points out in http://clerc.maurice.free.fr/pso/, the idea was to be faithful to the original PSO principles while suppressing, as much as possible the bias introduced by the code. For example, in the different versions (2006, 2007 and 2011) of the Standard PSO:

• The 2006 version proposed to use a number of particles N that was related to the dimension D of the search space as follows $N = \mathrm{int}(10+2\sqrt{D})$. Nevertheless, it was found that this expression did not provide the optimum number of particles for different experiments. Thus, this criterion disappeared in posterior versions.

• The velocity initialization used in the 2006 and 2007 versions was changed since most of the particles left the search space. Obviously this fact is related to the stability of the trajectories.

• Using benchmark functions it was found that certain positions of the global optimum were found more easily. Spears et al.[45] studied the biases in PSO and found that particles tend to concentrate along paths parallel to the coordinate axes. Also PSO has a bias towards the center of the search space and towards the origin of the coordinate system (null vector).[34] To avoid these biases, the 2011 version updates the velocity drawing the particles from hypersphere centered in $\mathbf{G}_i^k$ points:

$$\mathbf{G}_i^k = \mathbf{x}_i^k + c\,\frac{\mathbf{g}^k + \mathbf{l}_i^k - 2\mathbf{x}_i^k}{3}, \quad i = 1, \ldots, N$$

with $c = a_g = a_l$. Obviously, the $\mathbf{G}_i^k$ points include the global and local best positions of the particles, but the 2011 version is a complete different algorithm with respect to the most popular PSO version (4).

The Standard PSO can be used as a discrete PSO, simply by projecting the continuous positions to the closest discrete acceptable ones.

3.3. Algorithm Evaluation

The performance of any PSO variant obviously depends on several factors:

• The fitness function topography of the optimization problem.[9]

• The size of the search space and its relation to the fitness function (the location of the global optimum).

• All the numerical parameters (ω, a_g, a_l, N, $N_{iter}, \ldots$) used to perform the search.

Thus, the performance of any new proposed algorithm has to be related to a set of benchmark functions. Efforts have been made in this direction and such sets of benchmark functions exist, such us those used in the Competition on Testing Evolutionary Algorithms in the successive IEEE Congress editions on Evolutionary Computation (http://www.ntu.edu.sg/home/epnsugan/), but obviously the results are only valid in this context. Typically, the success of any new algorithm is defined by reaching a very low error (or a very high fitness). Nevertheless this criterion is not valid for many real situations (inverse problems) where the cost function involves

REVIEW

the observed data, and due to the presence of noise, it is unrealistic to believe that lowering the error under certain threshold improves the solution goodness. Also, in this case, it has been proved that the solution of lowest misfit never coincides with the model that have generated the observed data.[14] In this case, it is important to search for the set of equivalent solutions that are those which fit the observed data within the same error bounds. To perform this task it is not only important to look for the solutions with low misfits but also to be able to explore these sets of solutions.

The expressions *better exploration* and *better exploitation* appear frequently in papers about PSO but, unfortunately, the research has been focused more on the last task than in the former, judging the goodness of a new algorithm by its capability of dramatically lowering the error misfit. Nevertheless, as it was already mentioned, having at disposal exploratory PSO versions it is also very important. Usually, in the PSO case, exploration and exploitation are related. Clerc has proposed a quantitative definition of both terms for the last Standard PSO that is related with the position update: exploitation occurs when the next position is inside at least one hypersphere of the old one.

4. THE STABILITY ANALYSIS AND THE PSO CONVERGENCE

4.1. The Stability Analysis of the Trajectories

Due to the high number of PSO variants that have been proposed in the literature, involving sometimes a variety of heuristic mechanisms for the positions and velocity updates, it is impossible to approach the analysis of all these algorithms as a whole. Also these heuristic modifications complicate the mathematical modeling of the algorithm. Thus this section applies only to the basic PSO (4).

The stability analysis of the particles trajectories is very important because it could be considered for well-posed optimization problems (having a unique global optimum) as a sufficient condition for the algorithm to converge, provided that the algorithm is exploratory (not premature collapse towards any other point). The first attempts to analyze the stability involved simplified PSO versions, such as one dimensional dynamics, deterministic analysis of the trajectories, stochastic analysis with stagnated oscillation center, etc.[13,36,48] As result of these analysis, the mean trajectories were classified for different $(\omega, \overline{\phi})$ points lying within the first order stability region and it was seen that the mean particle trajectories were a combination of sinusoidal and exponential functions.

To understand the stability, the PSO algorithm (4) can be rewritten only in terms of position as a second order stochastic difference equation:

$$\mathbf{x}_i^{k+1} + (\phi_1^k + \phi_2^k - \omega - 1)\mathbf{x}_i^k + \omega \mathbf{x}_i^{k-1} = \phi_1 \mathbf{g}^k + \phi_2 \mathbf{l}_i^k \quad (9)$$

Taking means, the stability condition for the first order moment of the trajectories, $E(\mathbf{x}_i^k)$, implies that all the roots the characteristic equation

$$r^2 + (\overline{\phi} - \omega - 1)r + \omega = 0 \quad (10)$$

are less than 1 in absolute value. Since some roots might be not real, this condition means that all the root lie within the unit circle in the complex plane. This stability condition supposes that the global best, $\mathbf{g}^k$, and local best, $\mathbf{l}_i^k$, attractors of each particle follow dynamics that do not alter the homogeneous stability.

As a result of this analysis, the first order stability region turned out to be the triangle represented in Figure 5(a). This region analytically corresponds to:

$$\{(\omega, \overline{\phi}) : |\omega| < 1, 0 < \overline{\phi} < 2(\omega + 1)\} \quad (11)$$

Thus, choosing $(\omega, \overline{\phi})$ inside this triangle causes the mean trajectories of the particles to be stable. This region obviously includes the stability intervals found for the particular cases shown in Sections 1.3 and 2.2.

Four different zones, depending on the kind of trajectories, have been defined:[13]

(1) The complex zone (zone 1 in Fig. 6) where the roots of (10) are both complex. The trajectories are oscillatory, decreasing and symmetrical. The frequency of oscillation depends on the $(\omega, \overline{\phi})$ point that has been selected and is always less than 1/2.
(2) The real zone with both roots negative (zone 2 in Fig. 6). The trajectories are oscillatory, decreasing and symmetrical. The frequency of the oscillation is always 1/2.
(3) The real zone with one root negative and other positive (zone 3 in Fig. 6). The trajectories are oscillatory, decreasing and asymmetrical. The frequency of the oscillation is also always 1/2.
(4) The real zone with both roots positive (zone 4 in Fig. 6). The trajectories are decreasing exponentials with iterations.

The first order stability analysis, although it shed light about the PSO behavior, did not fully explain the success of some popular (ω, a_g, a_l) parameter sets in Section 2.6. It is important to understand that the particle trajectories are stochastic processes, and the first order stability is not enough: the stability of the second order moments, the variance of the trajectories and the temporal covariance are also very important to achieve convergence.

4.2. The Second Order Stability Analysis

The second order stability analysis of the PSO trajectories has been presented in Refs. [15, 17, 38]. The second order stability region turned out to be:

$$\{(\omega, \overline{\phi}) : -1 < \omega < 1, 0 < \overline{\phi} < \phi_h(\omega, \alpha)\}$$

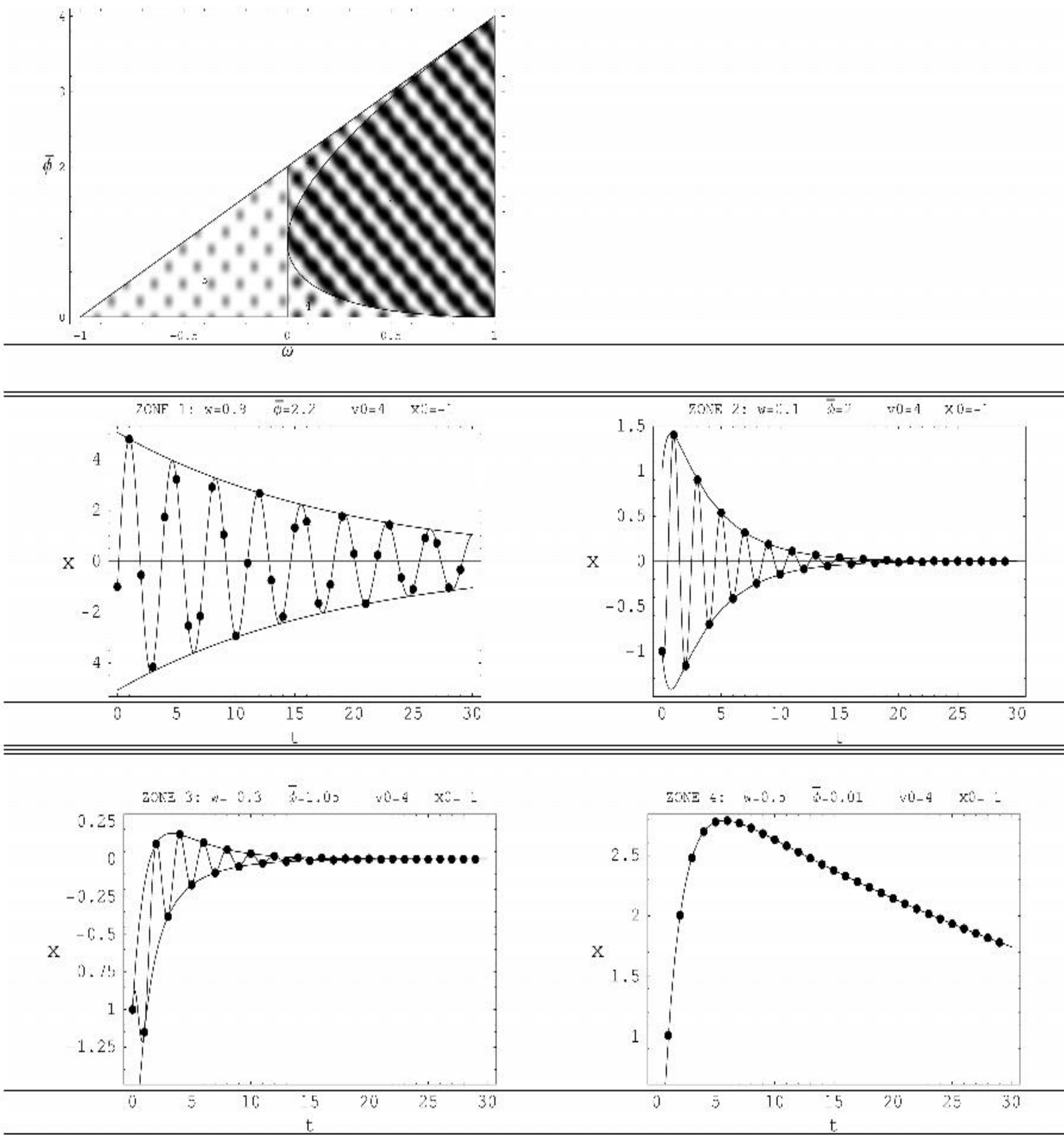

Fig. 6. The four different kinds of first order trajectories depending on the parameters $(\omega, \overline{\phi})$.

being the upper boundary the hyperbola (Fig. 5(b)):

$$\phi_h(\omega, \alpha) = \frac{12(1-\omega^2)}{4-4(\omega-1)+(\alpha^2-2\alpha)(1+\omega)}$$

$\alpha = (a_g/\overline{\phi}) = 2a_g/(a_g+a_l)$ is the ratio between the global acceleration and the total mean acceleration. This second order region is embedded within the first order stability region (11) and its size depends on the balance between the social and cognitive factors which is related to α. Unlike the first order stability region, the upper boundary $\phi_h(\omega, \alpha)$ is closely related to the best parameter sets (2.6) found in the literature.[17] Figure 5(c) serves to explain this fact.

Higher order moments are stable on regions that are embedded within the second order stability region.[15,38]

4.3. The Physical Analogies and the PSO Continuous Model

The understanding of the stability analysis of the particle trajectories and the PSO dynamics was done in parallel to the development of the physical analogies for the swarm system.

For instance, Mikki and Kishk[33] proposed a molecular dynamics formulation of the algorithm that led to a physical theory for the swarm environment.

A more fructiferous analogy consisted in relating the swarm to a damped mass spring system. In 1998 Clerc (http://clerc.maurice.free.fr/pso/) pointed out that a mechanical analogy of a particle in the swarm could be an oscillating unit mass attached to a spring. The spring analogy was also used by Brandstätter and Baumgartner[4] to adjust the PSO parameters a posteriori in order to solve an electrical engineering application. Nevertheless, these authors did not fully exploited this physical analogy. This came later when Fernández-Martínez et al.[13,15] introduced the PSO continuous model:

$$\begin{cases} x''(t)+(1-\omega)x'(t)+\phi x(t) \\ =\phi_1 g(t-t_0)+\phi_2 l(t-t_0), \quad t\in R \\ x(0)=x_0, \\ x'(0)=v_0 \end{cases} \tag{12}$$

This equation models the one dimensional movement of a unit mass damped spring system with rigidity constant ϕ and damping coefficient $1-\omega$. These authors have shown that PSO corresponds to a particular discretization, centered in acceleration $x''(t)$, and regressive in velocity $x'(t)$, of (12) adopting as time step $\Delta t=1$ and as a time delay $t_0=0$.

4.4. The GPSO and the PSO Family

The PSO continuous model also served to generalize PSO to any time step (the so-called GPSO)[15] and to construct a whole family of PSO optimizers having different stability regions and exploration/exploitation capabilities.[16,18] For instance, the GPSO algorithm updates velocities and positions as follows:

$$\begin{aligned} \mathbf{v}_i^{k+1} &= (1-(1-\omega)\Delta t)\mathbf{v}_i^k \\ &\quad +\phi_1^k\Delta t(\mathbf{g}^k-\mathbf{x}_i^k)+\phi_2^k\Delta t(\mathbf{l}_i^k-\mathbf{x}_i^k), \\ \mathbf{x}_i^{k+1} &= \mathbf{x}_i^k+\mathbf{v}_i^{k+1}\Delta t \end{aligned} \tag{13}$$

It is possible to observe that the introduction of the parameter time step Δt makes the algorithm dimensionally correct in terms of velocity, acceleration and position. The PSO (4) is a particular case of this algorithm for $\Delta t=1$. Values of $\Delta t>1$ make the algorithm more exploratory. This mechanism can be used in the first stages or when the swarm tends to collapse prematurely before achieving convergence. Also, in final stages, Δt can be reduced (less than 1) to perform a local search around the global best position. Also, Δt parameter can be varied with iterations (sand and lime algorithm[15]) and serves to increase exploration and prevent stagnation.

Using different discretizations for the acceleration and the velocity, a whole family of PSO algorithms has been constructed and their stability regions have been characterized.[16,18] The selection of the member of the PSO family depends on the exploration/exploitation balance that is desired.

5. APPLICATIONS

The successful application of PSO and its variants has been achieved in very different fields. One of the reasons could be that its simplicity. Also, PSO provides good results in most of the cases.

In 2008, Poli[37] analyzed the PSO papers that have been already published and found a wide range of applications. The main fields in alphabetical order were: antennas, biomedical, communication networks, clustering and classification, combinatorial optimization, control, design, distribution networks, electronics and electromagnetics, engines and motors, entertainment, diagnosis of faults, financial, fuzzy and neurofuzzy, graphics and visualization, image and video, metallurgy, modeling, neural networks, prediction and forecasting, power systems and plants, robotics, scheduling, security and military, sensor networks, signal processing.

Sedighizadeh and Masehian[42] pointed out that in all the fields the number of published PSO papers have grown with time. The main field was electrical engineering followed by computer science, mechanical engineering, civil engineering, chemical engineering and mathematics.

Banks[2] enumerated some applications on computer science like computer animation, face recognition,[47] E-commerce, digital image processing, computational biology and financial forecasting, etc.

Del Valle focuses in the applications of PSO to Power Systems[10] describing the particular PSO variants that are used for each problem. In most of these cases the Standard PSO is a valid option.

Rini et al.[41] classified the PSO applications, not by field, but depending on the kind of optimization problem that is solved. PSO was first applied mainly to single optimum problems, but the solution of dynamical and multiobjective problems has increased with time.

6. CONCLUSION

Particle Swarm Optimization is an algorithm that was born almost two decades ago. As a global optimization algorithm, the newborn presented some instability issues but after some improvements and theoretical analysis, it was successfully applied to a wide range of real problems. Although many PSO variants have been proposed trying to improve its convergence properties, the main breakthrough has come from the theoretical analysis, and particularly from the stochastic stability analysis of the particle swarm trajectories. The use of physical analogies has also contributed to improve the knowledge about PSO. In conclusion, PSO should be considered as a stochastic algorithm with a well established theoretical background. Thus, PSO should not be considered a heuristic algorithm anymore.

References and Notes

1. A. Banks, J. Vincent, and C. Anyakoha, A review of particle swarm optimization. part I: background and development. *Natural Computing* 6, 467 **(2007)**.
2. A. Banks, J. Vincent, and C. Anyakoha, A review of particle swarm optimization. Part II: hybridisation, combinatorial, multicriteria and constrained optimization, and indicative applications. *Natural Computing* 7, 109 **(2008)**.
3. J. Bansal, P. Singh, M. Saraswat, A. Verma, S. Jadon, and A. Abraham, Inertia weight strategies in particle swarm optimization, *2011 Third World Congress on Nature and Biologically Inspired Computing (NaBIC)*, October **(2011)**, pp. 633–640.
4. B. Brandstätter and U. Baumgartner, Particle swarm optimization—Mass-spring system analogon. *IEEE Transactions on Magnetics* 38, 997 **(2002)**.
5. H. Bremermann, The evolution of intelligence: The nervous system as a model of its environment. Technical report, Department of Mathematics, University of Washington **(1958)**.
6. A. Carlisle and G. Dozier, An off-the-shelf PSO, *Proceedings of the Particle Swarm Optimization Workshop*, Indianapolis, April **(2001)**, pp. 1–6.
7. M. Clerc and J. Kennedy, The particle swarm—Explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation* 6, 58 **(2002)**.
8. A. Colorni, M. Dorigo, and V. Maniezzo, Distributed optimization by ant colonies, *European Conference on Artificial Life* **(1991)**, pp. 134–142.
9. Z. Cui, X. Cai, and Z. Shi, Using fitness landscape to improve the performance of particle swarm optimization. *J. Comput. Theo. Nanosci.* 9, 258 **(2012)**.
10. Y. del Valle, Particle swarm optimization: Basic concepts, variants and applications in power systems. *IEEE Transactions on Evolutionary Computation* 12, 171 **(2008)**.
11. R. Eberhart and Y. Shi, Comparing inertia weights and constriction factors in particle swarm optimization, *Proceedings of the 2000 Congress on Evolutionary Computation* **(2000)**, Vol. 1, pp. 84–88.
12. R. Eberhart, P. Simpson, R. Dobbins, and R. Dobbins, Computational intelligence PC tools, Mathematical Computing, Academic Press Professional **(1996)**.
13. J. Fernández-Martínez, E. García-Gonzalo, and J. Fernández-Álvarez, Theoretical analysis of particle swarm trajectories through a mechanical analogy. *International Journal of Computational Intelligence Research* 4, 93 **(2008)**.
14. J. L. Fernández-Martínez, Z. Fernández-Muñiz, and M. J. Tompkins, On the topography of the cost functional in linear and nonlinear inverse problems. *Geophysics* 77, W1 **(2012)**.
15. J. L. Fernández-Martínez and E. García-Gonzalo, The generalized PSO: a new door to PSO evolution. *Journal of Artificial Evolution and Applications* 2008, 1 **(2008)**.
16. J. L. Fernández-Martínez and E. García-Gonzalo, The pso family: Deduction, stochastic analysis and comparison. *Swarm Intelligence* 3, 245 **(2009)**.
17. J. L. Fernández-Martínez and E. García-Gonzalo, Stochastic stability analysis of the linear continuous and discrete PSO models. *IEEE Transactions on Evolutionary Computation* 15, 405 **(2011)**.
18. E. García-Gonzalo and J. Fernádez-Martínez, Stochastic stability and numerical analysis of two novel algorithms of the pso family: PP-PSO and RR-PSO. *International Journal on Artificial Intelligence Tools* 21, 1240011 **(2012)**.
19. E. García-Gonzalo and J. L. Fernández-Martínez, Design of a simple and powerful particle swarm optimizer, *Proceedings of the International Conference of Computational and Mathematical Methods in Science and Engineering, CMMSE2009*, Gijón, Spain, June–July **(2009)**, Vol. 4, pp. 1280–1290.
20. S. Ghosh, S. Das, D. Kundu, K. Suresh, B. Panigrahi, and Z. Cui, An inertia-adaptive particle swarm system with particle mobility factor for improved global optimization. *Neural Computing & Applications* 21, 237 **(2012)**.
21. F. Heppner and U. Grenander, A stochastic nonlinear model for coordinated bird flocks, The Ubiquity of Chaos, edited by E. Krasner, AAAS Publications **(1990)**, pp. 233–238.
22. M. Jiang, Y. P. Luo, and S. Y. Yang, Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. *Information Processing Letters* 102, 8 **(2007)**.
23. C.-F. Juang, A hybrid of genetic algorithm and particle swarm optimization for recurrent network design. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 34, 997 **(2004)**.
24. V. Kadirkamanathan, K. Selvarajah, and P. Fleming, Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Transactions on Evolutionary Computation* 10, 245 **(2006)**.
25. J. Kennedy, The particle swarm: social adaptation of knowledge. *IEEE International Conference on Evolutionary Computation*, Indianapolis, IN, USA, April **(1997)**, pp. 303–308.
26. J. Kennedy, Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance. *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99* **(1999)**, Vol. 3, pp. 1931–1938.
27. J. Kennedy, Bare bones particle swarms, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium, 2003. SIS '03*, April **(2003)**, pp. 80–87.
28. J. Kennedy and R. Eberhart, Particle swarm optimization, *Proceedings IEEE International Conference on Neural Networks (ICNN '95)*, Perth, WA, Australia, November–December **(1995)**, Vol. 4, pp. 1942–1948.
29. J. Kennedy and R. Eberhart, A discrete binary version of the particle swarm algorithm, *Systems, Man, and Cybernetics, 1997. IEEE International Conference on Computational Cybernetics and Simulation*, October **(1997)**, Vol. 5, pp. 4104–4108.
30. J. Kennedy and R. Mendes, Population structure and particle swarm performance. *Proceedings of the Evolutionary Computation on 2002. CEC '02. Proceedings of the 2002 Congress—Volume 02, CEC '02*, IEEE Computer Society, Washington, DC, USA **(2002)**, pp. 1671–1676.
31. E. Laskari, K. Parsopoulos, and M. Vrahatis, Particle swarm optimization for integer programming, *Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC '02* **(2002)**, Vol. 2, pp. 1582–1587.
32. R. F. Malik, T. A. Rahman, S. Z. M. Hashim, and R. Ngah, New particle swarm optimizer with sigmoid increasing inertia weight. *International Journal of Computer Science and Security* 1, 35 **(2007)**.
33. S. M. Mikki and A. A. Kishk, Physical theory for particle swarm optimization. *Progress in Electromagnetics Research (PIER)* 75, 171 **(2007)**.
34. C. Monson and K. Seppi, Improving on the kalman swarm extracting its essential characteristics, *Proceedings of Genetic and Evolutionary Computation Conference (GECCO)*, Washington, DC, June **(2005)**.
35. M. Montes de Oca, T. Stutzle, M. Birattari, and M. Dorigo, Frankenstein's PSO: A composite particle swarm optimization algorithm. *IEEE Transactions on Evolutionary Computation* 13, 1120 **(2009)**.
36. E. Ozcan and C. Mohan, Particle swarm optimization: Surfing the waves, *Evolutionary Computation. Proceedings of the 1999 Congress on (CEC '99)*, Washington, DC, USA, July **(1999)**. Vol. 3, pp. 1939–1944.
37. R. Poli, Analysis of publications on particle swarm optimisation applications. *Journal of Artificial Evolution and Applications* 2008, 10 **(2008)**.

REVIEW

REVIEW

38. R. Poli, Dynamics and stability of the sampling distribution of particle swarm optimisers via moment analysis. *Journal of Artificial Evolution and Applications* 2008, 10 (**2008**).

39. R. Poli, J. Kennedy, and T. Blackwell, Particle swarm optimization. *Swarm Intelligence* 1, 33 (**2007**).

40. C. W. Reynolds, Flocks, herds and schools: A distributed behavioral model, *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87*, ACM, New York, NY, USA (**1987**), pp. 25–34.

41. P. Rini, S. M. Shamsuddin, and S. S. Yuhaniz, Article: Particle swarm optimization: Technique, system and challenges. *International Journal of Computer Applications* 14, 19 (**2011**). Published by Foundation of Computer Science.

42. D. Sedighizadeh and E. Masehian, Particle swarm optimization methods, taxonomy and applications. *International Journal of Computer Theory and Engineering* 1, 1793 (**2009**).

43. Y. Shi and R. Eberhart, A modified particle swarm optimizer, *IEEE International Conference on Computational Intelligence*, May (**1998**), pp. 69–73.

44. Y. Shi and R. Eberhart, Empirical study of particle swarm optimization, *Proceedings of the 1999 Congress on Evolutionary Computation, 1999. CEC 99* (**1999**), Vol. 3, pp. 1948–1950.

45. W. M. Spears, D. T. Green, and D. F. Spears, Biases in particle swarm optimization. *International Journal of Swarm Intelligence Research* 1, 34 (**2010**).

46. R. Storn and K. Price, Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization* 11, 341 (**1997**).

47. M. Sugisaka and X. Fan, An effective search method for neural network based face detection using particle swarm optimization. *IEICE—Trans. Inf. Syst.* E88-D, 214 (**2005**).

48. I. Trelea, The particle swarm optimization algorithm: Convergence analysis and parameter selection. *Information Processing Letters* 85, 317 (**2003**).

49. S. Uma, K. Gandhi, and E. Kirubakaran, A hybrid PSO with dynamic inertia weight and GA approach for discovering classification rule in data mining. *International Journal of Computer Applications* 40, 32 (**2012**).

50. D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation* 1, 67 (**1997**).

51. Y.-L. Zheng, L.-H. Ma, L.-Y. Zhang, and J.-X. Qian, Empirical study of particle swarm optimizer with an increasing inertia weight. *The 2003 Congress on Evolutionary Computation, 2003. CEC '03*, December (**2003**), Vol. 1, pp. 221–226.

52. Y.-L. Zheng, L.-H. Ma, L.-Y. Zhang, and J.-X. Qian, On the convergence analysis and parameter selection in particle swarm optimization, *International Conference on Machine Learning and Cybernetics (ICMLC'03)*, Xi'an, China, November (**2003**). Vol. 3, pp. 1802–1807.

Received: 12 May 2012. Accepted: 28 September 2012.