Particle Swarm Optimization and Inverse Problems

Esperanza García-Gonzalo and Juan Luis Fernández-Martínez

Abstract. In this paper we present a powerful set of Particle Swarm optimizers for inverse modeling. Their design is based on the interpretation of the swarm dynamics as a stochastic damped mass-spring system. All the PSO optimizers have very different exploitation and exploration capabilities. Their convergence can be related to the stability of their first and second order moments of the particle trajectories. Based on these results we present their corresponding cloud algorithms where each particle in the swarm has different inertia (damping) and acceleration (rigidity) constants. These algorithms show a very good balance between exploration and exploitation and their use avoids the tuning of the PSO parameters. These algorithms have been successfully applied to environmental geophysics and petroleum reservoir engineering where the combined use of model reduction techniques allow posterior sampling in high dimensional spaces.

Keywords: Inverse Problems, PSO, PSO Family, Cloud Design.

1 Particle Swarm Optimization Applied to Inverse Problems

Particle swarm optimization is a stochastic evolutionary computation technique used in optimization, which is inspired by the social behavior of

Esperanza García-Gonzalo and Juan Luis Fernández-Martínez
Department of Mathematics, University of Oviedo, Oviedo, Spain
e-mail: espe@uniovi.es,jlfm@uniovi.es

Juan Luis Fernández-Martínez
Energy Resources Department, Stanford University, Palo Alto, California, USA

Juan Luis Fernández-Martínez
Department of Civil and Environmental Engineering,
University of California at Berkeley, Berkeley, USA

C. Borgelt et al. (Eds.): Combining Soft Computing & Stats. Methods, AISC 77, pp. 289–296.
springerlink.com

individuals (called particles) in nature, such as bird flocking and fish schooling [15]. Inverse problems are very important in science and technology and sometimes referred to as, parameter identification, reverse modeling, etc.

Let us consider an inverse problem of the form $\mathbf{F}(\mathbf{m}) = \mathbf{d}$, where $\mathbf{m} \in \mathbf{M} \subset \mathbf{R}^n$ are the model parameters, $\mathbf{d} \in \mathbf{R}^s$ the discrete observed data, and

$$\mathbf{F}(\mathbf{m}) = (f_1(\mathbf{m}), f_2(\mathbf{m}), \ldots, f_s(\mathbf{m}))$$

is the vector field representing the forward operator and $f_j(\mathbf{m})$ is the scalar field that accounts for the j-th data. The “classical ” goal of inversion given a particular data set (often affected by noise), is to find a set of parameters $\mathbf{m}$, such the data prediction error $\|\mathbf{F}(\mathbf{m}) - \mathbf{d}\|_p$ in a certain norm p, is minimized.

The PSO algorithm to approach this inverse problem is at first glance very easy to understand and implement:

1. A prismatic space of admissible models, $\mathbf{M}$, is defined:

$$l_j \le m_{ij} \le u_j, \quad 1 \le j \le n, \quad 1 \le i \le N_{\text{size}}$$

   where l_j, u_j are the lower and upper limits for the j-th coordinate of each particle in the swarm, n is the number of parameters in the optimization problem and N_{size} is the swarm size. The misfit for each particle of the swarm is calculated, $\|\mathbf{F}(\mathbf{m}) - \mathbf{d}\|_p$ and then we determine for each particle its local best position found so far (called $\mathbf{l}_i(k)$) and the minimum of all of them which is called the global best ($\mathbf{g}(k)$).
2. The algorithm updates at each iteration the positions $\mathbf{m}_i(k)$, and velocities $\mathbf{v}_i(k)$, of each model in the swarm as follows:

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

   ω, a_g, a_l are the PSO parameters and are called inertia, local and global acceleration constants; $\phi_1 = r_1 a_g, \phi_2 = r_2 a_l$ are the stochastic global and local accelerations, and r_1, r_2 are vectors of random numbers uniformly distributed in (0, 1). In the classical PSO algorithm these parameters are the same for all the particles of the swarm. In an inverse problem the position are the coordinates of the model $\mathbf{m}$ on the search space and the velocities the perturbations needed to find the low misfit models.

1.1 Uncertainty or Why We Should Explore the Search Space

Inverse problems are a very special type of optimization problems that turn out to be ill-posed mainly due to several reasons:

1. There is a physical model $\mathbf{F}$ (forward operator) involved which is always a simplification of the reality. This includes physical hypothesis and numerical approximations of the forward operator. Typically the predic-

tion (forward problem) is well-posed, but not the inverse. The ill-posednes is somehow related to the kind of question we ask in the inverse problem.

2. The observed data are part of the cost function and typically are noisy and discrete in number (mainly due to economic and logistic reasons).
3. Finally, the forward problem usually involves the resolution of a partial differential, integral, or algebraic set of equations, and a very fine model discretization is used to achieve accurate data predictions. The number of model parameters is in most cases significantly greater than the number of discrete data points available.

Let us suppose that we have a model $\mathbf{m}_0$ that fulfills $\|\mathbf{F}(\mathbf{m}_0)-\mathbf{d}\|_2 < tol.$ It is possible to show analytically that the models in the neighborhood of $\mathbf{m}_0$ that fit the data within the same tolerance, *tol*, belong to the following hyperquadric:

$$(\mathbf{m}-\mathbf{m}_0)^T\,\mathbf{JF}_{\mathbf{m}_0}^T\mathbf{JF}_{\mathbf{m}_0}\,(\mathbf{m}-\mathbf{m}_0)+2\boldsymbol{\Delta}\mathbf{d}^T\,(\mathbf{m}-\mathbf{m}_0)+\|\boldsymbol{\Delta}\mathbf{d}\|_2^2 = tol^2$$

$\mathbf{JF}_{\mathbf{m}_0}$ is the Jacobian matrix of the operator $\mathbf{F}$ in $\mathbf{m}_0$ and $\boldsymbol{\Delta}\mathbf{d} = \mathbf{F}(\mathbf{m}_0)-\mathbf{d}$. This means that the equivalent models will have the direction of the vectors of the $\mathbf{V}$ base given by the singular value decomposition of $\mathbf{JF}_{\mathbf{m}_0}$ and whose axes are proportional to the inverse of the singular values λ_k in each direction. Due to the continuity of the Jacobian operator, we finally conclude that with no regularization term the misfit function has a flat and elongated valley shape. Also, in other kind of optimization problems (e.g., experimental fitting problems) many local minima might coexist. Thus, uncertainty in the model parameters is always important in inverse problems, forcing the modeler to explore the search space.

2 The Consistency of the PSO Family

Fernández Martínez and García Gonzalo([8], [4]), proved that the PSO algorithm can be physically interpreted as a particular discretization of a stochastic damped mass-spring system:

$$\mathbf{m}_i''(t)+(1-\omega)\mathbf{m}_i'(t)+\phi\mathbf{m}_i(t)=\phi_1\mathbf{g}(t-t_0)+\phi_2\mathbf{l}_i(t-t_0)$$

where $\phi=\phi_1+\phi_2$. This model has been addressed as the PSO continuous model since it describes (together with the initial conditions) the continuous movement of any particle coordinate in the swarm $\mathbf{m}_i(t)$, where i stands for the particle index, and $\mathbf{g}(t)$ and $\mathbf{l}_i(t)$ are its local and global attractors. In this model the trajectories $\mathbf{m}_i(t)$ are allowed to be delayed a time t_0 with respect to the attractors, $\mathbf{g}(t-t_0)$ and $\mathbf{l}_i(t-t_0)$.

Using this physical analogy we were able to analyze the PSO particle's trajectories [8] and to explain the success in achieving convergence of some popular parameters sets found in the literature [2], [3], [16]. Also we derived a whole family of PSO algorithms [5], [14] considering different difference schemes for $\mathbf{m}_i''(t)$ and $\mathbf{m}_i'(t)$:

1. GPSO or centered-regressive PSO $(t_0 = 0)$

$$\begin{aligned}&v(t+\Delta t) = (1-(1-\omega)\Delta t)\,v(t) + \phi_1 \Delta t\,(g(t) - m(t)) + \phi_2 \Delta t\,(l(t) - m(t)),\\ &m(t+\Delta t) = m(t) + v(t+\Delta t)\Delta t.\end{aligned}$$

   The GPSO algorithm is the generalization of the PSO algorithm for any time step Δt , (PSO is the particular case for $\Delta t = 1$). These expressions for the velocity and position are obtained by employing a regressive scheme in velocity and a centered scheme in acceleration.

2. CC-PSO or centered-centered PSO $(t_0 = 0)$

$$\begin{aligned}&m(t+\Delta t) = m(t) + \left[\tfrac{2+(w-1)\Delta t}{2} v(t) + \tfrac{\Delta t}{2}\phi_1(l(t)-m(t)) + \tfrac{\Delta t}{2}\phi_2(g(t)-m(t))\right]\Delta t,\\ &v(t+\Delta t) = \tfrac{2+(w-1)\Delta t}{2+(1-w)\Delta t} v(t) + \tfrac{\Delta t}{2+(1-w)\Delta t}\sum_{k=0}^{1}\begin{bmatrix}\phi_1(l(t+k\Delta t)-m(t+k\Delta t))\\ +\phi_2(g(t+k\Delta t)-m(t+k\Delta t))\end{bmatrix}.\end{aligned}$$

3. CP-PSO or centered-progressive PSO $(t_0 = \Delta t)$

$$\begin{aligned}&v(t+\Delta t) = \frac{\left((1-\phi\Delta t^2)v(t) + \phi_1\Delta t(g(t)-m(t)) + \phi_2\Delta t(l(t)-m(t))\right)}{1+(1-\omega)\Delta t},\\ &m(t+\Delta t) = m(t) + v(t)\Delta t.\end{aligned}$$

4. PP-PSO or progressive-progressive PSO $(t_0 = 0)$

$$\begin{aligned}&v(t+\Delta t) = (1-(1-\omega)\Delta t)\,v(t) + \phi_1 \Delta t\,(g(t) - m(t)) + \phi_2 \Delta t\,(l(t) - m(t)),\\ &m(t+\Delta t) = m(t) + v(t)\Delta t.\end{aligned}$$

5. RR-PSO or or regressive-regressive PSO $(t_0 = \Delta t)$

$$\begin{aligned}&v(t+\Delta t) = \frac{v(t) + \phi_1\Delta t\,(g(t)-m(t)) + \phi_2\Delta t\,(l(t)-m(t))}{1+(1-\omega)\Delta t + \phi\Delta t^2}\\ &m(t+\Delta t) = m(t) + v(t+\Delta t)\Delta t.\end{aligned}$$

The consistency of the different PSO family members has been related to the stability of their first and second order trajectories [8], [5]. The type of mean trajectories depend on the character of the eigenvalues of the first order difference equation as a function of the inertia parameter (ω) and the total mean acceleration $(\overline{\phi} = \overline{\phi}_1 + \overline{\phi}_2 = \dfrac{a_g + a_l}{2})$. Basically there are four kind of trajectories: damped oscillatory in the complex eigenvalue region, symmetrically and asymmetrically zigzagging in the regions of negative real eigenvalues and almost monotonous decreasing character in the region of positive real eigenvalues. Maximum exploration is reached in the complex region. The second order trajectories [5] show a similar kind of behavior. The second order spectral radius controls the rate of attenuation of the second order moments of the particle trajectories (variance and temporal covariance between $m(t)$ and

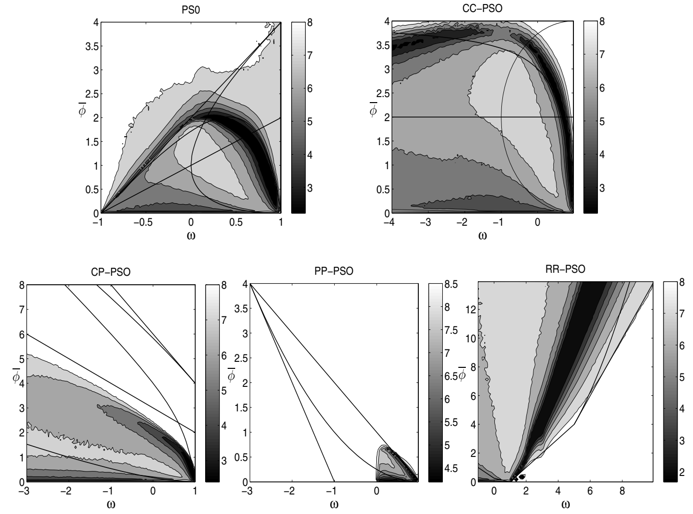

Fig. 1 Logarithmic median misfit errors for the Rosenbrock function in 50 simulations (after 300 iterations) for different family members. Similar results can be achieved for other benchmark functions.

$m(t+\Delta t)$). These results have been confirmed by numerical experiments with different benchmark functions in several dimensions. Figure 1 shows for each family member the contour plots of the misfit error (in logarithmic scale) after a certain number of iterations (300) for the Rosenbrock function. This numerical analysis is done for a lattice of $\left(\omega, \overline{\phi}\right)$ points located in the corresponding first order stability regions over 50 different simulations. For GPSO, CC-PSO and CP-PSO better parameter sets, (ω, a_g, a_l), are located on the first order complex region, close to the upper border of the second order stability region where the attraction from the particle oscillation center is lost, i.e. the variance becomes unbounded; and around the intersection to the median lines of the first stability regions where the temporal covariance between trajectories is close to zero [5]. The PP-PSO does not converge for $\omega < 0$, and the good parameter sets are in the complex region close to the limit of second order stability and $\overline{\phi} = 0$. The good parameters sets for the RR-PSO are concentrated around the line of equation $\overline{\phi} = 3(\omega - 3/2)$, mainly for inertia values greater than two. This line is located in a zone of medium attenuation and high frequency of trajectories. The CP-PSO and RR-PSO are the versions that have the greatest exploratory capabilities. Finally we performed the full stochastic analysis of the PSO continuous and discrete models [6], [7]. This analysis served to analyze the GPSO second order trajectories, to show the convergence of the discrete versions (GPSO) to the continuous PSO model as the discretization time step goes to zero, and to explain the role of the cost function on the first and second order continuous and discrete dynamical systems. Thus, PSO should not be considered heuristic.

3 How to Achieve Exploration: The Cloud Algorithms

Based on the consistency results shown above we have designed a PSO algorithm where each particle in the swarm has different inertia (damping) and local and global acceleration (rigidity) constants, being the $\left(\omega,\overline{\phi}\right)$ sets located in the low misfit regions. This idea has been implemented for the particle-cloud PSO algorithm in [13] and extended to CC-PSO and CP-PSO in [10]. Here we present the results for PP-PSO and RR-PSO. We also present the coordinates-cloud algorithm where all the same index coordinates of all the particles in the swarm will have the same (ω, a_g, a_l) constants.

The particle-cloud algorithm works as follows:

1. The misfit contours to design the clouds are based on the Rosenbrock function in 50 dimensions. The Rosenbrock function has been chosen for this purpose because in inverse problems the low misfit models are located along flat elongated valleys. Nevertheless the cloud could be designed using other benchmark functions. For each $\left(\omega,\overline{\phi}\right)$ located on the low misfit region, we generate three different (ω, a_g, a_l) points corresponding to $a_g = a_l$, $a_g = 2a_l$ and $a_l = 2a_g$. Particles are randomly selected depending on the iterations. The algorithm keep track of the (ω, a_g, a_l) points used to achieve the global best solution in each iteration. The criteria used to select the points belonging to the cloud is not very rigid, since points located on the low misfit region provide very good results.
2. The algorithm also uses the lime and sand modality, that is, varying Δt with iterations [4]. The first and second order stability regions increase their size when Δt goes to zero. In this case the exploration is increased around the global best solution. Conversely when Δt is greater than one the exploration is increased in the whole search space.

Table 1 shows the results obtained for different benchmark functions in 50 dimensions, using the particle cloud algorithm. The misfits are compared in to the reference values published in the literature. The RR-PSO, CC-PSO and PSO are the most performing algorithms.

The coordinate cloud algorithm gives also very good results but it is a more explorative version than the particle-cloud. Nevertheless, as pointed before, in inverse modeling it is not only important to achieve very low misfits but also to explore the space of possible solutions. When these algorithms have to be used in explorative form the cloud versions become a very interesting approach, because there is no need to tune the PSO parameters. Finally, the exploration can be also increased by introducing repulsive forces in the swarm by switching to negative the sign of the acceleration constants. This strategy has been used in [9], [11] to solve geophysical environmental inverse problems.

Table 1 Comparison between the particle-cloud modalities and the reference misfit values for Standard PSO [1] for different benchmark functions in 50 dimensions.

Median	Griewank	Rastrigin	Rosenbrock	Sphere
Standard PSO	9.8E-03	81	90	6.9E-11
PSO	9.6E-03	92	86	8.9E-19
CC-PSO	7.4E-03	99	90	1.0E-15
CP-PSO	1.8E-02	86	223	2.0E-07
PP-PSO	1.0E-01	91	251	8.4E-02
RR-PSO	1.2E-02	39	89	2.9E-25

4 Advantages and Drawbacks of Particle Swarm Optimization

Particle Swarm Optimization is a global stochastic search algorithm and it is typically used to solve optimization problems when the number of parameters is small (hundreds) and the forward problem is fast to compute. The advantage of these methods is that they address the optimization problem as a sampling problem. Thus, they do not do any regularization. In inverse problems, both a large number of parameters, and very costly forward evaluations hamper the use of global algorithms. The combined used of PSO and model reduction techniques allow us to address real world applications having thousands of parameters [12]. The use of model reduction techniques is based on the fact that the inverse model parameters are not independent. Conversely, there exist correlations between model parameters introduced by the physics of the forward problem in order to fit the observed data. Taking advantage of this fact it is possible to reduce the number of parameters that are used to solve the identification problem. Also the use of model reduction techniques helps to regularize the inverse problem, allowing to perform model appraisal by sampling the family of equivalent models that fit the observed data and are in accord with the prior information that it is at disposal. Nevertheless, their use should be evaluated from the perspective of each particular application, that is, model reduction techniques should be used with care and not just to accelerate computation.

References

1. Birge, B.: PSOt - a particle swarm optimization toolbox for use with Matlab. In: Proceedings of the 2003 IEEE Swarm Intelligence Symposium, SIS 2003, Indianapolis, Indiana, USA, pp. 182–186 (2003)
2. Carlisle, A., Dozier, G.: An off-the-shelf PSO. In: Proceedings of the Particle Swarm Optimization Workshop, Indianapolis, Indiana, USA, pp. 1–6 (2001)

3. Clerc, M., Kennedy, J.: The particle swarm - explosion, stability, and convergence in a multidimensional complex space. IEEE Trans. Evol. Comput. 6, 58–73 (2002)
4. Fernández Martínez, J.L., García-Gonzalo, E.: The generalized PSO: a new door to PSO evolution. J. Artificial Evol. Appl. ID 861275 (2008), doi:10.1155/2008/861275
5. Fernández-Martínez, J.L., García-Gonzalo, E.: The PSO family: deduction, stochastic analysis and comparison. Swarm Intell. 3, 245–273 (2009)
6. Fernández-Martínez, J.L., García-Gonzalo, E.: Stochastic stability analysis of the linear continuous and discrete PSO models. Technical Report, Department of Mathematics, University of Oviedo, Spain (2009)
7. Fernández-Martínez, J.L., García-Gonzalo, E.: What makes Particle Swarm Optimization a very interesting and powerful algorithm? In: Handbook of Swarm Intelligence – Concepts, Principles and Applications Series on Adaptation, Learning, and Optimization. Springer, Heidelberg (to appear, 2010)
8. Fernández-Martínez, J.L., García-Gonzalo, E., Fernández-Alvarez, J.P.: Theoretical analysis of particle swarm trajectories through a mechanical analogy. Int. J. Comput. Intell. Res. 4, 93–104 (2008)
9. Fernández-Martínez, J.L., García-Gonzalo, E., Fernández-Álvarez, J.P., Kuzma, H.A., Menéndez-Pérez, C.O.: PSO: A Powerful Algorithm to Solve Geophysical Inverse Problems. Application to a 1D-DC Resistivity Case. J. Appl. Geophys. (accepted for publication, 2010)
10. Fernández-Martínez, J.L., García-Gonzalo, E., Fernández-Muñiz, Z., Mukerji, T.: How to design a powerful family of Particle Swarm Optimizers for inverse modeling. New Trends on Bio-inspired Computation. Trans. Inst. Meas. Contr. (accepted for publication, 2010)
11. Fernández-Martínez, J.L., García-Gonzalo, E., Naudet, V.: Particle Swarm Optimization applied to the solving and appraisal of the Streaming Potential inverse problem. Geophys, Hydrogeophysics Special Issue (accepted for publication, 2010)
12. Fernández-Martínez, J.L., Mukerji, T., García-Gonzalo, E.: Particle Swarm Optimization in high dimensional spaces. In: Proceedings of the Seventh International Conference on Swarm Intelligence, ANTS 2010, Bruxelles, Belgium (2010)
13. García-Gonzalo, E., Fernández-Martínez, J.L.: Design of a simple and powerful Particle Swarm optimizer. In: Proceedings of the International Conference on Computational and Mathematical Methods in Science and Engineering, CMMSE 2009, Gijón, Spain (2009)
14. García-Gonzalo, E., Fernández-Martínez, J.L.: The PP-GPSO and RR-GPSO. Technical Report. Department of Mathematics. University of Oviedo, Spain (submitted for publication, 2010)
15. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of the IEEE International Conference on Neural Networks, ICNN 1995, Perth, WA, Australia, pp. 1942–1948 (1995)
16. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. Inf. Processing Lett. 85, 317–325 (2003)