Chapter 1
Beyond Standard Particle Swarm Optimisation

Maurice Clerc
Independent Consultant, France

ABSTRACT
Currently, two very similar versions of PSO are available that could be called “standard”. While it is easy to merge them, their common drawbacks still remain. Therefore, in this paper, the author goes beyond simple merging by suggesting simple yet robust changes and solving a few well-known, common problems, while retaining the classical structure. The results can be proposed to the “swarmer community” as a new standard.

TWO FOR ONE
Standard Descriptions
A Standard PSO is now freely available on the Particle Swarm Central (PSC, 2010) since a few years. The current version is called Standard PSO 2007 (SPSO-2007 in short). The list of contributors is quite long as it also includes the “negative” contributions, i.e. work from people who have tested some possible variants, and found them to be not suitable for inclusion in a “standard” that should be both simple and robust.

The velocity update equation for one particle in SPSO-2007 is given by

\[ v(t + 1) = w v(t) + c_1 (p(t) - x(t)) + c_2 (g(t) - x(t)) \]

(1)
Beyond Standard Particle Swarm Optimisation

The notations in this equation are as usually found in papers about PSO, that is, \( t \) is the time step, \( v \) is the velocity vector (in fact the displacement vector), and \( x \) is the position. The positions and are respectively the best previous position of the particle, and the best previous position known by its neighbours (or informants) at time \( t \). Also, this formula has to be applied independently for each dimension. \( \tilde{c}_1 \) (resp. \( \tilde{c}_2 \)) is a random number drawn from the uniform distribution on \([0,c_1]\) (resp. \([0,c_2]\)). The inertia weight is constant. The values of these three coefficients are derived analytically (Clerc, 2006b), and are

\[
\begin{align*}
\omega &= \frac{1}{2 \ln (2)} \approx 0.72 \\
c_1 &= \frac{1}{2} + \ln (2) \approx 1.2 \\
c_2 &= c_1
\end{align*}
\]  

These are slightly different from the ones used in SPSO 2006, which were derived using an older analysis (Clerc & Kennedy, 2002). The new position is given by

\[
x(t + 1) = x(t) + v(t + 1)
\]  

A typical example of a variant that is commonly used but is not included in this standard is the reduction of the weight \( w \) from \( w_{\text{max}} \) to \( w_{\text{min}} \) as a function (usually linear) of the time \( t \) (Shi & Eberhart, 1998). It does not mean that this method is bad. In fact, it is good for some problems, but is highly dependent on four parameters: \( w_{\text{max}} \), \( w_{\text{min}} \), the swarm size \( S \), and the pre-defined maximum number of fitness evaluations \( FE_{\text{max}} \). As the standard should be both simple and robust, and as this variant has too much dependence on too many parameters; it has not been retained in the standard.

Of course, SPSO-2007 also uses a swarm size, but it is not a parameter to be tuned. There, it is automatically computed by the formula

\[
S = 10 + 2\sqrt{D}
\]  

The initialisation of the positions is done at random (uniform distribution) inside the search space. The initialisation of the velocities is also done at random, by the “half-diff” method (see (Helwig & Wanka, 2008) for an analysis of various techniques). For each particle, it can be formulated as

\[
v(0) = 0.5 \left( \text{randomPosition} - x(0) \right)
\]  

To control excessive movements of the particles, SPSO 2007 uses a confinement method: when a particle tends to leave the search space, the component of the position that is too big (resp. too small) is set to the maximum admissible value (resp. to the minimum acceptable value) and the corresponding velocity component is set to zero.

Finally, to complete the description of SPSO-2007, we have to define the communication network between the particles (the topology). It depends on a parameter \( K \), set by default to 3. At the very beginning (initialisation), and after each unsuccessful iteration (i.e. if the current solution does not improve), each particle builds \( K \) information links at random, by using an uniform distribution over the whole swarm. Also, each particle informs itself. As a result, the number \( Y \) of informants of a given particle is given by a probability distribution

\[
\text{prob}(Y = n) = \frac{(S - 1)(K - 1)^{n-1}}{n(S - 1)} \left( 1 - \frac{K}{S} \right)^{S-n}
\]  

Related Content

A Novel Extremal Optimization Approach for the Template Design Problem
[www.igi-global.com/article/novel-extremal-optimization-approach-template/54068?camid=4v1a](www.igi-global.com/article/novel-extremal-optimization-approach-template/54068?camid=4v1a)

Unit Commitment by Evolving Ant Colony Optimization
[www.igi-global.com/article/unit-commitment-evolving-ant-colony/46138?camid=4v1a](www.igi-global.com/article/unit-commitment-evolving-ant-colony/46138?camid=4v1a)

Distributed Intelligence for Constructing Economic Models
Ting Yu (2012). *Intelligent and Knowledge-Based Computing for Business and Organizational Advancements* (pp. 206-219).
[www.igi-global.com/chapter/distributed-intelligence-constructing-economic-models/65795?camid=4v1a](www.igi-global.com/chapter/distributed-intelligence-constructing-economic-models/65795?camid=4v1a)

CSMA: Context-Based, Service-Oriented Modeling and Analysis Method for Modern Enterprise Applications
[www.igi-global.com/chapter/csma-context-based-service-oriented/65789?camid=4v1a](www.igi-global.com/chapter/csma-context-based-service-oriented/65789?camid=4v1a)