Particle Swarm Optimization: A Survey

Adham Atyabi  
Faculty of Computer Science,  
Flinders University  
aatyabi@acm.org

Sepide Samadzadegan  
Linköping University,  
Sweden Institute of Technology  
sepideh_samadzadegan@hotmail.com

Abstract

Swarm Intelligence (SI) describes the evolving collective intelligence of population/groups of autonomous agents with a low level of intelligence. Particle Swarm Optimization (PSO) is an evolutionary algorithm inspired by animal social behaviour. PSO achieves performance by iteratively directing its particles toward the optimum using its social and cognitive components. Various modifications have been applied to PSO focusing on addressing a variety of methods for adjusting PSO’s parameters (i.e., parameter adjustment), social interaction of the particles (i.e., neighbourhood topology) and ways to address the search objectives (i.e., sub-swarm topology). The PSO approach can easily fit in different search optimization categories such as Self Learning, Unsupervised Learning, Stochastic Search, Population-based, and Behaviour-based Search. This study addresses these principal aspects of PSO. In addition, conventional and Basic PSO are introduced and their shortcomings are discussed. Later on, various suggestions and modifications proposed by literature are introduced and discussed.

1. Introduction

Swarm Intelligence (SI) addresses the evolving collective intelligence of population/group of autonomous agents with a low level of intelligence [20]. The population of agents interacts with each other or their environment locally using decentralized and self-organizational aspects in their decision making. SI and related sub-methods that follow its principles are used for problem solving in variety of areas, such as robotics (Robotic Swarm/Swarm Robotics) and forecasting (Swarm Prediction). The principles of SI can be seen in nature in ant colonies, bird flocking, fish schooling, bacterial growth, and animal herding [14, 34, 35]. A variety of search optimization methods based on SI have been introduced, such as: Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Stochastic Diffusion Search (SDS), and Gravitational Search (GS). Above all, PSO has proved its potential in various problems. In addition to introducing the history and providing an overview of PSO, this study also addresses PSO’s weaknesses and their proposed solutions.
2. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is an evolutionary algorithm, introduced by Kennedy and Eberhart in 1995, that is inspired by the social and cognitive interactions of animals with one another and with the environment (e.g., fish schooling and birds flying). PSO achieves performance by iteratively directing its particles toward the optimum using social and cognitive components. Locations of particles, denoted by \( x_{i,j} \), are influenced by their velocity component in the n-dimensional search space, denoted by \( V_{i,j} \), where \( i \) represents the particle’s index and \( j \) is the dimension in the search space. In PSO, particles are considered to be possible solutions; they fly through the virtual space with respect to maximum velocity limitations, denoted by \( V_{\text{max}} \). Particles are generally attracted to the positions that yield the best results [15, 30, 33]. The best positions, for example \( \text{local}_\text{best} (p_i) \) and \( \text{global}_\text{best} (g_i) \), are stored in each particle’s memory. In general, the \( \text{local}_\text{best} \) of each particle can be seen as the position in which the particle achieved its highest performance; whereas the \( \text{global}_\text{best} \) of each particle can be seen as the best \( \text{local}_\text{best} \) position achieved by neighbouring particles. Various neighbourhood topologies and their impact on overall performance of the swarm are discussed in the following sections.

2.1 Conventional PSO (Original PSO)

In conventional PSO, as introduced by Kennedy and Eberhart, two elastic forces are considered in a way to represent: (1) an attraction to the fittest location by each particle; and (2) the fittest location observed by any member of the swarm. These attraction forces are governed by separate random magnitudes [38]. In general, the following definitions are used in PSO:

- **Particle**—an individual or a possible solution. It is common to define a particle as a sequence of bits.
- **Swarm**—the population of particles or solutions.
- **Dimension**—the total number of bits used in construction of a particle represents the dimension. The dimensionality is common for all particles of the swarm.

The original PSO equation is as follows:

\[
\alpha_i = \psi_1 R_1(x_{g_i} - x_i) + \psi_2 R_2(x_{p_i} - x_i)
\]

\[
v_i(t) = v_i(t - 1) + \alpha_i
\]
where \( x_s \) and \( x_p \) are global-best and personal-best and \( x \) is the particle's current location. \( R_1 \) and \( R_2 \) are random variables and \( \psi_1 \) and \( \psi_2 \) are coefficients that control the impact of cognitive and social interactions in the swarm.

### 2.2 Basic PSO

In Basic PSO, particles update their positions in the search space in consideration of their previous location in each of the previous iterations and their current velocity. Therefore, the choice of velocity function has a great impact on the overall behaviour of the swarm. In Basic PSO, velocity is influenced by three components, last velocity (corresponding to previous choice), cognitive, and social, referred to by the abbreviations \( V \), \( C \), and \( S \), respectively. The velocity equation of Basic PSO is shown in Eq. 2. In this equation, the local-best position can be extracted from the cognitive component, shown as the second equation, and the global-best position is achievable via the social component, presented as the third equation. These equations are written with respect to neighbourhood topology [42].

\[
\begin{align*}
V_{i,j}(t) &= wV_{i,j}(t-1) + C_{i,j} + S_{i,j} \\
C_{i,j} &= c_1r_1 \times (p_{i,j}(t-1) - x_{i,j}(t-1)) \\
S_{i,j} &= c_2r_2 \times (g_{i,j}(t-1) - x_{i,j}(t-1))
\end{align*}
\]  

(2)

Subsequently, Eq. 3 can be used to compute the new position (solution) in the search space:

\[
x_{i,j}(t) = x_{i,j}(t-1) + V_{i,j}(t)
\]

(3)

### 2.3 Parameter Adjustment

The neighbourhood parameters introduced Eq. 2 (\( c_1 \), \( c_2 \), \( w \), \( r_1 \), and \( r_2 \)) control the impact of each component on the overall performance of the swarm. In consideration of the role of these parameters, the choice of proper parameter adjustment criterion is most essential to the success of Basic PSO. Some possibilities are discussed as follows.
2.3.1 Acceleration coefficients

In Eq. 2, \( r_1 \) and \( r_2 \) are random values in the range of 0 and 1 based upon the uniform distribution. \( c_1 \) and \( c_2 \) are acceleration coefficients which control the impact of cognitive and social components in the process of finding solution. Various suggestions have been made based on using different adjustments for these acceleration coefficients such as using Fix/Constant (FAC), Random (RandAC), or Linearly Decreasing (LDAC) values.

**Linearly Decreasing Acceleration Coefficient (LDAC)** criterion is demonstrated in Eq.4 [11, 22, 32, 38]:

\[
\begin{align*}
  c_1 &= \left[ c_{1f} - c_{1i} \right] \frac{\text{iter}}{\text{maxiter}} + c_{1i} \\
  c_2 &= \left[ c_{2f} - c_{2i} \right] \frac{\text{iter}}{\text{maxiter}} + c_{2i}
\end{align*}
\]

In Eq. 4, \( c_{1f} \) and \( c_{1i} \) are final and initial values of \( c_1 \) respectively. Similarly, \( c_{2f} \) and \( c_{2i} \) are final and initial values of \( c_2 \) respectively. Literature claimed the best achievement using \((c_{1f}=0.5 \text{ and } c_{1i}=2.5)\) and \((c_{2f}=2.5 \text{ and } c_{2i}=0.5)\) [11, 22, 32].

The idea behind using LDAC is to increase the global search (exploration) in the beginning of the optimization and to converge particles toward global optima in later stages (exploitation).

2.3.2 Maximum and Minimum Velocity \((V_{\text{max}} \text{ and } V_{\text{min}})\)

Due to the fact that particles often choose positions outside of the search space or search boundary, or remain in the same location for a number of iterations using high or low valued velocities, it is common to limit the velocity using a maximum and minimum allowable value denoted as \( V_{\text{max}} \) and \( V_{\text{min}} \) as follows [30]:

\[
V_{i,j} = \begin{cases} 
V_{\text{min}} & \text{if } V_{i,j} \leq V_{\text{min}} \\
V_{\text{max}} & \text{if } V_{i,j} \geq V_{\text{max}} \\
V_{i,j} & \text{Otherwise}
\end{cases}
\]

It is common to define \( V_{\text{max}} \) as one half of the total search range.
The other possibility to control the velocity and guarantee particles’ maintenance in the search space is to use the Constriction Factor \((K)\) proposed by Clerc [12]. The method is reported to result in better quality solutions compared to the original PSO.

In PSO, particles show rapid and slow convergence in the first and end periods of the search respectively. In Clerc’s method, the constriction factor targeted the particles that are stagnated near the global optimum due to the lack of diversity. The principle concept behind this method was to use an extra velocity for some of the particles, similar to the idea of mutation in a Genetic Algorithm (GA). Since it is assumed that these particles have been converged somewhere near the global optimum, the mutation makes it possible for them to reach to the desired solution and subsequently, to invite other particles to imitate their behaviour (using personal global best) and converge toward the global optimum. Such a mechanism is mimicked by the use of a mutation operator with the probability of \(P_{\text{mate}}\). \(P_{\text{mate}}\) is equal to \(1/d\) where \(d\) is the number of components in the vector. Clerc proposed the following velocity equation [12]:

\[
v_{i,j} = K \left[ v_{i,j} + c_1 \frac{r_{1,i}}{r_i} (p_{i,j} - x_{i,j}) + c_2 \frac{r_{2,i}}{r_i} (g_i - x_{i,j}) \right]
\]

where \(r_i = r_{1,i} + r_{2,i}\) and \(K = \frac{2}{\phi^2 - \phi^2 + \phi}\) where \(\phi = c_1 + c_2\)

2.3.3 Inertia Weight \((w)\)

Inertia weight is considered to control the impact of previous decisions in future acts of particles. In general, various strategies are commonly practiced, such as: i) using fixed/predefined values\(^1\), ii) using random values and iii) using linearly decreasing values.

**Linearly Decreasing Inertia Weight (LDIW) criterion** is demonstrated in Eq.7 [38]:

\[
w = (w_1 - w_2) \times \frac{\text{max iter} - t}{\text{max iter}} + w_2
\]

Exploring and exploiting the search space is achievable via the use of large or small values as inertia weight. In general, the use of high values for inertia weight results in exploration and

\(^1\)The most common value for \(w\) is 0.729844 [5, 37].
consequently preventing particles from achieving the local minima. Exploitation is possible using low values for inertia weight. This results to convergence toward the optimal solution\(^2\).

### 2.3.4 Personal-best and Global-best

Following equations are proposed in Basic PSO for personal-best and global-best particles/solutions/positions.

\[
p_i(t) = \begin{cases} 
p_i(t-1) & \text{if } f(x_i(t)) \geq f(p_i(t-1)) \\
x_i(t) & \text{Otherwise} 
\end{cases} \tag{8}
\]

\[
g(t) = \arg \min \{ f(p_1(t)), f(p_2(t)), \ldots, f(p_N(t)) \} \tag{9}
\]

In here, \(f\) represents the fitness (evaluation) function. It is common to use benchmark functions for the evaluation purposes. However, other types of fitness functions have also been proposed especially in the robotic field [2-5].

### 2.4 Neighbourhood Topology

Watts in [38, 39] noted certain aspects with a major impact on information flow in networks, including: i) the degree of connectivity, ii) the proportion of clustering, and iii) the average of shortest distance between nodes. In Basic PSO, the neighbourhood topology has a major impact on the behaviour of the swarm, which is due to its influence on the global best position in the neighbouring particles [28]. The neighbourhood topology constrains inter-swarm communications, sometimes with adverse effects; various topologies have been proposed to influence this impact in different problems. The primary topologies are as follows:

- **Ring (Lbest or local best):** In this topology, neighbouring particles are predefined in a way that each particle is in neighbourhood of a group of subset particles (particles with closest indices) [43].
- **Star (Gbest or global best):** Refers to neighbourhood topology in which, the entire particles of the swarm share their information with each other [26].
- **Wheel:** In this topology, all nodes are connected to a central node.

\(^2\)The best experimental results were obtained by initializing the algorithm with a high inertia and linearly decreasing the value during the iterations [37].
- **Von Neumann**: In this topology, each node is connected to four other nodes around it (i.e., tope, down, left and right side).

- **Cluster**: In this topology, sub groups of neighbouring nodes are connected to each other through one of their nodes.

- **Pyramid**: Refers to a triangular networking grid.

In general, the Star and Ring topologies are the most common. This is due to Star topology capability in terms of guaranteeing optimum convergence and ring topology capability in terms of providing fast convergence toward either a local or global optimum. It is necessary to consider the slow convergence in a Star topology and the possibility of not reaching the global optimum in a Ring topology.

Various studies have investigated methods to improve neighbourhood topology. Kennedy in [16] found that swarms with higher interconnectivity perform well under unimodal problems while fewer interconnections are advised for multimodal problems. Even though it is common to assume neighbouring clusters to be homogeneous, it is also possible to consider heterogeneous neighbouring topologies in which heterogeneity is mimicked by considering neighbouring clusters with various numbers of members or nodes with various numbers of neighbouring nodes. In addition to the stated heterogeneity variations, it is also possible to assume neighbourhood topologies in which neighbouring nodes contain various characteristics. This type of heterogeneous neighbourhood topology is practiced in [2-7] with macroscopic versions of PSO—Area Extended PSO (AEPSO) and Cooperative AEPSO (CAEPSO)—which emphasize the use of: i) dynamic velocity adjustment, ii) intelligent balance between essential components of the swarm, and iii) dynamic heterogeneous neighbourhood topology. These versions have been shown to be successful in a robotic swarm based on search optimization in simulated uncertain and time-dependent environments.

### 3. PSO Vs. GA

Various researchers have focused efforts on comparing *PSO* and *GA* in different problems and domains. This is largely due to their similarities, in terms of both approaches being population-based and stochastic-based. In addition, both are categorized as unsupervised learning. In several instances, similar or in some cases better performance has been shown for *PSO* compared to *GA* in several instances. Eberhart and Kennedy claimed similar results to
GA for PSO on the Schaffer f6 function [17]. In a study done by Fourie [13], PSO outperformed GA in optimizing some standard size shape design problems [13]. Pugh, Zhang, and Martinoli, outperformed GA in a multi-robot learning scenario using a local neighbourhood version of PSO [25–27]. Kennedy and Spears [18] also claimed better performance for PSO comparing to GA in a factorial time-series experiment. Literature stated the use of fewer numbers of parameters and easier implementation as main advantages of PSO comparing to GA [29].

4. Conventional weaknesses of PSO

Some open problems for PSO have been noted, such as: i) how to control parameters, ii) how to prevent premature convergence, iii) how to define dynamism in velocity of particles, iv) what kind of fitness criteria is the most proper for each problem [1, 8, 37]. Some details about these problems are as follows:

1. **Parameter control.** Due to the fact that in PSO, particles periodically update their position based on their velocity, the choice of controlling parameters (i.e., $c_1, c_2, w, w_1, w_2$) has major impact on the overall performance of the swarm. It should be considered that the choice of controlling parameter is highly problem dependent. As an instance, the use of *Linear Decreasing Inertia Weight (LDIW)* results in exploration in early stages and exploitation and fast convergence in further periods of the search. However, *Fixed Inertia Weight (FIW)* performed better in some problems. Various methods for addressing parameters are presented in the following section.

2. **Premature convergence.** Since particles in PSO fly in a virtual search space, they sometimes tend to become trapped in local optima. The use of a poor neighbourhood topology in addition to the lack of dynamic velocity adjustment results in attraction of the entire swarm toward this local optimum, causing premature convergence. Lack of more appropriate solutions near the local optimum causes particles’ stagnation. Other possibility occurs when particles flicker and slow down near the global optimum caused by improper inertia weight choice (also called close clustering problem).

3. **Lack of dynamic velocity adjustment.** In Basic PSO, particles choose their behaviours according to a predefined, fixated mechanism. In PSO, a general combination of all three components (social, cognitive, and previous decision of the particle) affected from the chosen parameter controlling mechanism (Eq. 2) and also search diversity mechanism (which ensures the maintenance of balance between
behaviours in the swarm) is employed to provide the new velocity of each particle in each of the iterations. This simply indicates that PSO does not choose the most proper components or behaviours in each of the iterations. Therefore, there is no guarantee that the chosen solution for a particle in a specific iteration is the most proper one. This problem can also be seen in situations where particles flicker around showing stochastic and aimless movement. This mostly results in stagnation around or near an optimum [19, 37]. Due to the use of fixed velocity criterion, particles are incapable of preventing such a stagnation which ends in premature convergence.

4. **Multi-objective problems.** In multi-objective problems, Conventional PSO has a number of shortcomings. This is the nature of particles in PSO to converge primarily toward the first optimum they manage to find (either local or global). It is also the case that PSO is not robust in dynamic problems and problems affected by uncertainty.

5. **Solutions and Proposed Modifications**

In respect to the problems inherent to Basic PSO that were outlined in the previous section, various solutions and improvements have been proposed. Some such enhancements are listed as follows:

1. **Mechanisms to control velocity criterion parameters.** Some methods that have been proposed in the literature to overcome the lack of dynamic velocity adjustment and to provide better control over particles components are i) linearly decreasing inertia weight \((LDIW)\) [22], ii) time varying inertia weight \((TVIW)\) [22, 32, 37], iii) random inertia weight \((RANDIW)\) [32], iv) fix inertia weight \((FIW)\), v) time varying acceleration coefficient \((TVAC)\) [22, 32, 37], vi) random acceleration coefficients \((RANDAC)\), and vii) fix acceleration coefficients \((FAC)\) [32]. Another set of solutions were proposed focusing on neglecting one of PSO’s velocity components or introducing new components to the velocity criteria [11, 12, 19, 21, 24, 30-33].

2. **Neighbourhood topologies.** As it was mentioned, topologies such as Ring, Star, Wheel, Von Neumann, Cluster, and Pyramid appear in the literature. Among all neighbourhood topologies, the Ring and Star topologies have been shown the most interest in studies, primarily due to their ability to provide better convergence toward optimum (a Star topology guarantees convergence toward the optimum while a Ring
topology provider fast convergence toward either local or global optima) [36, 37]. However, Kennedy and Mendes suggested the possibility of achieving better performance by using both topologies at the same time (i.e. a dynamic neighbourhood topology), which takes advantage of their respective capabilities (i.e., both fast convergence and guaranteeing convergence toward the global optimum).

3. **Other strategies.** Various solutions such as Mutation, Re-Initialization, clearing personal best and global best, and the use of sub-swarms are proposed to address premature convergence, search diversity and PSOs lacking in multi-objective problems [23, 28, 40, 41]. Mutation can handle premature convergence and stagnation problems due to its ability to force some particles to leave the optimum and explore further in the search space. It is also possible to mutate/clear particle’s memory (personal best and global best) or reinitialize particles whenever they reach to an objective [37]. It is also possible to address multi-objective problems using sub-swarms that are allocated to achieve specific objectives [9, 10].

Even though numerous modifications of PSO have been proposed to address its shortcomings, the main focus of such proposals has been on defining better controls for velocity parameters, with faster convergence toward the optimum as the desired result [30]. It should be considered that the proposed parameter-adjusted methods for PSO are still domain and problem dependent, and that there is still no guarantee in terms of being able to address domain generality, due to the fact that the main concern is to maintain search balance (exploration and exploitation) in the swarm. This results in producing improper search methodology. However, other methods like AEPSO and CAEPSO have revealed the possibilities of providing proper use of essential behaviours of the swarm by using the most fit component or combination of components in each particle in each of the iterations [2-7].

The use of macroscopic modeling of PSO in AEPSO and CAEPSO furthermore reduces the amount of computation in the swarm. It should be noticed that in AEPSO and CAEPSO, in each of the iterations, each particle evaluates seven possible combination of components/behaviours and chooses the most proper one. AEPSO and CAEPSO still benefit from PSO’s advantages, such as less amount of parameters and computation, and also the simplicity of the algorithm.
6. Discussion and Conclusion

Although numerous benefits are noted for Basic PSO, it is claimed that it is insufficient in problems constrained by time-dependency, dynamism, and uncertainty. Although it is possible to provide balance between essential components and behaviours of Basic PSO, unlike in other domains, such a balance does not have a key role in these problems. In such problems, it seems to be more efficient to use dynamic priority allocation mechanism for behaviours in a way that favors swarm performance; in other words, particles evaluate their states and choose the most proper behaviour instead of being eager for search diversity.

In general, due to the stochastic-based nature of Basic PSO, it has been observed that particles frequently do not choose the most proper behaviour in some iterations. This is attributable to the particles’ lack of a proper evaluation system, which results in choosing or considering detrimental stages and states. At times, improper decision-making leads to inaccuracy in terms of decision-making and results in stochastic behaviours.

Although inaccuracy may arise with Basic PSO, in contrast, in methods like AEPSO higher possibilities can be attained in terms of using proper behaviour in each situation or iteration by enabling a dynamic velocity adjustment capability in addition to the use of macroscopic modeling of PSO. Furthermore, the use of higher degrees of cooperation between particles helps to overcome a high level of uncertainty in the search space and problem. Such cooperation supports particles to learn from each other’s mistakes and to mimic behaviours of others; learning that leads to robust performance. In addition to benefiting from the use of dynamic velocity adjustment, AEPSO also takes advantage from the use of sub-swarms that follow a dynamic neighbourhood topology. This improves the reliability of global-best positions between sub-swarms, which influences the overall performance of the swarm.

References


