

A Review of Parameters for Improving the Performance of Particle Swarm Optimization

Avneet Kaur¹ and Mandeep Kaur²

¹ Student ,Computer Science Department,
Guru Nanak Dev University,
Regional Campus, Jalandhar.

²Lecturer, Computer Science Department ,
Guru Nanak Dev University,
Regional Campus, Jalandhar, India.

avneet.kaur.arora1990@gmail.com , mandeep8@gmail.com

Abstract

Particle swarm optimization (PSO) is an artificial intelligence (AI) technique that can be used to find approximate solutions to extremely difficult or impossible numeric maximization and minimization problems. Particle swarm optimization is an optimization method. It is an optimization algorithm, which is based on swarm intelligence. Optimization problems are widely used in different fields of science and technology. Sometimes such problems can be complex due to its practical nature. Particle swarm optimization (PSO) is a stochastic algorithm used for optimization. It is a very good technique for the optimization problems. But still there is a drawback that it gets stuck in local minima. To improve the performance of PSO, the researchers have proposed some variants of PSO. Some researchers try to improve it by improving the initialization of swarm. Some of them introduced new parameters like constriction coefficient and inertia weight. Some define different methods of the inertia weight to improve performance of PSO and some of them work on the global and local best. This paper transplants some of the parameters used to enhance the performance of Particle Swarm Optimization technique.

Keywords: heuristic, optimal solution, optimization, particle, population, position, stochastic, swarm

1. Introduction

1.1. Optimization

Mathematical optimization is the selection of a best element from a set of available alternatives. Optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. Optimization includes finding "best available" values of some objective function given a defined domain or a set of constraints, which includes a variety of different types of objective functions and different types of domains. Optimization is the search for a set of variables that either maximize or minimize a scalar cost function. The decision maker wants a more efficient method than trial to obtain a quality decision [8].

1.2. Particle

In PSO, the term "particles" refers to population members which are mass-less and volume-less and are subject to velocities and accelerations towards a better mode of

behaviour [2]. A possible solution to the numeric optimization problem under investigation is represented by the position of a particle. Each particle has a current velocity, which represents the magnitude and direction toward a new and better position. A particle also has a measure of the quality of its current position, the particle's best known position and quality of the best known position of particle [7].

1.3. Particle Swarm Optimization

Particle swarm optimization (PSO) is a population based stochastic optimization technique [6] proposed by Kennedy and Eberhart in [1]. Particle swarm optimization (PSO) is an artificial intelligence (AI) technique that can be used to find approximate solutions to extremely difficult or impossible numeric maximization and minimization problems [7]. In computer science, particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. The PSO optimizes a problem by having a number of candidate solutions, called particles; and moving these particles around in the n- dimensional search-space according to some simple mathematical formulae over the particle's position and velocity[1]. Particle Swarm Optimization is a population based search algorithm [9]. Kennedy and Eberhart developed a PSO algorithm based on the behavior of individuals of a swarm. It has been noticed that members within a group seem to share the information among them. It leads to an increased efficiency of the group. The PSO algorithm searches using a group of individuals similar to the other AI-based heuristic optimization techniques [8]. The algorithm of PSO emulates from behavior of animals societies that don't have any leader in their group or swarm, such as bird flocking and fish schooling [11].

1.4. PSO as an Optimization Technique

In a physical n-dimensional search space, the PSO finds its best position by comparing different particles in a population, with each other. Comparison in PSO can be done in three ways: *pbest*, *lbest* and *gbest* .

1. The *pbest* is the best solution achieved by the particle so far
2. *lbest* is the another best value that is tracked by the particle swarm optimizer. It means the best value obtained so far by the particle in its neighborhood.
3. The *gbest* is the globally found best value in the swarm [5].

The PSO algorithm updates its velocity and position when the particle finds its best solution and improves the swarm towards better position.

1.5. PSO Algorithm

In an n-dimensional search space, position and velocity of individual are initialized. The PSO finds its best position by comparing different particles in a population, with each other. Comparison in PSO can be done in three ways:

1. *pbest*- the best solution achieved by the particle so far.
2. *lbest* - the another best value that is tracked by the particle swarm optimizer . It is the best value, obtained so far by any particle in the neighbors of the particle
3. *gbest* - is the globally found best value in the swarm [7].

The PSO algorithm updates its velocity and position when the particle finds its best solution and improves the swarm towards better position. The PSO is simple and efficient to find global solutions to hard problems [1].

2. Background of Particle Swarm Optimization (PSO)

2.1. PSO as a Member of Swarm Intelligence

Particle swarm optimization is a heuristic global optimization method [10]. Particle Swarm Optimization (PSO) was introduced to simulate the natural swarming behavior of birds as they search for food. Swarm intelligence is a kind of multi-agent system where individual agents behave according to some simple rules but interact to produce a surprisingly capable collective behavior. PSO is one form of swarm intelligence since each particle flies through the search space by updating its individual velocity at regular intervals towards:

1. *the best position or location it personally has found (i.e. the personal best),*
2. *toward the globally best position found by the entire swarm (i.e. the global best).*

The function value of each particle is iteratively evaluated in order to determine which offers the lowest function value. Since this information affects the velocity and direction of every other particle, a capable collective behavior emerges. The global best, or the neighborhood best, are stored to a memory location that all particles access and utilize to determine their individual velocities [12].

2.2. PSO Elements

Some of the elements used in the PSO algorithm includes:

- a) **Particle:** We can define the particle as p for real numbers.
- b) **Fitness Function:** Fitness Function is the function used to find the optimal solution.
- c) **Local Best:** It is the best position of the particle among its all positions visited so far.
- d) **Global Best:** The position where the best fitness is achieved among all the particles visited so far.
- e) **Velocity Update:** Velocity is a vector to determine the speed and direction of the particle.
- f) **Position Update:** Each particle in PSO updates their positions to find the global optimal position [3].

2.3. Parameter Settings for the PSO Algorithm

The parameters required for the PSO algorithm:

- a) **Initial Population:** The initial populations are generated randomly, and it is a set of particles at time t .
- a) **Swarm:** It is the disorganized population of moving particles that tend to cluster together while each particle seems to be moving in a random direction.
- b) **Population Size:** The performance of the standard algorithm is not sensitive to the population size but to the convergence rate. According to the requirement and keeping the computational requirements low, the number of particles can be fixed (say 200).
- c) **Search Space:** The range in which the algorithm computes the optimal control variables is called the search space. The algorithm will search for optimal solution in search space between 0 and 1. When any of the optimal control value of any particle exceed the searching space, the value will be reinitialized. The lower and upper boundaries are set to 0 and 1.
- d) **Maximum Generations:** This refers to the maximum number of generations allowed for the fitness value to converge with the optimal solution. The maximum generation can be set (say 20) [3].

2.4. Variants of Particle Swarm Optimization

The different variants of PSO have been introduced which can improve the performance of Particle Swarm Optimization. These include:

1. **Binary PSO (BPSO):** The BPSO is different from PSO in regard to the defined searching spaces. In PSO, moving in the space means a change in the value of position coordinates in one or more of existing dimensions. But in the BPSO moving in the spaces means a change in the probability of the fact that the value of position coordinate is zero or one.
2. **Constrained optimization via PSO (COPSO):** The COPSO algorithm is applied to constrained single-objective problems. A technique is employed to investigate the constraints. It includes an external file, called “Tolerant”, to save the particles.
3. **Evolutionary Programming and PSO (EPPSO):** This algorithm is a combination of the PSO and EP It helps the PSO capability by making a balance between local and global search to the faster convergence of the EP algorithm. The EP lacks diversity among particles with mutation between elements. This has been removed to some extent.
4. **Genetic binary PSO model (GBPSO):** This algorithm was developed to increase the dynamic conditions and discovery power in the swarm. In the BPSO, the positions and velocities are updated and some of the child particles are then added to the swarm and some others die and are separated from the swarm.
5. **Self-organization PSO (SOPSO):** In this algorithm, in addition to particle information and total swarm information, a feedback agent is used to improve the particle performance. It sets and improves its neighbour in next iteration. This agent leads in improvements in discovery and extraction of the particles. It avoids premature convergence of the total algorithm.
6. **PSO with area extension (AEP SO):** This algorithm was designed for movement of several robots in an area. It has in fact some in regard to information increasing from an extended area.
7. **Comprehensive learning PSO (CLPSO) L:** The particles are allowed to learn from different lBest positions on different dimensions. It finds multiple optimal solutions in the search space simultaneously.
8. **SCPSO:** It adopts a species conservation technique and uses the seeds of the species to determine local best positions for velocity updates. It prevents the premature convergence.
9. **Dynamic multiswarm PSO:** DMS-PSO is characterized by small swarm sizes and dynamically changing swarm topologies. More freedom is given to the neighbourhood structure of the algorithm. This algorithm performs better on complex multimodal problems.
10. **Adaptive PSO (APSO):** The algorithm adaptively replaces the current inactive particles (*i.e.* they do not have the ability of local and global searching and do not change their positions a lot, so their velocity is nearly reached to zero) with fresh particles in a way that the existing PSO-based relationships among the particles are kept.

Another PSO variant includes *PSO hybridized with auxiliary search techniques*, which enhance the performance of PSO [3, 13].

2.5. Parameters for Improving the Performance of Particle Swarm Optimization

To improve the performance of PSO, a number of PSO variants have been developed. These approaches include tuning the control parameters so as to maintain the balance

between local search and global search [6]. There are certain parameters which can help to improve the performance of PSO algorithm. These parameters include:

- a) **Initialization:** Initialization of the particles has an important role in the performance of PSO. If the initialization is not good then algorithm may search in unwanted area and it will be hard to search for the optimal solution [8].
- b) **Inertia Weight:** Inertia weight is an approach to balance the exploration-exploitation trade off. The bigger value of inertia weight encourages the exploration and smaller value of inertia weight encourages for the exploitation [9].
- c) **Mutation Operators:** To improve the performance of PSO and to escape it from the local minima, some researchers mutate the global best particle and some mutate local best particle with different techniques.
- d) **Velocity clamping:** Velocity clamping controls the global exploration of the particle. If the velocity of a particle exceeds the maximum allowed speed range, then it will set a maximum value of velocity [14].
- e) **Multi-objective optimization with PSO:** Many real world optimization problems require simultaneous optimization of a number of objectives (multi-objectives).
- f) **Dynamic Environment with PSO:** In dynamic Environments, PSO should be fast to allow quick re-optimization. It is desirable to find a good solution before the next environment change. There are several solutions for the dynamic environment. These include:
 - a) **Environment change detection:** It is to allow timeous and efficient tracking of optimum.
 - b) Response to environment changes
 - c) Changing the inertia weight update
 - d) Reinitialize Particle Solution
 - e) Limit Memory
 - f) Local Search
 - g) Split adaptive PSO
 - h) Fine-Grained
 - i) charged Swarm [4]

3. The PSO Algorithm

PSO is an iterative process. On each iteration in PSO, current velocity of each particle is first updated based on three parameters:

- i. *the particle's current velocity*
- ii. *the particle's local information*
- iii. *global swarm information.*

Then, each particle's position is updated using particle's new velocity. The two updated equations are:

$$v(t+1) = (w * v(t)) + (c_1 * r_1 * (p(t) - x(t))) + (c_2 * r_2 * (g(t) - x(t)))$$

$$x(t+1) = x(t) + v(t+1)$$

The term $v(t+1)$ means the velocity at time $t+1$. Velocity is a vector value and has multiple components such as $\{1.85, -0.43\}$, instead of being a single scalar value.

The new velocity depends on the following three terms:

First - $w * v(t)$, The w factor is called the inertia weight and is a constant like 0.73 ; $v(t)$ is the current velocity at time t .

Second- $c_1 * r_1 * (p(t) - x(t))$, The c_1 factor is a constant called cognitive weight, The r_1 factor is a random variable in the range $(0, 1)$, which is greater than or equal to 0 and strictly less than 1, the $p(t)$ vector value is the particle's best position found so far, the $x(t)$ vector value is the particle's current position.

Third- $c_2 * r_2 * (g(t) - x(t))$, the c_2 factor is a constant called the global weight, the r_2 factor is a random variable in range (0, 1), the $g(t)$ vector value is the best known position found by any particle in the swarm so far.

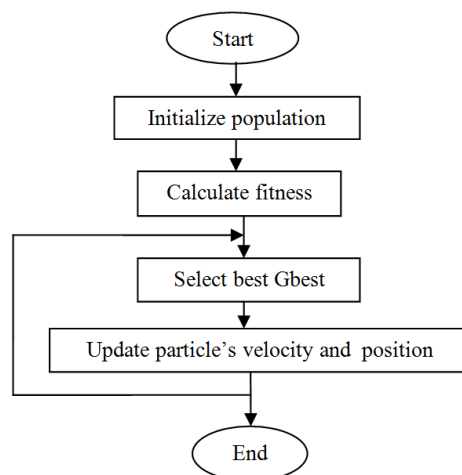
Once the new velocity, $v(t+1)$, has been determined, it's used to compute the new particle position $x(t+1)$.

3.1. PSO Pseudo Code

```
The pseudo code of original PSO is:  
Initialize the population randomly  
While (Population Size)  
{  
  Loop  
  Calculate the fitness  
  If the fitness value is better from the best fitness value (pbest) found so far then,  
  Update the pbest with new pbest  
End loop  
Select particle with the best fitness value from all particles as the gbest  
While maximum iterations or the minimum error criteria is not attained  
{  
  For each particle  
  Calculate the particle velocity  
  Update particle position according to the equation  
Next  
}}
```

[3]

3.2. Flow Chart of the Basic PSO



4. Related Work Done

- [1] *J. Kennedy and R. C. Eberhart, "Particle swarm optimization,"* This paper transplants the concept for the optimization of nonlinear functions using particle swarm methodology. The evolution of several paradigms was outlined. Benchmark testing of the paradigm was described, and the applications, including nonlinear function optimization and neural network training, were also proposed. The relationships between the particle swarm optimization and both artificial life and genetic algorithms were described in this paper.

- [2] *S.Vijayalakshmi, D.Sudha, S.Mercy Sigamani, K.Kalpana Devi, "Particle Swarm Optimization with Aging Leader and Challenges for Multiswarm Optimization"* In reality every organism ages and has a limited life span. Aging is important for maintaining the diversity. In nature the aging leader becomes weak which leaves opportunities to the other individuals to challenge the leadership position.
- [3] *Muhammad Imran , Rathiah Hashima and Noor Elaiza Abd Khalid'* "The authors concluded that there is a drawback in the PSO that it stuck in the local minima. To improve the performance of the PSO, the researchers proposed a number of different variants of PSO. Some of the methods include: by improving initialization of the swarm, introduce the new parameters like constriction coefficient and the inertia weight; defining the different method of inertia weight to improve performance of PSO, working on the global and local best particles by introducing mutation operators in the PSO. In this paper, the different variants of PSO with respect to initialization, inertia weight and mutation operators have been presented.
- [4] *Prabha Umapathy, C. Venkateshaiah,¹ and M. Senthil Arumugam , " Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution"* Particle swarm optimization is a heuristic global optimization method ,based on swarm intelligence. It is based on the research on the moving behaviour of the birds and fish flock. The algorithm is used for its easy implementation and few particles required to be tuned. It presents some kinds of improved versions of PSO.

5. Conclusion

Several parameters have been found to improve the performance of PSO. These include: initialization of population- If these particles be initialized correctly then the performance can be enhanced to a great extent, inertia weights- balance the exploration-exploitation trade off, mutation operators- the global best particle the local best particle with different techniques can be mutated. velocity clamping.- If the velocity of a particle exceeds the maximum allowed speed limit, it will set a maximum value of the velocity . Some environments are proposed which can improve the performance of PSO. These include: multi-objective optimization with PSO and dynamic environment with PSO. Many different variants of PSO have also been proposed here.

6. Future Scope

Some of the parameters are presented here but there are still more which can be unfolded to improve the performance of PSO (Particle Swarm Optimization) to much more extent. These parameters can be worked upon to form a more robust PSO algorithm.

References

- [1] J. Kennedy and R. C. Eberhart, "Particle swarm optimization", Proc. IEEE Int. Conf. Neural Networks., (1995) November-December, pp. 1942–1948.
- [2] S. Vijayalakshmi, D. Sudha, S. M. Sigamani and K. K. Devi, "Particle Swarm Optimization with Aging Leader and Challenges for Multiswarm Optimization", International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) vol. 3, no. 3, (2014) March.
- [3] M. Imran, R. Hashima and N. E. A. Khalid "An Overview of Particle Swarm Optimization Variants", Malaysian Technical Universities Conference on Engineering & Technology 209, MUCET 209, Procedia Engineering, vol. 53, (2013), pp. 491– 496.
- [4] P. Umapathy, C. Venkateshaiah and M. S. Arumugam, "Particle Swarm Optimization with Various Inertia Weight Variants for Optimal Power Flow Solution", Discrete Dynamics in Nature and Society vol. (2010).
- [5] S. S. Aote, "A Brief Review on Particle Swarm Optimization: Limitations & Future Directions", International Journal of Computer Science Engineering (IJCSE).

- [6] B. Wang and Z.-S. Wu, Z.-W. Zhao and H.-G. Wang, "Retrieving evaporation duct heights from radar sea clutter using particle swarm optimization (ps) algorithm", Progress in Electromagnetics Research M, ISSN: 1937-8726, vol. 9, pp. 79-91.
- [7] Particle Swarm Optimization by James McCaffrey.
- [8] W. N. Lee and J. B. Park, "Educational Simulator for Particle Swarm Optimization and Economic Dispatch Applications", IEEE Transactions on Power Systems 03/2005.
- [9] R. Forbes and M. N. Teli, "Particle Swarm Optimization on Multi-Funnel Functions".
- [10] Q. Bai, "Analysis of Particle Swarm Optimization Algorithm", vol. 3, no. 1, (2010) February.
- [11] D. P. Rini, S. M. Shamsuddin and S. S. Yuhaniz, "19 Particle Swarm Optimization: Technique, System and Challenges", International Journal of Computer Applications (0975 – 8887) vol. 14, no. 1, (2011) January.
- [12] A. E. Smith, "Swarm intelligence: from natural to artificial systems [book reviews]", IEEE Transactions on Evolutionary Computation, vol. 4, no. 2, (2000), pp. 192–193.
- [13] W.-N. Chen, J. Zhang, N. Chen, Z.-H. Zhan, H. S.-H. Chung, Y. Li and Y.-H. Shi "Particle Swarm Optimization with an Aging Leader and Challengers", Student and Senior Members, IEEE.
- [14] B. Santosa, "Tutorial Particle Swarm Optimization", (2006).

Authors



Avneet Kaur, she is pursuing M.Tech, Computer Science and Engineering from Guru Nanak Dev University, Regional Campus, Jalandhar (India). She has published four papers in journals and conferences.



Mandeep Kaur, she Lecturer, Computer Science department in Guru Nanak Dev University, Regional Campus, Jalandhar (India). She is an expert in the field of "System Simulation". Has an experience of 12 years and published 6 papers in conferences.