Particle Swarm Optimization: Development and Implementation

Pichet Sriyanyong*

Abstract

This scholarly article presents the development and implementation of a potential global optimizing algorithm, named Particle Swarm Optimization (PSO). In recent years, substantial efforts related to the applications of PSO to various areas in engineering problems have been carried out. Therefore, this article briefly gives the details of PSO development associated with the different research areas. Furthermore, it is aimed at encouraging the understanding about PSO searching characteristics; thus, the Schaffer's f6 benchmark function is also adopted for validation. Finally, the future trend of improving the PSO performance is suggested.

Keyword: Particle Swarm Optimization

1. Introduction

Particle Swarm Optimization (PSO) is one of the modern algorithms used to solve global optimization problems [1], and it is based on similar principles as the previous methods i.e. Genetic Algorithm (GA), Evolutionary Programming (EP), etc. Thus, to solve an optimization problem, PSO applies a simplified social model, which for instance Zoologists might use to explain the movement of individuals within a group [2]. To begin with, PSO initializes a population of random solutions each of which is defined as a "particle". Initially, every particle flies into a problem hyperspace at a random velocity. Thereafter, each particle adjusts its travelling speed dynamically corresponding to the flying experiences of itself and its colleagues [3, 4]. The PSO computation will keep updating the position of the particles until it finds a global optimal solution. Compared to other methods, application of the PSO is simple to implement, it can quickly find a number of high quality solutions, and has stable convergence characteristics [4, 5]. In addition, PSO is robust in solving continuous non-linear optimization problems, and contrary to other evolutionary algorithms it has a flexible and well-balanced mechanism for improving and adjusting the global and local search capabilities [6]. According to various key advantages over other optimization methods, that will result in increasing growth in various research articles as published in the Institute of Electrical and Electronics Engineers (IEEE) and the Institution of Electrical Engineers (IEE) databases [7].

However, there are some shortcomings of PSO algorithm, where it seems sensitive to the tuning of some of its weights or parameters. In addition, PSO can sometimes suffer from the lack of the diversity amongst the particles, which can lead to a stagnation stage [8]. Therefore, although PSO has been a subject of an extensive research, there are a number of issues that need to be addressed in order to exploit the full potential of PSO in solving complex engineering problems [5].

2. Main Categories of PSO Resarch areas

Recently, a number of studies have been conducted to develop suitable PSO algorithms that can be used to solve complex problems in various applications. These studies looked at different aspects of PSO improvements, and according to Eberhart and Shi [9], they can be classified into the following five main categories :

2.1 Algorithm Development

The original PSO algorithm was principally developed in order to solve non-linear continuous optimization problems [4, 7]; however, a discrete binary version of PSO [10] was introduced subsequently to solve non-linear discrete optimization problems. Moreover, PSO algorithms can be divided into the global version (Gbest model) and the local version (Lbest model) types, with the ability of the Lbest model to prevent a solution being trapped in local minima as illustrated in Figure 2. The Gbest model, on the other hand, has more chance to get trapped into a local optimum. However, the global version is superior to the local version in terms of the speed of convergence to the optimum solution and the computation time [4, 9, 11]. The global version will there fore be taken into account in this paper.

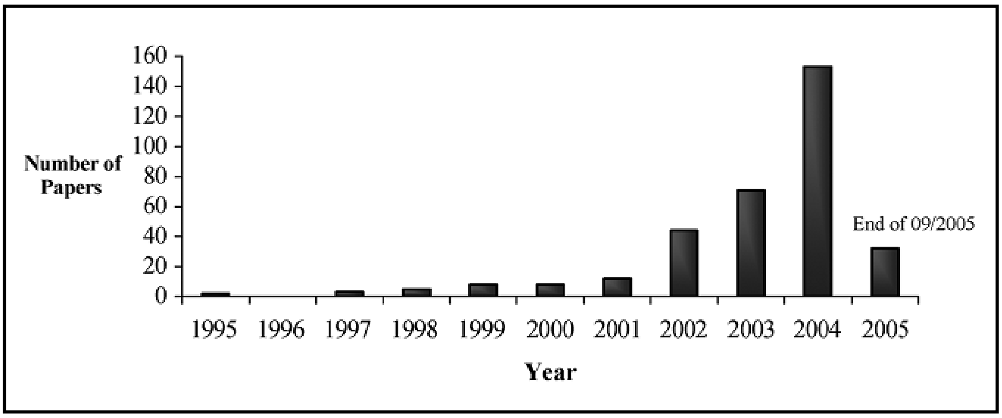

***Figure 1**: The trend of applying Particle Swarm Optimization in all research areas [7].*

* Department of Teacher Training in Electrical Engineering, Faculty of Technical Education, King Mongkut's University of Technology North Bangkok.

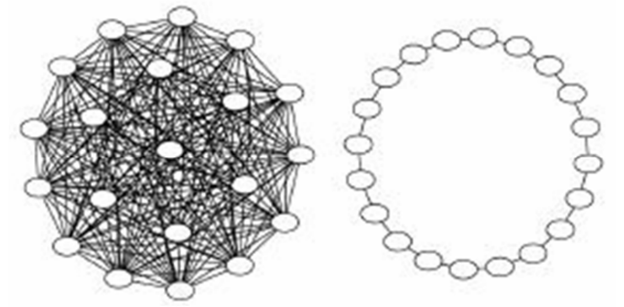

Figure 2: Example of PSO models [12].

2.2 Configuration of Topology

The aspect of a neighbourhood topology [12-15] examines the effect of different configurations or structures on PSO algorithm, i.e. circle topology, wheel topology, etc.

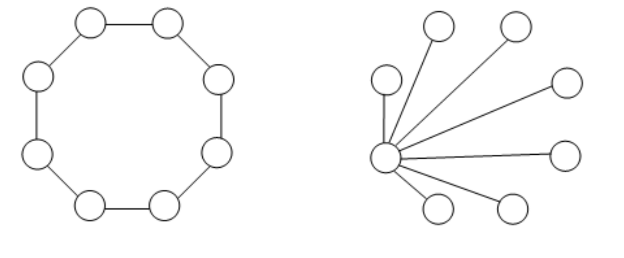

Figure 3: Example of topologies.

2.3 Parameters

As mentioned before, PSO is sensitive to the tuning of its parameters; therefore, proper setting of the parameters can significantly improve the searching capabilities of PSO methods [16]. Shi and Eberhart [17, 18] primarily introduced the incorporation of inertia weight factor (w) into the original PSO in order to balance the global and local explorations. Further, Clerc [19, 20] proposed an application of a constriction factor (k) to guarantee the convergence of the PSO.

2.4 Hybrid PSO

Often, PSO methodologies utilize the operators in the same manners as they are used in evolutionary computation techniques (i.e. selection, crossover and mutation) so as to avoid the stagnation problem that is a result of plunging into the suboptimal areas [9]. From the literature review, it can be seen that PSO with mutation operators can effectively optimise a wide range of engineering problems [21-23].

2.5 Applications

Due to easy implementation with less computation time [7], PSO has been extensively applied to a wide variety of the problems, i.e. engineering optimization problem with constraints [24], multi-objective optimization problems [25], sequencing problem [26], travelling salesman problem [27], navigation of mobile robot problem [28], control problem [29], scheduling problem [30], transportation problem [31], bin packing problem [32], etc.

3. Particle swarm optimization Algolithm

The PSO algorithms use a swarm of particles to represent the prospective solutions to an optimization problem under consideration. Each particle is treated as a zero-volume point in the n-dimensional search space and can be represented by a real-valued vector whose components are n-dimensional variables in the problem, respectively. The swarm of particles is flying with a velocity (the rate of its position change) to discover the global optimal position. Suppose there are m particles in the swarm and denote the swarm as $P = \{x_i \mid i = 1, \ldots, m\}$. . At the k-th stop, the i-th particle's position can be represented as:

$$x_i^k = (x_i^k(1), x_i^k(2), \ldots, x_i^k(j), \ldots, x_i^k(n)), \quad i = 1, 2, \ldots, m,\ j = 1, \ldots, n \tag{1}$$

and its velocity used for flying from the k-th to the (k+1)-th stop are denoted as:

$$v_i^{k+1} = (v_i^{k+1}(1), v_i^{k+1}(2), \ldots, v_i^{k+1}(j), \ldots, v_i^{k+1}(n)), \quad i = 1, 2, \ldots, m,\ j = 1, \ldots, n \tag{2}$$

There are various velocity-updating mechanisms used to update the velocity of the i-th particle as follows [33]:

3.1 Original PSO Algorithm (OPSO)

OPSO updates the velocity for the i-th particle after the k-th stop by:

$$v_i^{k+1} = v_i^k + c_1 \times rand1 \times (pbest_i^k - x_i^k) + c_2 \times rand2 \times (gbest^k - x_i^k), \quad i = 1, 2, \ldots, m \tag{3}$$

where c_1 and c_2 are generally set to 2, rand1 and rand2 are two uniformly distributed random values in the range [0,1]. $pbest_i^k$ is the best position of the i-th particle up to the k-th stop; $gbest^k$ is the best position of the swarm up to the k-th stop under the assumption that each particle has all the other particles as its neighbor.

3.2 Basic PSO (BPSO)

BPSO is OPSO with an inertia weight factor [17] and sometimes referred to as the standard PSO. It updates the velocity for the i-th particle after the k-th stop by:

$$v_i^{k+1} = w \times v_i^k + c_1 \times rand1 \times (pbest_i^k - x_i^k) + c_2 \times rand2 \times (gbest^k - x_i^k), \quad i = 1, 2, \ldots, m \tag{4}$$

where w is the inertia weight factor. In practice, w usually varies with the stops in the flying course to adjust the balance between global and local search. It is a common practice to set w to vary linearly between 0.9 and 0.4 during each run [34].

3.3 Constriction factor PSO (CPSO)

CPSO is OPSO with a constriction factor recommended by Clerc [19, 35, 36] to ensure convergence of the PSO algorithm. It updates the velocity for the i-th particle after the k-th stop by:

$$v_i^{k+1} = K[v_i^k + c_1 \times rand1 \times (pbest_i^k - x_i^k) + c_2 \times rand2 \times (gbest^k - x_i^k)], \quad i = 1, 2, \ldots, m \tag{5}$$

where K is the constriction factor, which can be determined by

$$K = \frac{2}{\left|2-\varphi-\sqrt{\varphi^2-4\varphi}\right|},\ \varphi = c_1 + c_2 \ ,\ \varphi > 4. \qquad (6)$$

where φ is generally set to 4.1, both $c_1 + c_2$ are set to 2.05 and K is 0.729 as presented in [36].

3.4 Original PSO with both inertia weight and constriction factor (CBPSO)

CBPSO is an algorithm that uses both inertia weight and constriction factor in updating velocities [22]. It updates the velocity for the *i*-th particle after the *k*-th stop by:

$$v_i^{k+1} = K[w \times v_i^k + c_1 \times rand1 \times (pbest_i^k - x_i^k) + c_2 \times rand2 \times (gbest^k - x_i^k)],\quad i = 1,2,\ldots,m. \qquad (7)$$

Consequently, the new positions at the (*k*+1)-th stop will be updated by the following equation:

$$x_i^{k+1} = x_i^k + v_i^{k+1},\quad i = 1,2,\ldots,m. \qquad (8)$$

Based on the updated position (Eq.(3)-(7)) and velocity (Eq.(8)), the *k*-th particle will fly to the (*k*+1)-th position as illustrated in Figure 4.

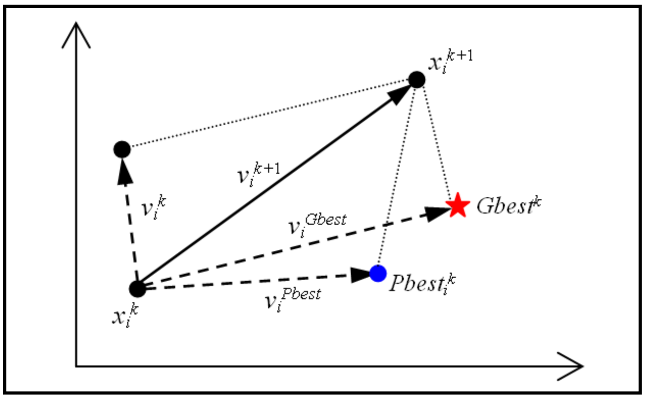

***Figure 4**: Basic concept of searching mechanism of PSO [2].*

4. Example of PSO implementation

Generally, it is possible to update the particle's velocity using mechanisms as mentioned in previous section. Since space limitation, the basic PSO (BPSO) or the standard PSO is selected for an example of implementation. The procedures of the implementation in this section can be expressed in details as follows:

Step 1. Initialization

- Determine the number of particles, *m*;
- Randomly generate *m* feasible particles to be the candidate solutions to the optimization problem;
- Set the termination criteria;
- Initialize $pbest_i$, $i = 1,2,\ldots,m$, to be the initial position of the *i*-th particle;
- Initialize *gbest* to be the best position of all particles in the swarm;
- Initialize the velocities of each particle randomly within the limit [$-V_{max}$, V_{max}], Generally, V_{max} is suggested to be 4 [37];
- Set the values of parameters c_1 and c_2;
- Set the starting and ending values for the weight factor, w.

Step 2. Update the velocity for each particle by Eq. (4) and make sure all its components within the limit its components within the limit [$-V_{max}$, V_{max}].

Step 3. Update the position for each particle using Eq. 8.

Step 4. Update the $pbest_i$ $i = 1,2,\ldots,m$ and *gbest*, respectively.

Step 5. Check the termination criteria. If the criteria are met, go to Step 6, otherwise,
go to Step 2.

Step 6. Output *gbest* as the solution to the problem.

5. Example of validation using the benchmark fuction

For better understanding about the searching behavior of the BPSO, a mathematical benchmark function, Schaffer's f6 function [38],is considered with the aim at searching for its global minimum. The expression and conditions of the benchmark function are tabulated in Table 1. Concerning the benchmark function, there are 2 dimensions with a number of local minima that is also plotted in three dimensions as shown in Figure 5. It is shown from the Figure 5 that the global minimum is zero, where x and y are zero as well. The simulations are implemented in Matlab, where parameters of PSO are initial inertia weight W_{max} = 0.9, final inertia weight W_{min} = 0.4, acceleration constants (c_1, c_2) = 2, population = 20, and V_{max} = 4, respectively.

***Table 1.** Optimization test function*

Function Name	Expression and Conditions
Schaffer's f6	$f(x,y) = 0.5 + \frac{(\sin\sqrt{x^2+y^2})^2 - 0.5}{(1.0+0.001(x^2+y^2))^2}$ Dimension = 2 $x, y \in [-100,100]$, min. ($\int$) = $\int$ (0,0) = 0.

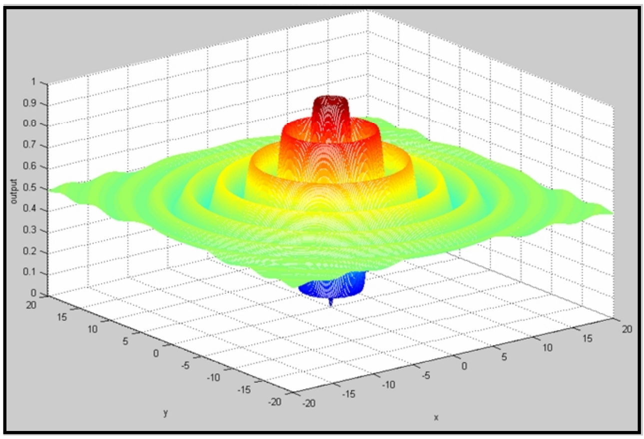

***Figure 5** :(a). The plot of Schaffer's f6 function in three dimensions.*

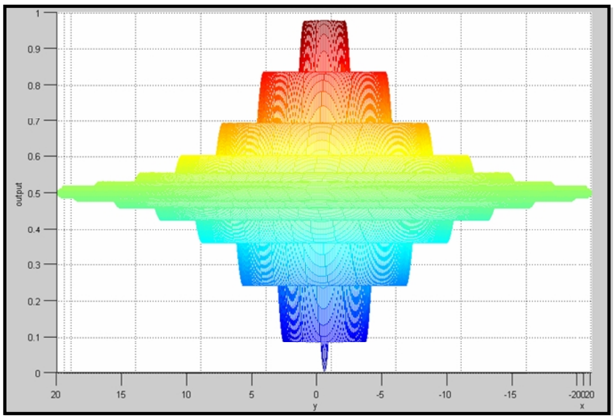

***Figure 5 (b)** : The minimum position of Schaffer's f6 function*

The example of experimental results are plotted in Figure 6-9 with two aspects: (1) the position of the particles and Gbest, (2) the convergence curve of the Gbest. The circle and star markers represent the particle's position and the Gbest's position, respectively. Figure 6 shows the 1st iteration of searching, where the position of the particles randomly generated 20 feasible particles. The best position among the particles (Gbest) is (-21.4125, -6.2482), where the fitness value of the Gbest position is 0.48867. Subsequently, the velocity and position of particles will be updated, whilst the new Gbest will be updated if it is less than the previous Gbest as illustrated in Figure 7. It can be seen from the Figure 8 that the group of particles will attempt to move toward Gbest's position. In this case, Gbest acts as the leader of the group. The searching process will be terminated if the predefined maximum number of generations is reached or without changing of the Gbest value for a number of predefined iterations as shown in Figure 9. Finally, the position among the particles (Gbest) is (-0.002, 0.037), where its fitness value is 0.001, that is close to the global minimum. However, the final solution is possible to be dissimilar for each run because PSO employs the principle of a random initialized population.

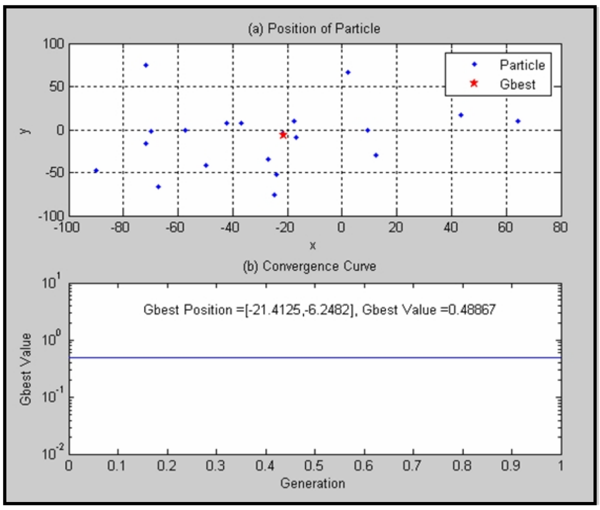

***Figure 6** : An example of PSO searching behavior in the 1st iteration.*

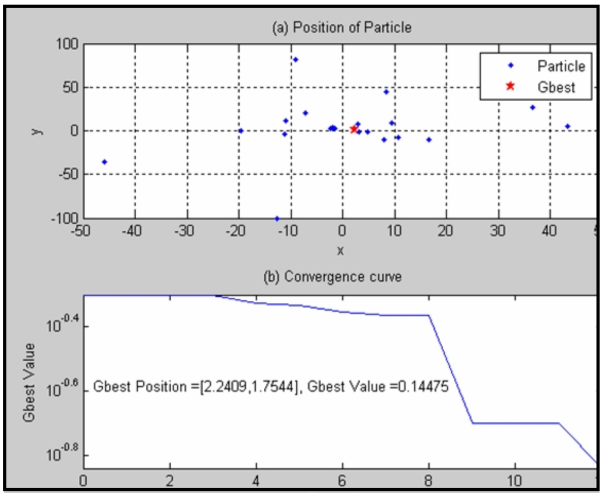

***Figure 7**: An example of PSO searching behavior in the 12th iteration.*

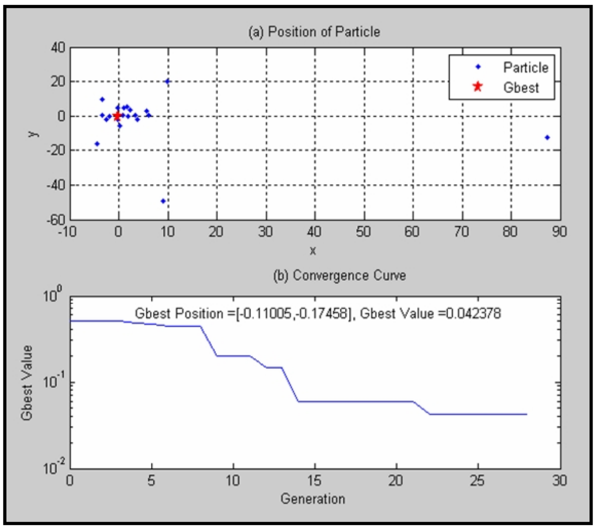

***Figure 8** : An example of PSO searching behavior in the 28th iteration.*

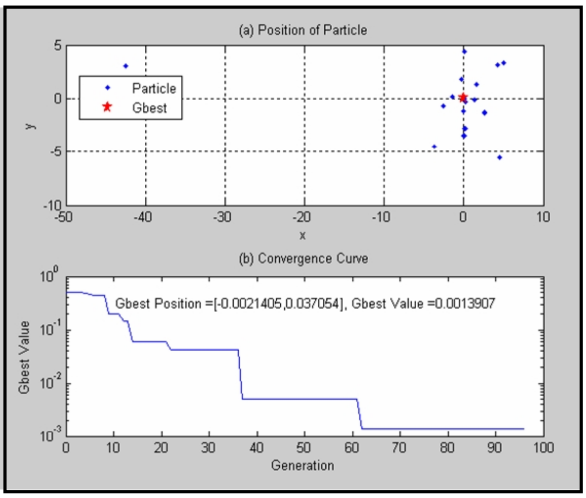

***Figure 9** : An example of PSO searching behavior in the 96th iteration.*

6. Conclusion, Discussion, and Suggestion

In this scholarly article, the development and implementation of the Particle Swarm Optimization (PSO) are introduced. The attractive features of PSO include: ease of implementation, fast convergence compared with the traditional evolutionary computation techniques; consequently, it is very suitable for solving a wide variety of the engineering problems. From the example of PSO validation, it can be seen that the PSO mayget trapped into a local optimum. This is reasonable because PSO have to balance its mechanism for improving and adjusting the global and local search i.e. the setting of its weight w linearly decreases from a big value to a small value at the final iteration. In other words, the PSO algorithm seems sensitive to the tuning of its weights or parameters for a particular problem, which may result in convergence to global solution. For that reason, it is the most importance to develop the new concept for overcoming this problem. The constriction factor PSO is also a promising method so as to guarantee the PSO convergence. In addition, the research can be further developed and extended in the concept of incorporate a particular operator (i.e. mutation operator) into the traditional PSO algorithms in such a way that the resulting algorithms are PSO-dominated and retain the attractive features of PSO while the particular operator acts as a fractional complement.

7. Acknowledgment

The author gratefully acknowledges financial support from King Mongkut's University of Technology North Bangkok, Thailand.

8. References

[1] B. Zhao, C. X. Guo, and Y. J. Cao, "A multiagent-based particle swarm optimization approach for optimal reactive power dispatch", *IEEE Trans. Power Syst,* vol. 20, pp. 1070 - 1078, May. 2005.

[2] J.-B. Park, K.-S. Lee, J.-R. Shin, and Kwang Y. Lee, "A particle swarm optimization for economic dispatch with nonsmooth cost functions", *IEEE Trans. Power Syst,* vol. 20, pp. 34-42, Feb. 2005.

[3] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO-SQP for economic dispatch with valve-point effect", *Electric Power Systems Research,* vol. 71, pp. 51-59, 2004.

[4] K. Y. Lee and M. A. El-Sharkawa, *A Tutorial Course on Evolutionary Computation Techniques for Power System Optimization. IFAC Symposium on Power Plants and Power,* Seoul Korea, Sep. 2003.

[5] Z.-L. Gaing, "Discrete particle swarm optimization algorithm for unit commitment", *IEEE Power Eng. Soc. General Meeting,* vol. 1, pp. 418-424, Jul. 2003.

[6] M. A. Abido, *Optimal power flow using particle swarm optimization,* International Journal of Electrical Power & Energy Systems, vol. 24, pp. 563-571, 2002.

[7] M. R. AlRashidi and M. E. El-Hawary, "A Survey of Particle Swarm Optimization Applications in Electric Power Systems", *IEEE Trans. on Evolutionary Computation: Accepted for future publication, Object Identifier 10.1109/TEVC.2006.880326,* pp. 1 - 1, 2006.

[8] A. Stacey, M. Jancic, and I. Grundy, "Particle swarm optimization with mutation", *in Proc. Congr. Evol. Compt.,* vol. 2, pp. 1425 - 1430, Dec. 2003.

[9] R. C. Eberhart and Y. Shi, "Guest Editorial Special Issue on Particle Swarm Optimization," *IEEE Trans. on Evolutionary Computation,* vol. 8, pp. 201 - 203, June. 2004.

[10] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm", *in Proc. of IEEE Inte. Conf. on Systems, Man, and Cybernetics,* vol. 5, pp. 4104 - 4108, Oct. 1997.

[11] P. S. Andrews, "An Investigation into Mutation Operators for Particle Swarm Optimization", *in Proc. Congr. Evol. Compt.*, pp. 1044 - 1051, July. 2006.

[12] J. Kennedy and R. Mendes, "Population structure and particle swarm performance", *In Proc. Congr. Evol. Compt.(CEC 2002),* vol. 2, pp. 1671 - 1676, May. 2002.

[13] J. Kennedy, "Small worlds and mega-minds: effects of neighborhood topology on particle swarm performance", *In Proc. Congr. Evol. Compt.(CEC 99),* vol. 3, pp.1931-1938, July. 1999.

[14] J. Kennedy and R. Mendes, "Neighborhood topologies in fully-informed and best-of-neighborhood particle swarms", *In Proc. Conf. IEEE International Workshop on Soft Computing in Industrial Applications (SMCia/03),* pp. 45 - 50, June. 2003.

[15] E. S. Peer, F. v. d. Bergh, and A. P. Engelbrecht, "Using neighbourhoods with the guaranteed convergence PSO", *In Proc. of Conf. on Swarm Intelligence Symposium (SIS 2003),* pp. 235 - 242, Apr. 2003.

[16] M.-P. Song and G.-C. Gu, "Research on particle swarm optimization: A review", *In Proc. Int. Conf. Machine Learning and Cybernetics,* vol. 4, pp. 2236 - 2241, Aug. 2004.

[17] Y. Shi and R. Eberhart, "A modified particle swarm optimizer", In Proc. Congr. Evol. Compt., pp. 69 - 73, May. 1998.

[18] Y. Shi and R. Eberhart, "Parameter Selection in Particle Swarm Optimization", *In Proc. of the 1998 Annual Conference on Evolutionary programming,* San Diego, 1998.

[19] M. Clerc, "The swarm and the queen: towards a deterministic and adaptive particle swarm optimization," *in Proc. Congr. Evol. Compt.*, vol. 3, pp. 1951-1957, July. 1999.

[20] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. on Evolutionary Computation,* vol. 6, pp. 58 - 73, Feb. 2002.

[21] P. J. Angeline, "Using selection to improve particle swarm optimization", *in Conf. of IEEE World Congress on Computational Intelligence,* pp. 84 - 89, May. 1998.

[22] N. Higashi and H. Iba, "Particle swarm optimization with Gaussian mutation", *in Proc. IEEE Swarm*

Intelligence Symposium (SIS'03), pp. 72 - 79, Apr. 2003.
[23] M. Lovbjerg, T. K. Rasmussen, and T. Krink, "Hybrid Particle Swarm Optimiser with Breeding and Subpopulations", *In Proc. the Genetic and Evolutionary Comunication Conference,* 2001.
[24] X. Hu, R. C. Eberhart, and Y. Shi, "Engineering optimization with particle swarm", *in Proc. IEEE Swarm Intelligence Symposium(SIS'03),* pp. 53-57, Apr. 2003.
[25] M. Salazar-Lechuga and J. E. Rowe, "Particle swarm optimization and fitness sharing to solve multi-objective optimization problems", *In Proc. Congr. Evol.Compt.,* vol. 2, pp. 1204 - 1211, Sept. 2005.
[26] L. Cagnina, S. Esquivel, and R. Gallard, "Particle swarm optimization for sequencing problems : a case study", *In Proc. Congr. Evol. Compt. (CEC '04),* vol. 1, pp. 536 - 541, June. 2004.
[27] C. Wang, J. Zhang, J. Yang, C. Hu, and J. Liu, "A Modified Particle Swarm Optimization Algorithm and its Application For Solving Traveling Salesman Problem", *In Proc. Int. Conf. Neural Networks and Brain (ICNN&B '05),* pp. 689 - 694, Oct. 2005.
[28] Y.-Q. Qin, D.-B. Sun, N. Li, and Y.-G. Cen, "Path planning for mobile robot using the particle swarm optimization with mutation operator", *In Proc. Int. Conf. Machine Learning and Cybernetics,* vol. 4, pp. 2473 - 2478, Aug. 2004.
[29] J. Chen, Z. Ren, and X. Fan, "Particle swarm optimization with adaptive mutation and its application research in tuning of PID parameters", *1st International Symposium on Systems and Control in Aerospace and Astronautics (ISSCAA 2006),* pp. 990-994, Jan. 2006.
[30] J. Zhu and X. Gu, "A New Particle Swarm Optimization Algorithm for Short-Term Scheduling of Single-Stage Batch Plants with Parallel Lines", *In Proc. Sixth International Conference on Intelligent Systems Design and Applications (ISDA '06),* vol. 2, pp. 673 - 678, Oct. 2006.
[31] Z.-F. Hao, H. Huang, and X.-W. Yang, "A Novel Particle Swarm Optimization Algorithm for Solving Transportation Problem", *In Proc. Int. Conf. Machine Learning and Cybernetics,* pp. 2178 - 2183, Aug. 2006.
[32] D. S. Liu, K. C. Tan, C. K. Goh, and W. K. Ho, "On Solving Multiobjective Bin Packing Problems Using Particle Swarm Optimization", *In Proc. Congr. Evol. Compt. (CEC '06),* pp. 2095 - 2102, Jul. 2006.
[33] R. C. Eberhart and Y. Shi, "Particle swarm optimization: developments, applications and resources", *in Proc. Congr. Evol. Comput.,* vol. 1, pp. 81 - 86, May. 2001.
[34] Y. Shi and R. C. Eberhart, "Empirical study of particle swarm optimization", *in Proc. Congr. Evol. Compt.,* vol. 3, pp. 1945-1950, Jul. 1999.
[35] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multidimensional complex space", *IEEE Trans. on Evolutionary Computation,* vol. 6, no. 1, pp. 58-73, Feb. 2002.
[36] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization", *in Proc. Congr. Evol. Compt.,* vol. 1, pp. 84 - 88, Jul. 2000.
[37] Y. Fukuyama, "Particle Swarm Optimization Techniques with applications in Power System", *in Evolutionary Computation Techniques for Power System Optimization,* K. Y. Lee and M. A. El-Sharkawa, Eds. Tutorial given at The IFAC Symposium on Power Plants and Power, Seoul Korea, pp. 45-62, Sep. 2003.
[38] T.-O. Ting, M. V. C. Rao, C. K. Loo, and S.-S. Ngu, "A new class of operators to accelerate particle swarm optimization", *In Proc. Congr. Evol. Compt. (CEC '03),* vol. 4, pp. 2406 - 2410, Dec. 2003.