

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

Ratanavilisagul, C., Kruatrachue, B. (2014). A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period. In: Boonkrong, S., Unger, H., Meesad, P. (eds) Recent Advances in Information and Communication Technology. Advances in Intelligent Systems and Computing, vol 265. Springer, Cham. https://doi.org/10.1007/978-3-319-06538-0_4

A Modified Particle Swarm Optimization with Dynamic Particles Re-initialization Period

Chiabwoot Ratanavilisagul¹ and Boontee Kruatrachue¹

¹ Department of Computer Engineering, Faculty of Engineering, King Mongkut's Institute of Technology Ladkrabang, Bangkok, Thailand
chaibwoot@hotmail.com, boontee@yahoo.com

Abstract. The particle swarm optimization (PSO) is an algorithm that attempts to search for better solution in the solution space by attracting particles to converge toward a particle with the best fitness. PSO is typically troubled with the problems of trapping in local optimum and premature convergence. In order to overcome both problems, we propose an improved PSO algorithm that can re-initialize particles dynamically when swarm traps in local optimum. Moreover, the particle re-initialization period can be adjusted to solve the problem appropriately. The proposed technique is tested on benchmark functions and gives more satisfied search results in comparison with PSOs for the benchmark functions.

Keywords: Particle Swarm Optimization, Particles Re-initialization, Mutation operator, Multi-start Particles.

1 Introduction

Kennedy and Eberhart [1, 2] introduced Particle Swarm Optimization (PSO) in 1995. It is motivated from the behavior of flying bird and their communication mechanism in solving optimization problems. In comparison with several other population-based Stochastic optimization methods, such as genetic algorithms (GAs) and evolutionary programming (EP), PSO performs better in solving a variety of optimization problems with fast and stable convergence rate [3-5]. In addition, PSO is capable of solving the function optimization problems [6-8], artificial neural network training [9, 10], as well as pattern classification and fuzzy system control [11, 12].

PSO has many advantages such as rapid convergence, simplicity, and little parameters to be adjusted. Its main disadvantage is trapping in local optimum and premature convergence. To overcome both problems, many researchers [13-19]

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

increased searching diversity in the population of PSO by adding the particle re-initialization or the mutation operators in the process of PSO. The result showed that both methods can increase PSO optimization performance.

However, the particle re-initialization encounter parameters adjusting to suitably deal with any given problem. As inappropriate parameters could contribute to poorer results vis-à-vis the standard PSO. The essential parameter of particle re-initialization is the particles re-initialization period (re-initialization period) which is the amount of time to be certain that the swarm trapped in local optimum and the reinitialized particles with PSO should be performed.

Researchers proposed various ways to define re-initialization period using the number of iteration with the best position unchanged [14], and the number of iteration with the consecutive neighborhood best position unchanged [15].

This paper investigates adjusting re-initialization period to suitably deal with any given problem applied with PSO. Normally, the reinitialized particles (resetting) should occur when swarm traps in local optima. Hence, this paper proposes the re-initialization period defined by the unchanged number of consecutive best particle which indicates the state of local optima trapping. Moreover, re-initialization period is automatically adjusted according to the difference of the best solution found after resetting and the previous best solution. A set of benchmark test functions is used to compare the standard PSO, the proposed PSO algorithm, PSO algorithm's [14], and PSO algorithm's [15]. The results show that the proposed PSO algorithm obtains the best results in all test functions.

The rest of this paper is organized as follows. Section 2 explains basic PSO and PSO with particles re-initialization. Section 3 explains PSO with dynamic particles re-initialization period. Section 4 explains the benchmark functions, the experiment setup and presents the experiment results. Section 5 concludes the paper with a brief summary.

2 Related work

2.1 Particle Swarm Optimization

In standard PSO, each member of the population is called a “particle” with its own position and velocity. Each individual particle performs searching in the search space according to its velocity, the best position found in the whole swarm (GBEST) and the individual's best position (PBEST). The standard PSO algorithm starts with randomizing particle positions and their respective velocities, and the evaluation of the position of each particle is achieved using the objective function of the optimization problem. In a given iteration, each individual particle updates its position and velocity according to the expression below:

$$V_{id}' = \omega V_{id} + \eta_1 rand() (P_{id} - X_{id}) + \eta_2 rand() (P_{gd} - X_{id}) \quad (1)$$

$$X_{id}' = X_{id} + V_{id}' \quad (2)$$

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

Where X'_{id} denotes the current positions of i particle and d dimension, X_{id} the previous positions of i particle and d dimension, V_{id} the previous velocity of i particle and d dimension, V'_{id} the current velocity of i particle and d dimension, P_{id} PBEST of i particle and d dimension, and P_{gd} GBEST of d dimension. $0 \leq \varpi < 1$ is an inertia weight, η_1 and η_2 are acceleration constants, and $\text{rand}()$ generates random number from interval $[0,1]$. A limit velocity calls $V_{\max}$. If calculate velocity of a particle exceeds this value, it will replace value of $V_{\max}$.

2.2 PSO with Particles Re-initialization

The particle re-initialization techniques prevent premature convergence and trapping in local optimum. They create a variation of a population. Many researchers proposed various techniques such as:

Den Bergh [13] proposed Multi-start PSO (MPSO). The concept of this algorithm can be summarized as follows: after some iteration with the maximum swarm radius less than some threshold value, all particles of swarm are initialized in order to increase the diversity of swarm and expand the searching space. This algorithm can jump out of the local minimum and converge to the new area. However, this technique destroys the current structure of the swarm with total re-initialization of particles which results in decreasing the convergence speed and the search accuracy [14, 15].

In fact, the swarm structure is not totally destroyed because GBEST is not reset. As a result, the re-initialized swarm is position very far from the GBEST and will rapidly converge to the GBEST. If swarm can locate a better position than the previous GBEST, the better position is then occupied as a new GBEST. However, MPSO has difficulty in adjusting suitably re-initialization period. Applying MPSO with general problems is complicated. In addition, the results from non-reset PBEST contribute to non-distributed searching with rapid convergence, so MPSO is rarely improve the searching performance of PSO.

Ning Li et.al. [14] proposed PSO with mutation by considering from the Euclidean distance and GBEST (PSOMEG). The concept of this algorithm can conclude as follows: The velocity and position of particles are initialized depend on mutation probability. The initialization occur when GBEST unchanged more than some the threshold value (MaxStep) and the maximum Euclidean distance between all the particles is less than the threshold value (BorderRadius). This technique improves the diversity of swarm without decreasing the convergent speed and the search accuracy. However, PSOMEG has three additional parameters from standard PSO. MaxStep must be in some range, otherwise the results is poor. BorderRadius is very difficult to set with any given problem. Mutation probability can be set to over-mutation which results in similar searching to MPSO. On the other hand, set to under-mutated results in remain trapping. The difficulty of parameters setting reduces the practicality of PSOMEG with general problems.

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

Xuedan Liu et.al. [15] proposed PSO with mutation by considering from the neighborhood's best position (PSOMN). In this paper, the velocity and position of particles is initialized when the unchanged numbers of the consecutive neighborhood best position are more than some threshold value (MaxStep). PSOMN increase the global search capability and avoid premature convergence. However, selection of MaxStep is recommended within some range to obtain good results.

3 PSO with Dynamic Particles Re-initialization Period

As previously mentioned, the main problem of applying particles re-initialization technique is parameters adjusting. To avoid this problem, the proposed algorithm automatically adjusts its parameters. In addition, the number of parameters is kept minimal.

The trapping in local optimum is possible for standard PSO, which leads to stagnation of the searching in which the solution (GBEST) obtained at the point of trapping, is repeatedly produced irrespective of the length of searching time. Therefore, the unchanged number of consecutive GBEST can indicate state of swarm when swarm traps in local optimum.

Normally, the particles re-initialization (resetting) should be used when swarm trap or trend to trap in local optimum in order to distribute particles to search in other area. The easy techniques to indicate the trapping state of the swarm is to monitor the unchanged number of consecutive GBEST. Thereby, this paper proposed the particles re-initialization period (re-initialization period) is considered from the unchanged number of consecutive GBEST.

If re-initialization period was set too small, particles will be reset before convergence. On the other hand, if re-initialization period was set too large, particles remain trapping in local optimum instead of search in other areas.

Therefore, this paper proposes the novel technique which can decrease problem of adjusting re-initialization period. The proposed technique can adjust the re-initialization period to solve any optimization problems appropriately. The concept of proposed technique is explained as follows: Before reset, if the best solution in current resetting round is worse than previous round GBEST, particles could not converge to the GBEST. Hence, the re-initialization period should be increased to enhance convergence time of particles to the GBEST in next resetting round. On the other hand, if the best solution in current resetting round equals GBEST, particles can converge to the GBEST. In this case, the re-initialization period should be decreased to reduce the searching time in local optimum. In addition, the proposed technique choose to initialize all particles to improve chance in obtain better solution than initializing some particles. Because the more the number of particles reset, the better chance in obtaining the new GBEST while particles return to GBEST. Furthermore, PBEST of all particles should be reset to enhance distribution of particles. The proposed technique is called PSO with dynamic particles re-initialization period (PSODR). Pseudo code of PSODR is shown below:

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

```
Initial position and velocity of each particle
Initial PBEST GBEST RBEST
While termination condition ≠ true do
  For each particle
    Evaluate particle fitness
    If fitness of each particle is better than that of PBEST
      Update PBEST ( $P_i = X_i$ )
    End If
    If fitness of each particle is better than that of GBEST
      Update GBEST ( $P_g = X_i$ )
    End If
    If fitness of each particle is better than RBEST
      Update RBEST (RBEST =  $X_i$ )
      Set the times of RBEST consecutive unchanged to 0
    End If
    Update particle position according to formula (1) and (2)
  End For
  Times of RBEST consecutive unchanged ++;
  If the times of RBEST consecutive unchanged ≥ PRP
    If fitness of RBEST is worse than fitness of GBEST
      PRP = PRP + NPRP
    Else
      PRP = PRP - NPRP
    End If
    If PRP is less than the MINPRP
      PRP = MINPRP
    Else If PRP is more than the MAXPRP
      PRP = MAXPRP
    End If
    Reset RBEST
    Reinitialize PBEST, velocity, and position of all particles
    Set the times of GBEST consecutive unchanged to 0
  End If
End while
```

PRP is the particle re-initialization period (re-initialization period). RBEST is the best solution in each resetting round. NPRP is adjusting round number of the particle re-initialization period (this is the dynamic adjustment of PRP). MINPRP is

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

the minimum value of the particle re-initialization period. MAXPRP is the maximum value of the particle re-initialization period.

4 Experiments and Results

4.1 Benchmark Functions

The proposed algorithm is tested on nine well-known benchmark functions, listed in Table 1. These functions consists of six multimodal functions from function one to six. The remaining three functions are unimodal functions. The results of PSODR on the benchmark functions are compared with PSO [1, 2], PSOMEG [14] and PSOMN [15]. To guarantee fairness, the inertia weigh of PSOMN algorithm cannot be adapted as written in original paper. The inertia weigh of PSOMN algorithm is fixed as other algorithms.

4.2 Parameters Setting

Parameters are as follows for all experiments: acceleration constants of η_1 and η_2 are both set to be 1.496180 and inertia weight $\omega = 0.729844$, as suggested by den Bergh [13], the number of population is 20. The number of experiments of each function is 100 runs. The maximum iteration is 5000 independently of number of resetting. Hence, the algorithm that locates local optimum faster has more number of trial runs.

The non-PSO parameters are as follows: For PSODR algorithm, MINPRP = 10, MAXPRP = 1000, NPRP = 50, PRP = 200. The compared algorithm parameters are set according to suggested by the original papers. For PSOMEG algorithm, the mutation probability = 0.04, MaxStep = 10, the divided number in calculation of BorderRadius is 1000. For PSOMN algorithm, MaxStep = 30, the neighborhood length = 2.

Table 1. Details of Benchmark Test Functions.

Function name	Expression
Ackley	$f(x) = -20 \exp(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$
Griewank	$f(x) = \sum_{i=1}^n \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(x_i / \sqrt{i}) + 1$
Rastrigrin	$f(x) = 10n + \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i))$
Rosenbrock	$f(x) = \sum_{i=1}^n [100(x_i^2 - x_{i+1}) + (x_i - 1)^2]$
Schwefel	$f(x) = 418.9829 \times n + \sum_{i=1}^n (x_i \times \sin(\sqrt{ x_i }))$
Exponential	$f(x) = -\exp(-0.5 \sum_{i=1}^n x_i^2) + 1$
Sphere	$f(x) = \sum_{i=1}^n x_i^2$
Parallel Ellipsoid	$f(x) = \sum_{i=1}^n (i \times x_i^2)$
Rotated Ellipsoid	$f(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j^2)$

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

Table 1. (continued).

Function name	Search space $[X_{\max}, X_{\min}]^n$	Objective function value	Dim.	$V_{\max}$
Ackley	$x \in [-32.768, 32.768]^n$	0	40	32.768
Griewank	$x \in [-300, 300]^n$	0	40	300
Rastrigrin	$x \in [-5.12, 5.12]^n$	0	40	5.12
Rosenbrock	$x \in [-2.048, 2.048]^n$	0	40	2.048
Schwefel	$x \in [-500, 500]^n$	0	40	500
Exponential	$x \in [-1, 1]^n$	0	40	1
Sphere	$x \in [-5.12, 5.12]^n$	0	40	5.12
Parallel Ellipsoid	$x \in [-5.12, 5.12]^n$	0	40	5.12
Rotated Ellipsoid	$x \in [-65.536, 65.536]^n$	0	40	65.536

This research is conducted by a personal computer of Intel Core i5 2450 with a 2.5-GHz CPU and 8 GB RAM, and Visual C++ as the programming language. The measures of algorithm performance in the experiments are as follows:

The mean best fitness value (MBF) is the mean of best fitness in the final iteration from all running (100 runs). MBF indicates the solution searching efficiency of an algorithm. All experiments benchmark functions have zero results as a minimum point. In Table 1, an entry less than 10^{-324} is given the value of zero. The closer the MBF to the zero point of a method, the better the method.

The mean particles re-initialization period (MPRP) is the mean of particles re-initialization period in the final iteration from all running. MPRP is the result of the calculation of PSODR algorithm.

SD is the standard deviation.

The mean of time per mean of fitness (TPF) is the mean of time in running multiply the mean of fitness of GBEST in the final iteration from all running. The closer the TPF to the zero point of a method, the better the method. Because, this method loses a less time in the solution searching and obtains a good solution.

Table 2. The results of PSODR by fixed PRP on Benchmark Test Functions averaged over 100 runs.

Techniques	PSO	PSODR	PSODR	PSODR	PSODR	PSODR	PSODR
PRP	-	2	10	100	200	500	1000
Problems	MBF	MBF	MBF	MBF	MBF	MBF	MBF
Ackley	5.87489	5.20859	2.23955	2.16283	2.22361	2.89651	3.23259
Griewank	0.15754	1.59019	0.044938	0.0445354	0.048823	0.046841	0.052307
Rastrigin	158.724	332.885	135.183	139.055	145.385	146.352	150.278
Rosenbrock	9.67E-22	31.8709	2.10E-12	3.95E-24	4.89E-23	2.08E-20	1.37E-20
Schwefel	3979.11	3429.55	3143.13	3175.25	3102.46	3338.6	3614.64
Exponential	2.56E-15	0.0126073	2.92E-15	2.59E-16	2.24E-16	2.22E-16	2.45E-16
Sphere	3.66E-37	0.698723	2.13E-18	7.98E-35	2.43E-36	4.77E-37	2.04E-37
Parallel Ellipsoid	4.18E-36	30.2192	2.76E-14	5.29E-37	1.23E-36	3.28E-34	4.35E-37
Rotated Ellipsoid	2.42E-37	4651.77	2.94E-12	9.04E-35	1.20E-34	5.28E-37	2.30E-36

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

Table 3. Comparative results of PSO, PSOMEG, PSOMN, and PSODR on Benchmark Test Functions averaged over 100 runs.

Techniques	PSO	PSOMEG	PSOMN	PSODR	
Problems	MBF	MBF	MBF	MBF	MPRP
Ackley	5.87489	5.6928	3.2664	2.21957	103
Griewank	0.15754	0.133446	0.103264	0.04741	78
Rastrigin	158.724	153.939	148.766	138.981	67.8
Rosenbrock	9.67E-22	1.49E-19	2.60E-20	6.47E-23	192
Schwefel	3979.11	3910.7	3857.61	3091.22	66
Exponential	2.56E-15	1.15E-13	7.08E-14	2.71E-16	43.5
Sphere	3.66E-37	8.32E-36	1.51E-34	3.42E-37	200
Parallel Ellipsoid	4.18E-36	1.55E-34	1.04E-33	1.84E-36	200
Rotated Ellipsoid	2.42E-37	4.51E-36	9.08E-35	1.77E-37	200

Table 4. Comparative results of SD and TPF of PSO, PSOMEG, PSOMN, and PSODR on Benchmark Test Functions averaged over 100 runs.

Techniques Problems	PSO		PSOMEG		PSOMN		PSODR	
	SD	TPF	SD	TPF	SD	TPF	SD	TPF
Ackley	2.7838	1.2855	2.6532	1.3634	3.104	0.8522	0.5379	0.5029
Griewank	0.39	0.03535	0.1375	0.03749	0.2525	0.02883	0.04894	0.0108
Rastrigin	29.4346	33.4701	29.1122	33.143	31.5726	32.2524	25.87	29.3666
Rosenbrock	7.46E-13	1.54E-22	1.52E-20	2.48E-20	9.69E-15	4.24E-21	3.34E-21	1.03E-23
Schwefel	1063.64	1001.14	975.27	1006.73	955.22	1006.0646	933.454	791.661
Exponential	2.78E-14	3.67E-16	7.40E-14	1.66E-14	1.63E-12	1.02E-14	7.44E-17	3.912E-17
Sphere	1.78E-31	5.08E-38	1.59E-36	1.18E-36	6.44E-33	2.17E-35	3.47E-38	4.8E-38
Parallel Ellipsoid	1.41E-33	5.91E-37	2.64E-35	2.24E-35	2.37E-26	1.4823E-34	1.42E-36	2.622E-37
Rotated Ellipsoid	1.07E-35	5.53E-38	2.10E-35	1.03E-36	1.13E-28	2.0893E-35	3.20E-36	3.9205E-38

From the experimental results of Table 2 show both unimodal functions and multimodal functions, if the setting re-initialization period is 2, swarm will be over-reset. Its searching solution is poorer than that of the standard PSO, such as Griewank function, Rastrigin function, Rosenbrock function, Exponential function, Sphere function, Parallel Ellipsoid function, and Rotated Ellipsoid function. This is due to particles hardly converge. In the case of, re-initialization period is more than or equal to 100 in unimodal functions, overall its searching solutions are similar to that of the standard PSO since the particles re-initialization algorithms is not executed. For multimodal problems, if the setting re-initialization period is 1000, swarm will be under-reset. Its searching solution is poorer than setting re-initialization period to 100. Since swarm wastes time searching in local optimum rather than starting new search in other areas.

For multimodal problems in Table 2, PSODR gets good results because of the re-initialization period setting ranged from 10 to 200. The PSODR algorithm can adjust re-initialization period in this range and it can obtain the good solutions as

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

shown in Table 3. Therefore, the PSODR algorithm can dynamically adjust to suitably deal with the problem.

From the experimental results of Table 3 show that PSODR outperforms PSO, PSOMEG, and PSOMN in multimodal function. PSODR produces a better quality solution because of its lowest MBF in multimodal function. For unimodal function, PSO and PSODR have similar overall results. The MBF results of PSO and PSODR are higher than that of PSOMEG and PSOMN because PSOMEG and PSOMN could execute the particles re-initialization which will hinder the convergence of particles in unimodal function. For multimodal function, some MBF results of PSOMEG and PSOMN is poorer than the standard PSO such as Exponential function and Rosenbrock function. If adjusting re-initialization period to deal with problems appropriately, the solutions of PSOMEG and PSOMN can obtain better than that of the standard PSO. However, adjusting re-initialization period of PSOMEG and PSOMN to deal with any given problem appropriately is complicated.

From the experimental results of Table 4 show that the reliability of PSODR is better than that of PSO, PSOMEG, and PSOMN because of its lowest SD in all test functions. Moreover, PSODR obtains the best results when the results from the solution searching are compared with lost time in the solution searching. Because, TPF of PSODR is less than that of PSO, PSOMEG, and PSOMN in all test functions.

5 Conclusion

The particles re-initialization is applied with PSO to solve problems of trapping in local optimum and premature convergence. However, applying the particles re-initialization with general problems is complicated which is the cause of complicatedly adjusting the particle re-initialization period in order that the particles re-initialization can enhance the solution searching performance and can be easily applied to general problems. This paper proposes the re-initialization period that is automatically adjusted according to the difference of the best solution found after resetting and the previous best solution. The proposed technique is called PSODR. On benchmark functions, the proposed PSODR, PSO, PSOMEG and PSOMN are tested and the results are compared. From the experimental results show that PSODR can adjust the particle re-initialization period to solve the problem appropriately and PSODR outperforms PSO, PSOMEG, and PSOMN.

References

1. J. Kennedy and R. C. Eberhart, Particle Swarm Optimization, IEEE International Conference on Neural Networks, pp.1942-1948 (1995)
2. R. C. Eberhart and J. Kennedy, A New Optimizer Using Particle Swarm Theory, Proceedings of the 6th International Symposium on Micro Machine and Human Science, pp.39-43 (1995)

The text within this document is the similar to the original document.
If you read this document, please refer to my research.

3. K. E Parsopoulos, V. P. Plagianakos, G. D. Magoulas, M. N. Vrahatis, Objective function stretching to alleviate convergence to local minima, *Nonlinear Analysis TMA* 47, pp. 3419-3424 (2003)
4. R. Eberhart and Y. Shi, Comparison between genetic algorithms and particle swarm optimization, *The 7th Annual Conference on Evolutionary Programming*, San Diego (1998)
5. J. Kennedy, R.C. Eberhart, *Swarm Intelligence*, Morgan Kaufmann, San Mateo (2001)
6. V.S. Pappala, I. Erlich, K. Rohrig, J.A. Dobschinski, A stochastic modal for the optimal operation of a wind-thermal power system, *IEEE Trans. Power Syst.*, pp. 940-950, May (2009)
7. P. Acharjee, S.K. Goswami, A decoupled power flow algorithm using particle swarm optimization technique, *Energy Convers. Manage.*, pp. 2351-2360 (2009)
8. S. Dutta, S.P. Singh, Optimal rescheduling of generators for congestion management based on particle swarm optimization, *IEEE Trans. Power Syst.*, (2008)
9. S. Kiranyaz, T. Ince, A. Yildirim, M. Gabbouj, Evolutionary artificial neural networks by multi-dimensional particle swarm optimization, *Neural Networks*, pp. 1448-1462 (2009)
10. H.L. Wei, S.A. Billings, Y.F. Zhao, L.Z. Guo, Lattice dynamical wavelet neural networks implemented using particle swarm optimization for spatio-temporal system identification, *IEEE Trans. Neural Network*, pp. 181-185 (2009)
11. C.J. Lin, C.H. Chen, C.T. Lin, A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications, *IEEE Trans. Syst. Man Cybernetics Part C*, pp. 55-68 (2009)
12. L. Zhao, F. Qian, Y. Yang, Y. Zeng, H. Su, Automatically extracting T-S fuzzy models using cooperative random learning particle swarm optimization, *Appl. Soft Comput.*, pp. 938-944 (2010)
13. F. van den Bergh. An Analysis of Particle Swarm Optimizers. PhD thesis, Department of Computer Science, University of Pretoria, South Africa (2002)
14. Ning Li, Debao Suan, Yigang Cen, and Tong Zou, Particle swarm optimization with mutation operator, *Computer Engineering and Applications*, vol. 17, pp. 12-14 (2004)
15. Xuedan Liu, Qiang Wang, Haiyan Liu, Lili Li, Particle Swarm Optimization with Dynamic Inertia Weight and Mutation, *Third International Conference on Genetic and Evolutionary Computing*, pp. 620-623 (2009)
16. Chiabwoot Ratanavilisagul, Boontee Kruatrachue, Mutation Period Calculation for Particle Swarm Optimization, *1st International Symposium on Technology for Sustainability*, pp. 213-216 (2011)
17. Andrews P S. An investigation into mutation operators for particle swarm optimization. In: *Proceedings of the IEEE Congress on Evolutionary Computation*. Vancouver, Canada: IEEE, pp. 1044-1051 (2006)
18. MO Lin, ZHENG Hua. Improved PSO Algorithm with Adaptive Inertia Weight and Mutation, *2009 World Congress on Computer Science and Information Engineering*, pp. 622-625 (2009)
19. N. Higashi and H. Iba. Particle swarm optimization with Gaussian mutation, *Proc. of the 2003 IEEE Swarm Intelligence Symposium*, pp. 72-79 (2003)