

A MODIFIED PARTICLE SWARM OPTIMIZATION WITH RANDOM ACTIVATION FOR INCREASING EXPLORATION

^aAlrijadjis, ^bShenglin Mu, ^cKanya Tanaka, ^dShota Nakashima

^aPoliteknik Elektronika Negeri Surabaya

^bHiroshima National College of Maritime Technology

^{c,d}Yamaguchi University

E-Mail: alrijadjis@pens.ac.id

Abstract

Particle Swarm Optimization (PSO) adalah teknik optimumisasi yang diinspirasi dari perilaku kawanan burung atau ikan dalam mencari makanan. Teknik ini cukup terkenal sebagai algoritma pencarian metaheuristik yang baru dan diperkenalkan pertama kali oleh Eberhart dan Kennedy tahun 1995. Tetapi, PSO tipe standar memiliki kelemahan, yaitu konvergensi prematur dan mudah terjebak dalam local optimum. Inertia weight adalah salah satu parameter penting dalam PSO yang sangat mempengaruhi performansi PSO. Banyak strategi pengaturan inertia weight yang telah dikembangkan untuk mengatasi kelemahan PSO. Makalah ini membahas modifikasi PSO yang baru dengan menggunakan aktivasi acak untuk meningkatkan kemampuan eksplorasi, membantu partikel yang terjebak dalam local optimum dan menghindari konvergensi prematur. Dalam metode ini, inertia weight diturunkan secara linear sampai setengah iterasi maksimum, lalu bilangan acak sebagai inertia weight digunakan sampai akhir iterasi. Untuk membedakan dengan metode sebelumnya, PSO modifikasi ini disebut PSO modifikasi dengan aktivasi acak atau modified PSO with random activation (MPSO-RA). Eksperimen dengan tiga fungsi uji yang terkenal menunjukkan bahwa akurasi dan tingkat keberhasilan dari MPSO-RA meningkat 43,23% dan 32,95% dibandingkan dengan PSO standar.

Kata kunci: *Particle Swarm Optimization, inertia weight, premature convergence, local optimum, random activation.*

Abstract

Particle Swarm Optimization (PSO) is a popular optimization technique which is inspired by the social behavior of birds flocking or fishes schooling for finding food. It is a new metaheuristic search algorithm developed by Eberhart and Kennedy in 1995. However, the standard PSO has a shortcoming, i.e., premature convergence and easy to get stuck or fall into local optimum. Inertia weight is an important parameter in PSO, which significantly affect the performance of PSO. There are many variations of inertia weight strategies have been proposed in order to overcome the shortcoming. In this paper, a new modified PSO with random activation to increase exploration ability, help trapped particles for jumping-out from local optimum and avoid premature convergence is proposed. In the proposed method, an inertia weight is decreased linearly until half of iteration, and then a random number for an inertia weight is applied until the end of iteration. To emphasis the role of this new inertia weight adjustment, the modified PSO paradigm is named Modified PSO with random activation (MPSO-RA). The experiments with three famous benchmark functions show that the accuracy and success rate of the proposed MPSO-RA increase of 43.23% and 32.95% compared with the standard PSO.

Key words: Particle Swarm Optimization, inertia weight, premature convergence, local optimum, random activation.

INTRODUCTION

It is too difficult to solve optimization problems in large-scale complex engineering using conventional optimization technique. It needs a lot of time or work. The difficulty has contributed to the development of new optimization technique. To overcome the problem, an optimization technique based on intelligent computation for searching near-optimum solution to problem has been developed.

Particle Swarm Optimization (PSO) is one of the intelligent computational optimization technique introduced by Kennedy and Eberhart in 1995 [1]. It is a famous population-based stochastic search algorithm inspired by the social behavior of the birds flocking or fish schooling. The basic idea of PSO is derived from the research of the social behavior for the bird swarm or fish schooling to find food. PSO has shown good performance in finding better solution to optimization problems, and turned out to be another powerful tool besides other evolutionary algorithm such as Genetic Algorithm (GA) [2][3]. Compared with other optimization techniques, e.g., GA, Ant Colony Optimization (ACO), PSO has simple algorithm or form, faster convergence, efficient in time-calculation and is easily implemented as well as the adjustable parameters are few, so PSO is adept in solving many non-derivative and multi-peak complex optimization problems. PSO has been successfully applied to many sciences and practical fields [4][5][6][7].

Although PSO has superior features, it does exhibits some disadvantages: it is sometimes easy to be trapped in local optimum, and the convergence rate decreased considerably in the later period of evolution; when reaching a near optimum solution, the algorithm stops optimizing, and thus the accuracy of the algorithm is limited [8][9]. Many efforts have been made to overcome the problem. Among them, many approaches and strategies are proposed to enhance the performance of PSO via adjusting inertia weight, a most important parameter in PSO, such as linearly decreasing inertia weight (LDW-PSO) [8], nonlinearly decreasing inertia weight (NDW-PSO) [10], random inertia weight (RIW-PSO) [11], sigmoid increasing inertia weight (SIW-PSO) [11], sigmoid decreasing inertia weight (SDW-

PSO) [12], etc. PSO using linearly decreasing inertia weight (PSO-LDW) is most commonly used or standard type of PSO and it can improve the performance of PSO to some extent, but it may be trapped in local optimum and fail to achieve high search accuracy.

In this paper, a new modified particle swarm optimization using random activation (MPSO-RA) is proposed. In particular, random activation is used because of its effectiveness in helping the trapped particles to jump out from the local optimum. Firstly, in this algorithm, a standard strategy for adjusting an inertia weight, i.e., linearly decreased inertia weight is used until a half of iteration. It was reported that after a half of iteration, the particles tend to converge and their speed dropped dramatically. Then, a random activation is applied to generate a new inertia weight randomly in order to help particles jump out from local optimum. Finally, the superiority of the proposed algorithm is verified in numerical simulation.

PARTICLE SWARM OPTIMIZATION (PSO)

PSO is a population-based stochastic optimization method based on the concept of cooperation inspired by the behavior of organism, such as birds flocking or fish schooling, in search for food [1]. The idea of PSO is checked as follows. Let think about the optimization problem of maximizing the evaluation function $f : M \rightarrow M' \subset \mathbb{R}$ for variable $x \in M \subset \mathbb{R}^n$. Let there be N particles (mass point) on M dimensional space, where the position vector and velocity vector of $i (= 1, 2, 3, \dots, N)$ -th particle for m searching number are x_i^m and v_i^m . The best position for each particle in the evaluation function $f(x)$ of $x_i^1, x_i^2, \dots, x_i^m$ searching point is represented as Pb_i ($Pbest$), while the best position of $f(x)$ in the searching point for the whole particle is represented as gb ($gbest$). The particles are controlled according to Equation (1) and Equation (2).

$$v_i^{m+1} = w.v_i^m + c_1.r_1.\{Pb_i - x_i^m\} + c_2.r_2.\{gb - x_i^m\} \quad (1)$$

$$x_i^{m+1} = x_i^m + v_i^{m+1} \quad (2)$$

where w is the inertia weight; c_1 and c_2 are cognitive and social constant; r_1 and r_2 are

random numbers. There are three parts or vectors that affect the particle's movement, i.e., momentum vector, ($w.v$); cognitive vector, ($Pb - x$); and social vector, ($gb - x$). According to Equation (1) and Equation (2), the particle's movement in PSO can be illustrated in Figure 1. The next position of particle is the resultant of these three vectors.

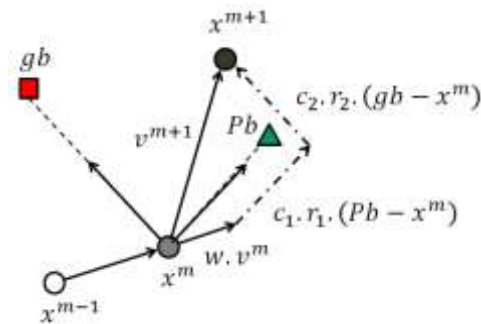


Figure 1. Particle's movement in PSO

The working mechanism of the standard type of PSO algorithm can be described in the flowchart as shown in Figure 2. First, it is an initialization of PSO. PSO has some parameters to be set, e.g., number of particles (N), maximum iteration (m_{max}), cognitive and social coefficients (c_1 and c_2), maximum and minimum value for inertia weight (w_{max} and w_{min}) and initial values for $pbest$ and $gbest$. Second, an initial population or particles as candidate solution is generated in searching area randomly. Each particle can handle a candidate solution with M -dimension. Third, it is a calculation of an inertia weight at each iteration. Fourth, the fitness value of each particle is evaluated to determine $pbest$ and $gbest$. Fifth, by using Equation (1) and (2) the position and velocity of each particle is updated. Finally, it is checking of the termination condition. If the maximum iteration is not met, return to the third step. Otherwise, if the maximum iteration is met, the process is complete and the optimum solution is the particle with $gbest$.

THE PROPOSED MPSO-RA

There are three parameters to be tuned in PSO, i.e., inertia weight, cognitive coefficient and social coefficient. Among them, the most important parameter is inertia weight because of its capability to control the balance of exploration-exploitation abilities. Cognitive

and social coefficients give minor effect, so practically they are set with a constant value, e.g., 1.0 or 1.5. Recently, research to improve PSO is being conducted intensively. Improving PSO is focused on how to adjust inertia weight in order to get a proper balance.

PSO has faster convergence and it was reported that in standard type of PSO after one-third of iteration the particles have lost mobility and tend to stagnate. Consequently, a searching ability decreases and the swarm can not find an optimum solution.

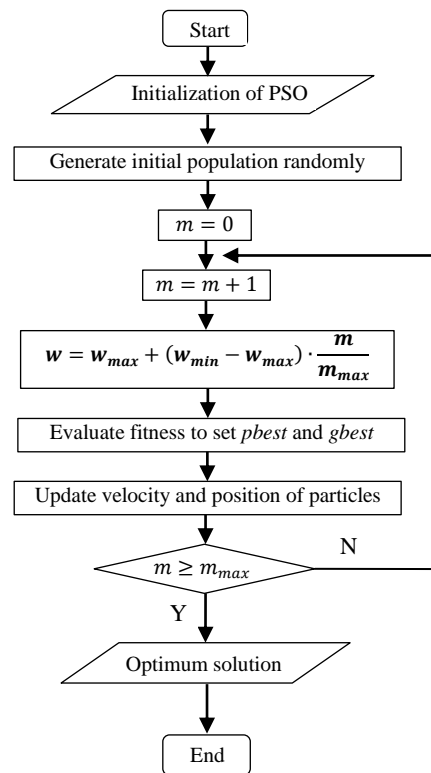


Figure 2. The flowchart of PSO

In the proposed method, random activation is added after half of iteration in order to avoid premature convergence and help the particles to jump out from local optimum. Random activation means that inertia weight will be adjusted again randomly to generate the mobility of swarm and increase exploration ability, so that they can seek a more optimum solution. Therefore, in MPSO-RA there are two strategies for adjusting inertia weight as seen in Equation (3) and Equation (4)

$$w = w_{max} + (w_{min} - w_{max}) \cdot \frac{m}{m_{max}} \cdot \begin{matrix} 1, & \text{for } m \leq \\ 0.5, & m_{max} \end{matrix} \quad (3)$$

$$w = w_{min} + (w_{max} - w_{min}) \cdot R, \text{ for } m > 0.5m_{max} \quad (4)$$

The proposed MPSO-RA algorithm is shown in Figure 3.

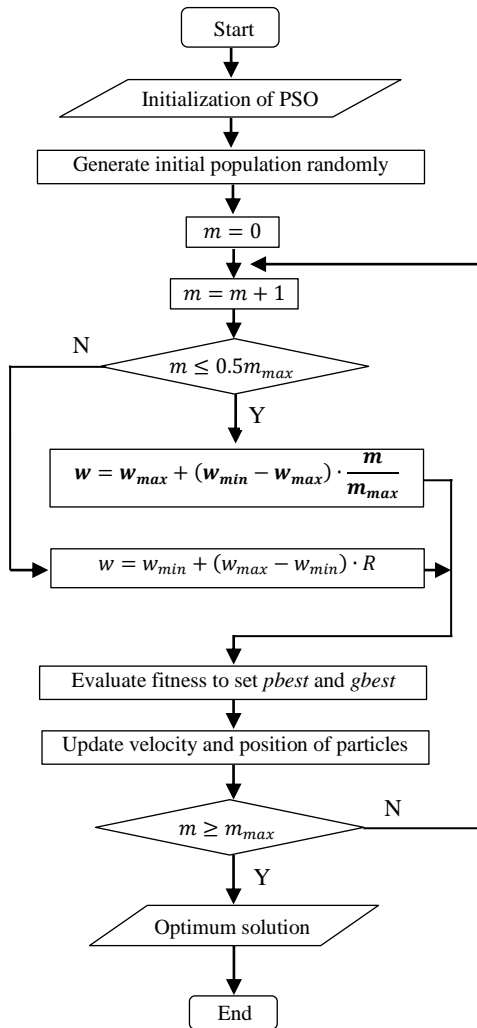


Figure 3. The flowchart of MPSO-RA

RESULTS AND DISCUSSION

This section compares the performances of the proposed MPSO-RA with the standard type of PSO or PSO-LDW. To verify and evaluate the efficiency and the effectiveness of the proposed approach we have used three widely known benchmark functions with different characteristics, i.e., Sphere function, Rosenbrock’s function and Griwank’s function,

as seen in Equation (5), Equation (6), and Equation (7).

$$f_1(x, y) = (x - 15)^2 + (y - 20)^2 \quad (5)$$

$$f_2(x, y) = 10 \cdot (x^2 - y)^2 + (1 - x)^2 \quad (6)$$

$$f_3(x, y) = 1 + \frac{x^2 + y^2}{400} - \cos(x + y) \quad (7)$$

The first function represents the optimization problem with single optimum solution (unimodal function) and strongly convex. In this case, a minimum value is zero which is achieved when $x = 15$ and $y = 20$. The second function describes the optimization problem with two optimum solution, i.e., local minimum and global minimum and it has the reputation of being a difficult optimization problem. In brief, a minimum value is zero which is achieved when $x = 1$ and $y = 1$. The third function expresses the optimization problem with multiple optimum solution, i.e., one global minimum and many local minimums. In short, a minimum value is zero which is achieved when $x = 0$ and $y = 0$.

For the purpose of comparison, all the simulation deploy the same parameter settings in both of PSO (PSO-LDW and MPSO-RA) such as the maximum number of iterations, $iter_{max} = 100$; cognitive constant, $c_1 = 1.0$; social constant, $c_2 = 1.0$; number of particles, $N = 5$, maximum inertia weight, $w_{max} = 0.9$; and minimum inertia weight, $w_{min} = 0.1$. Since PSO is a stochastic algorithm that randomly searches the best solution, so for testing we have done as much as 1000 runs.

Experimental results of MPSO-RA and PSO-LDW for Sphere function, Rosenbrock function and Griwank function averaged over 1000 runs are recorded in Table 1-3, respectively. Mean error indicates an average error over all runs. Maximum and minimum errors show the maximum and minimum value of error in the experiments. Standard deviation error designates a variation or dispersion of a set error in the experiments. Success rate represents the success of method in obtaining a predetermined minimum error within all runs. In the experiments, the predetermined minimum error for Sphere function, Rosenbrock function and Griwank function are set to be: 1×10^{-12} , 1×10^{-1} , and 1×10^{-3} , respectively. Table 1 shows the performance comparison between MPSO-RA and PSO-LDW for solving Sphere function.

Table 1. Statistical analysis for Sphere function

Type of PSO	Mean Error	Max Error	Min Error	Std Error	SR [%] (fit < 1e-12)
PSO-LDW	3.506	3502.9	9.54e-26	110.77	61
MPSO-RA	0.586	585.33	3.32e-23	18.509	89.6

From this table, by looking at mean error, maximum error, minimum error, standard deviation error and success rate, it is easy to see that MPSO-RA shows a better accuracy than the PSO-LDW. Mean error of MPSO-RA is 0.568, while mean error of PSO-LDW is 3.506. It means that due to the proposed method, the error can be reduced up to 83.29% or the solution accuracy can be increased up to 83.29%. Success rate of MPSO-RA is 89.6%, while success rate of PSO-LDW is 61%. We can say that due to MPSO-RA, success rate can be improved by 46.89%.

Table 2. Statistical analysis for Rosenbrock function

Type of PSO	Mean Error	Max Error	Min Error	Std Error	SR [%] (fit < 0.1)
PSO-LDW	21.095	734.95	3.06e-11	35.506	21.7
MPSO-RA	18.767	818.08	6.73e-13	34.566	27.9

Table 2 displays the performance comparison between MPSO-RA and PSO-LDW for solving Rosenbrock function. In here, mean error of MPSO-RA is 18.767, while mean error of PSO-LDW is 21.095. It means that due to the proposed method, the error can be reduced up to 11.04% or the solution accuracy can be increased up to 11.04%. Success rate of MPSO-RA is 27.9%, while success rate of PSO-LDW is 21.7%. In other words, due to MPSO-RA, success rate can be improved by 28.57%.

Table 3. Statistical analysis for Griwank function

Type of PSO	Mean Error	Max Error	Min Error	Std Error	SR [%] (fit < 0.001)
PSO-LDW	7.18e-2	10.949	5.33e-12	4.48e-1	28.2
MPSO-RA	4.21e-2	2.5159	3.11e-12	1.60e-1	34.8

The performance comparison between MPSO-RA and PSO-LDW for solving Griwank function is shown in Table 3. In this table, mean error of MPSO-RA is 4.21×10^{-2} , while mean error of PSO-LDW is 7.18×10^{-2} . It is observed that due to the proposed method, the error can be reduced up to 41.36% or the solution accuracy can be increased up to 41.36%. Success rate of MPSO-RA is 34.8%, while success rate of PSO-LDW is 28.2%. In other words, due to MPSO-RA, success rate can be improved by 23.40%. On the average from Table 1 - 3, it can be said that the accuracy of MPSO-RA for solving three benchmark functions is increased by 45.23%, while the success rate of MPSO-RA is improved by 32.95%. Since Rosenbrock function has a little difference between global optimum and local optimum, it is hard to find a global optimum or the best solution. Because of this, effectiveness of MPSO-RA for solving Rosenbrock function is lower than for solving other functions. However, in general the proposed MPSO-RA can improve the performance of PSO. From these tables, it is clearly obvious that the giving random activation after half of maximum contributes a good impact for increasing accuracy and success rate.

For the three functions, the convergence characteristics of the best solutions were plotted in Figure 4, 5 and 6, respectively. From the results, there are some comparisons between MPSO-RA and the PSO-LDW algorithms. In general, there is no difference significantly about convergence rate between MPSO-RA and PSO-LDW in a first half of iteration, but PSO-LDW seems easily to fall into a local minimum and may get into premature convergence.

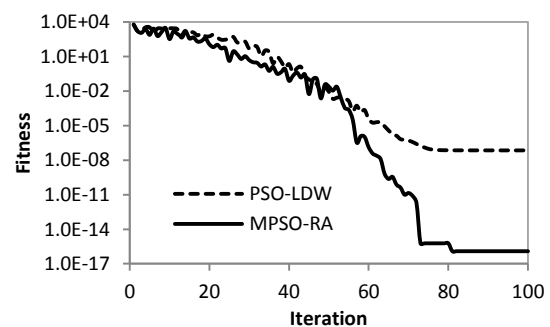


Figure 4. Fitness convergence of Sphere function

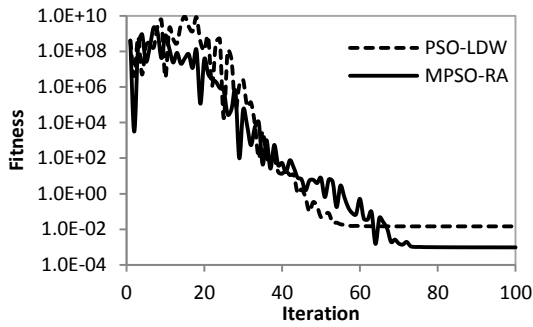


Figure 5. Fitness convergence of Rosenbrock function

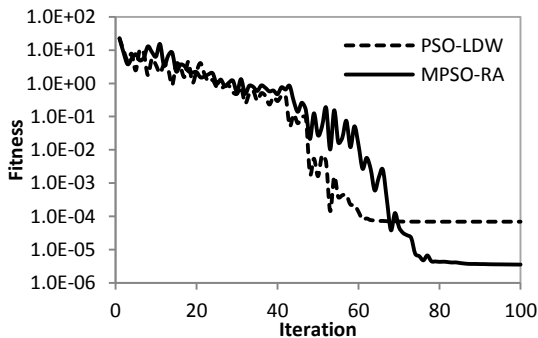


Figure 6. Fitness convergence of Griwank function

Particularly, in the cases of Sphere functions, PSO-LDW tends to converge at the points that are quite far from the destinations, the global optimum. In the contrary, though the MPSO-RA can quickly go down toward the destinations after the first half of iteration. It means that random activation gives additional velocity to the particles to jump-out from local optimum and to avoid premature convergence. The additional velocity increases exploration ability of particles to more search the best solution in wider area. In other words, the solutions searched by the MPSO-RA are much nearer to their destinations when compared with those by PSO-LDW. The original PSO-LDW algorithm rapidly becomes “stagnation”, with its searched solution no longer improved; while the MPSO-RA algorithm can still search solution gradually until the global optimum is found. That is, the MPSO-RA algorithm has a higher reliability than the original PSO-LDW algorithm to obtain the global optimum solution.

Error distribution in histogram of MPSO-RA and PSO-LDW for Sphere function, Rosenbrock function and Griwank function within 1000 runs are shown in Figure 7-12,

respectively. Because of the difference in accuracy, the discrete interval or bins for Rosenbrock function is made wider than Sphere function and Griwank function in order to get an easy viewer. From these figures, although the possibility of failure is still enough high, we can say that MPSO-RA shows a little better in accuracy because the error distribution accumulates in smaller bin.

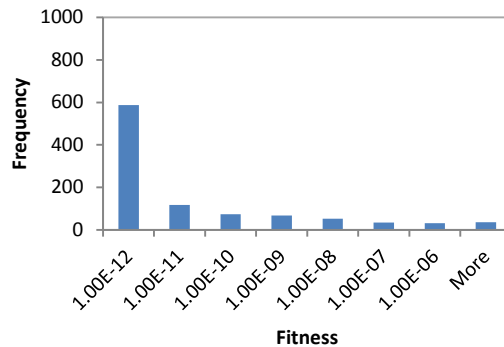


Figure 7. Histogram of Sphere function searched by PSO-LDW

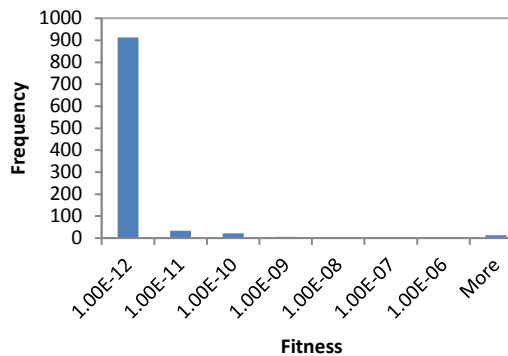


Figure 8. Histogram of Sphere function searched by MPSO-RA

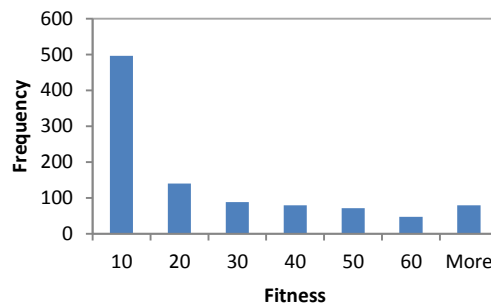


Figure 9. Histogram of Rosenbrock function searched by PSO

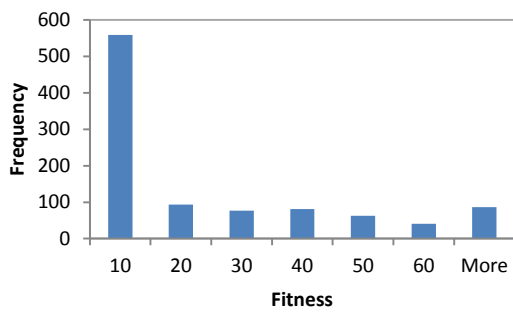


Figure 10. Histogram of Rosenbrock function searched by MPSO-RA

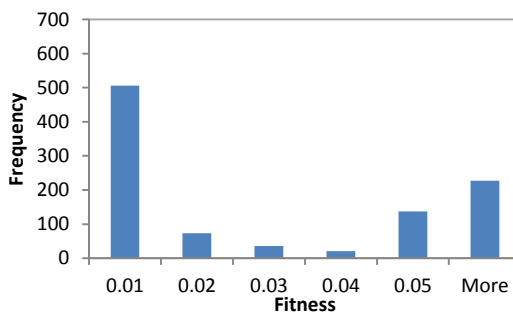


Figure 11. Histogram of Griwank function searched by PSO

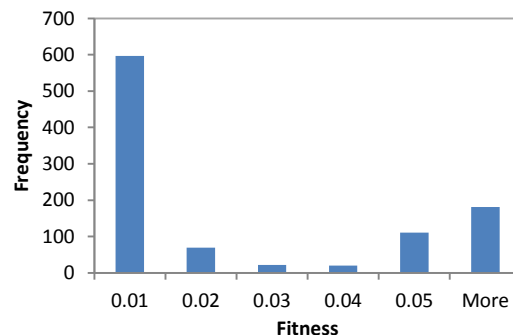


Figure 12. Histogram of Griwank function searched by MPSO-RA

CONCLUSION

This paper proposed a new modified PSO by adding random activation after half of iteration to increase exploration ability, help trapped particles for jumping-out from local optimum and avoid premature convergence. The proposed modified PSO is called Modified PSO with random activation (MPSO-RA). Compared with the standard PSO or PSO-LDW, the MPSO-RA demonstrates its superiority in solving three benchmark

functions in term of higher accuracy and higher success rate. The adding random activation has proved effectively for reducing a risk of premature convergence and increasing solution accuracy. The advantage of MPSO-RA is to use simple technique can improve the performance of PSO. However, the convergence speed is still slow as the standard PSO. In the future, we may try to modify PSO for increasing its convergence speed.

REFERENCES

- [1] J. Kennedy, R.C. Eberhart, "Particle Swarm Optimization", *Proceeding IEEE International Conference on Neural Networks*, pp. 1942-1945, 1995.
- [2] A. Kaveh, S. Malakouti, "Hybrid genetic algorithm and particle swarm optimization for the force method-based simultaneous analysis and design", *Iranian Journal of Science and Technology, Transaction B: Engineering*, vol. 34, no.B1, pp. 15-34, 2010.
- [3] J. Kennedy, R.C. Eberhart, Y. Shi. *SwarmIntelligence*. San Fransisco: Morgan KaufmanPubhisher, 2001.
- [4] A. Chander, A. Chatterjee, P. Siarry, "A new social and momentum component adaptive PSO algorithm for image segmentation", *Expert System with application*, no. 38, pp. 4998-5004, 2011.
- [5] C.L. Chen, R.M. Jan, T.Y. Lee, C.H. Chen, "A novel particle swarm optimization algorithm solution of economic dispatch with valve point loading", *Journal of Marine Science and Technology*, vol. 19, no. 1, pp. 43-51, 2011.
- [6] H. Zhu, C. Zheng, X. Hu, X. Li, "Adaptive PSO using random inertia weight and its application in UAV path planning", *Proceeding of SPIE*, vol. 7128, pp. 1-5, 2008
- [7] Alrijadjis, K. Tanaka, S. Nakashima, "Adaptive PSO Based Self-Tuning PID

Controller for Ultrasonic Motor”, *International Journal of Innovative Computing, Information and Control*, vol.9, no.10, pp. 3903-3914, 2013.

- [8] Y. Shi, R.C. Eberhart, “A modified particleswarm optimizer”, *Proceeding of IEEE International Conference on Evolutionary Computation*, Anchorage, Alaska, pp. 69-73, 1998.
- [9] J. Kennedy, R.C. Eberhart, Y. Shi. *Swarm Intelligence*. San Fransisco: Morgan Kaufman Pubhisher, 2001
- [10] A. Chatterjee and P. Siarry, “Nonlinear Inertia Weight variation for dynamic adaptation in particle swarm optimization”, *Computer and Operation Research* vol. 33, pp. 859-871, 2006.
- [11] R.C. Eberhart and Y. Shi, “Tracking and optimizing dynamic systems with particle swarms In Evolutionary Computation”, *Proceedings of the 2001 Congress on*, vol. 1, pp. 94–100. IEEE, 2002.
- [12] R.F. Malik, T.A. Rahman, S.Z.M. Hashim, and R. Ngah, “New Particle Swarm Optimizer with Sigmoid Increasing Inertia Weight”, *International Journal of Computer Science and Security (IJCSS)*, vol.1, no.2, pp. 35, 2007.