Journal of Information Hiding and Multimedia Signal Processing ©2018 ISSN 2073-4212
Ubiquitous International Volume **9**, Number **4**, July **2018**

Improved Particle Swarm Optimization Algorithm Based on Gaussian-Grid Search Method

Kun Wang[1,2], Yaqiu Liu[1], Desheng Liu[2*] and Qingwu Shi[2]

[1]College of Information and Computer Engineering
Northeast Forestry University
Harbin City, Heilongjiang Province, 150040, China
[2]College of Information and Electronic Technology
Jiamusi University
Jiamusi City, Heilongjiang Province, 154007, China
Correspondence should be addressed to Desheng Liu;liudesheng1@163.com

Received October, 2017; revised May, 2018

Abstract. *As we all know, Particle Swarm Optimization(PSO) algorithm produces premature convergence individual. But its local optimization ability is poor due to the loss of species diversity in the search space. So we propose an improved PSO algorithm based on Gaussian grid search algorithm. This new algorithm initializes the particle swarm based on Gaussian distribution and improves the particle velocity updating formula. Then the parameters are optimized by the grid search method. We give detailed procedures of improved PSO algorithm based on Gaussian grid search. Finally, we use two examples to show the new algorithm's better performance index and application prospect.*

Keywords: PSO algorithm, Gaussian-grid search, Process optimization, Gaussian distribution

1. **Introduction.** PSO algorithm[1,2] is derived from feeding behavior simulation of birds or fish. It is a new Global optimization evolutionary algorithm which has been used widely in system identification and neural network training area. But PSO has the premature convergence problem especially in the more complex multimodal search problems. Some researchers had studied PSO. Delice Y [3] proposed a new modified particle swarm optimization algorithm with negative knowledge. Its aim was to solve the mixed-model two-sided assembly line balancing problem. The new algorithm included new procedures such as generation procedure which was based on combined selection mechanism and decoding procedure. Zhang [4] proposed improved adaptive particle swarm optimization involving many conflicting objectives and constraints. It applied an adaptive dynamic parameter control mechanism into parameter settings. Azadeh [5] represented a new solution approach based on the particle swarm optimization (PSO) algorithm. It used group-based operators, in the body of the updating equations analogous to those of the classical PSO equations. But they still have the low speed. Also there are some improved PSO [6-9] algorithms.

So this paper proposes an improved PSO algorithm based on Gaussian search. This new algorithm changes traditional particle swarm population initialization strategy. We make Gaussian distribution for particle swarm taking controlling quantity initialization value as the center and increase the probability of optimization in a small area. We also change the speed updating method of traditional particle swarm and put controlling quantity initialization value into speed updating formula. This new method will play an important role in control systems.

This paper is organized as follows. In section 2, we introduce the PSO algorithm. Section 3 gives the improved PSO algorithm based on Gaussian-grid search. In section 4, we conduct experiments through two numerical examples. There is a conclusion in section 5.

2. **PSO algorithm.** Particle swarm optimization(PSO) algorithm is a random optimization algorithm based on population [10,11]. We set the particle population size as L. Particle i can be expressed by: $pop_i = (u_i, F_i), 0 < i < L$, where $u_i = (u_{i1}, u_{i2}, \cdots, u_{im})$ is the position vector of i. F_i is current adaptive value vector. Let velocity vector of i be $v_i = (v_{i1}, v_{i2}, \cdots, v_{im})$. The initial position of the population is: $u_i = (u_1^{min} + r(u_1^{max} - u_1^{min}), u_2^{min} + r(u_2^{max} - u_2^{min}), \cdots, u_m^{min} + r(u_m^{max} - u_m^{min}))$, r is random value of [0,1]. At the $t-th$ iteration, the adjustment formula of speed, position and inertia weight are:

$$v_{im}^{t+1} = \omega v_{im}^{t} + c_1 r_1 (F_{im}^{b} - u_{im}^{t}) + c_2 r_2 (F_{m}^{g} - u_{im}^{t}). \tag{1}$$

$$u_{im}^{t+1} = u_{im}^{t} + v_{im}^{t+1}. \tag{2}$$

$$\omega = 1 - I \cdot \frac{\omega_0}{I_{max}}. \tag{3}$$

Where c_1 and c_2 are accelerating factor. r_1 and r_2 is random value of [0,1]. ω_0 is initial weight. ω is inertia weight. $F_i^b = (F_{i1}^b, F_{i2}^b, \cdots, F_{im}^b)$ represents the historical best position of i. $F^g = (F_1^g, F_2^g, \cdots, F_m^g)$ is the best position found by the whole particles in particle swarm.

However, when the traditional particle swarm optimization initializes the particle swarm, each particle is randomly dispersed to variation range of controlling the quantity, then it will execute a search strategy. Because there is no relation between every controlling quantity solution with that in previous time which limits the change of controlling quantity. It may result in a big rate of change.

3. **Improved PSO based on Gaussian-grid search.** This paper presents an improved PSO based on Gaussian search to solve the control variable. The new method not only limits controlling quantity rate of change, but ensures the control effect target. It must strengthen the local search ability of particle swarm. We need to find the optimal solution near initial value. So we must take the previous controlling quantity into consideration and take it as the Gaussian center of particle swarm. Particle swarm scatters in the search area in the Gaussian distribution form. It starts to search the optimal solution from the particle swarm center. Among the iteration process, it will modify updating way of particle swarm velocity. Finally, the objective function meets the set threshold value or reaches the maximum number of iterations. This new scheme designs a diffusion operator and puts it into PSO algorithm. We set the initial position of the particle: $u'(k) = (u_1(k), u_2(k), \cdots, u_n(k))$. After diffusional operation, the position of particle is $u'(k+1) = (u_1(k+1), u_2(k+1), \cdots, u_n(k+1))$.

So the diffusion operator can be expressed by:

$$u'(k+1) = u'(k) + N_i(0, \delta_i'). \tag{4}$$

Where $i = 1, 2, \cdots, n$, δ_i' is standard deviation of Gaussian distribution. $N_i(0, \delta_i')$ is a random value following Gaussian distribution. Its mean value is zero and standard deviation is δ_i'. The particle's probability of one local search near initial value is very big.

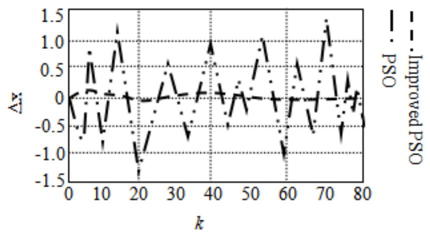

FIGURE 1. x varies with k.

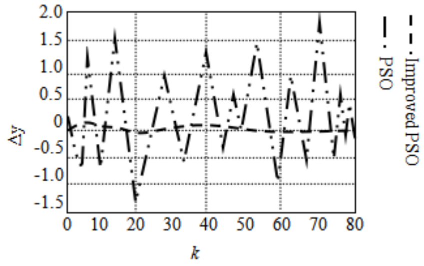

FIGURE 2. y varies with k.

We use equation 4 to initialize the particle swarm which is conducive to get the optimal solution on a small scale.

In order to enhance the search ability of PSO, this paper improves speed updating way of particle. In the basic speed updating way, the local optimum and globe optimum of particle have a big effect on state of particle. In the process of iteration, particle updates itself continuously by tracking the two extreme values. So we can get updated speed and updated position by combining initial value $u(k)$ with speed updating formula.

$$v_{im}^{t+1} = \omega v_{im}^{t} + c_1 r_1 (F_{im}^{b} - u_{im}^{t}) + c_2 r_2 (F_{m}^{g} - u_{im}^{t}) + c_3 r_3 (u(k) - u_{im}^{t}). \tag{5}$$

$$u(k') = u(k) + v_{im}^{t+1}. \tag{6}$$

This designing enhances intervention of initial value for particle traveling route. That can make particle search optimal solution and also take account of minimization of the search step. The detailed improved PSO algorithm processes are as follows:

1. According to the controlling quantity $u(k)$ of previous moment, we take it as the Gaussian center of particle swarm. Using formula (5,6) generates new initial particle swarm $u'(k+1)$.
2. Calculate adaptive value respectively by generated particle swarm. We can obtain the global optimum value F_m^g of particle swarm and record particle with the optimal adaptive value.
3. Judging the F_m^g meets the iteration stopping condition or not. If YES, then quit the optimization process. We can take the optimum value as the controlling quantity $u(k+1)$ of current time.
4. According to (5,6), we update the speed and position of each particle.

4. **Experiments and analysis.** In order to verify the effectiveness of this paper's improved algorithm. We conduct two experiment examples.

Experiment 1.

We use the multi-modal function as simulation model [12] and make a comparison with traditional PSO.

$$f(x, y) = cos(2\pi x) \cdot cos(2\pi y) \cdot e^{-0.1(x^2+y^2)}. \tag{7}$$

Where y is predictive output. $x, y \in [-1, 1]$. We set $\delta^2 = 0.05$. In this new algorithm, let the variate be: $\delta'_x = \delta'_y = 0.1$. $P = 3$ is control horizon. Maximum number of iteration is 200. Threshold $F_{ok} = 10^{-7}$. Control objective is sine function $y_r(k) = 0.8sin(k\pi/20)$. k is simulation step. Whole simulation steps are 80. We can get the results as figure 1 and figure 2.

From this two figures, we can know that x and y of traditional PSO have a big changes in search range. It is bad for the controller. However, x and y of improved PSO have a small changes in a small range. Experiment results show that PSO based on Gaussian search can ensure controlling effect and restrain the rate of change of x and y. In order to compare the effect of PSO and improved PSO, we give the average variation rate of control $\Delta u'$ as comparison metric.

$$\Delta u' = \frac{\sum_{k=1}^{T-1} \left|\frac{u(k+1)-u(k)}{u(k)}\right|}{T-1}. \tag{8}$$

Where T is total number of simulation. u denotes x or y. We make ten contrast experiments and get average control rate of change as table1. Table1 shows that we use PSO based on Gaussian search and get the lower changing rate of x and y. Improved PSO has the superior performance than PSO.

Table 1. Control rate of change with different algorithms in Experiment 1.

Test Number	PSOΔx/%	Improved PSOΔx/%	PSOΔy/%	Improved PSOΔy/%
1	3.58	0.65	6.78	0.16
2	5.43	0.48	3.18	0.74
3	3.15	0.20	2.90	0.73
4	3.19	0.63	6.24	0.45
5	4.39	0.85	2.96	0.44
6	3.65	0.22	2.66	0.33
7	2.63	0.29	2.84	0.82
8	3.63	0.14	3.71	0.46
9	3.17	0.71	2.12	0.45
10	2.57	0.25	2.43	0.81
Average	3.54	0.44	3.58	0.53

Experiment 2.

We take an Input and Output System to conduct experiment under MATLAB platform. Input variable includes μ_1 and μ_2. Input optimizing function is y_1. Output optimizing function is y_2. It tests transfer function of grinding system by step function response. The model is as (9).

$$(y_1 \quad y_2)^T = \left(-\frac{0.425e^{-1.52s}}{11.7s+1} \quad \frac{1.062e^{-2.26s}}{2.5s+1}\right)^T (u_1 \quad u_2)^T. \tag{9}$$

The sampling time selects 0.5s. To simulate the scene, we add 0.01 Gaussian noise of superposition variance. We set $u_1 \in [0,1]$, $u_2 \in [0,1.2]$. $\delta^2 = 50$, $\delta'_x = \delta'_y = 0.1$. $P = 3$. Maximum number of iterations is 500. Threshold $F_{ok} = 10^{-6}$. Controlling effect of system with improved PSO algorithm are as figures 3-10.

Figures 3,4 show that when the working point changes large, it adopts this paper's method which can track the change of the set value without overshoot and has a fast enough response speed. From figures 5,6 we can know that when we change the setting value greatly, controlled quantity will not change much. Because the PSO algorithm is random search algorithm, the output of controlling system has the smallest static error. Figures 7,8 show that the closed loop of ore grinding granularity will return to normal after affected by disturbance with the improved PSO algorithm. Its robustness is illustrated very well. Figures 9,10 reflect the speed changing. Speed of improved PSO is superior to PSO. It will become convergence at a short time. We also get control rate of change with different algorithms as shown in TABLE 2. It obtains that the search strategy of new algorithm is better than traditional PSO algorithm.

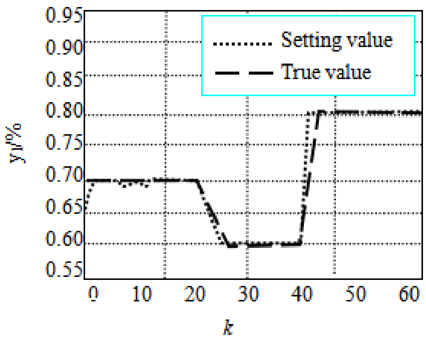

FIGURE 3. Input function error.

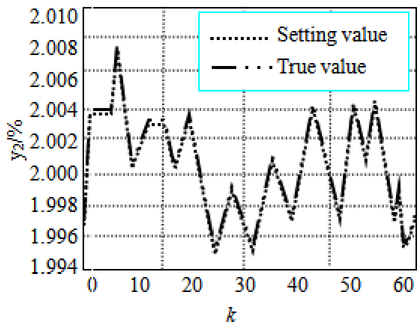

FIGURE 4. Output function error.

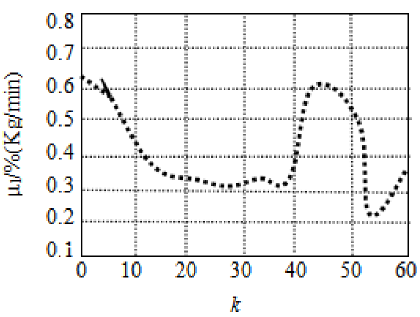

FIGURE 5. Change of variable μ_1.

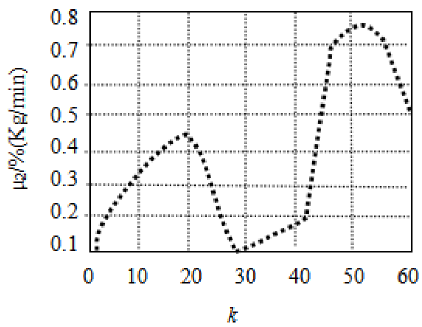

FIGURE 6. Change of variable μ_2.

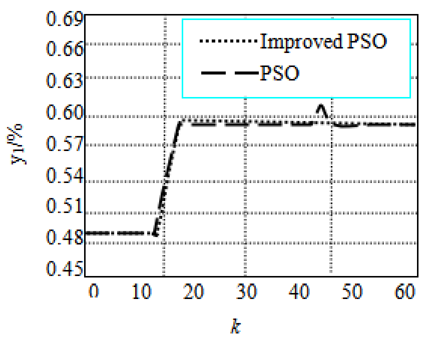

FIGURE 7. Comparison with PSO and new method for y_1.

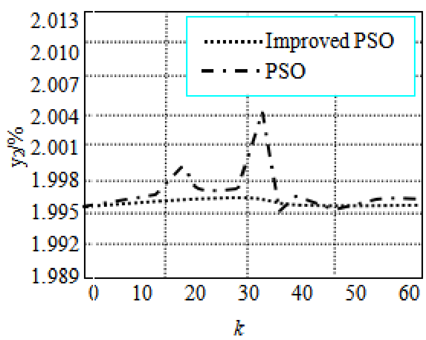

FIGURE 8. Comparison with PSO and new method for y_2.

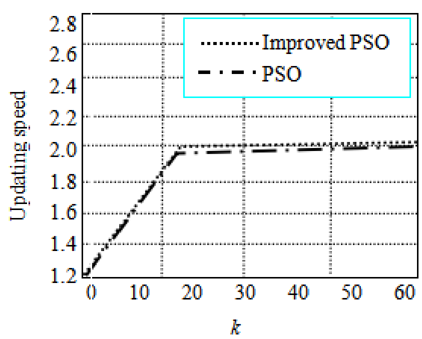

FIGURE 9. Updating speed comparison between Improved PSO and PSO.

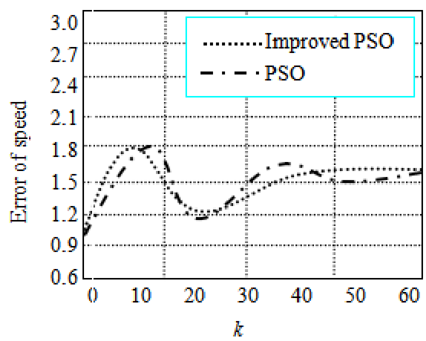

FIGURE 10. Error speed comparison between Improved PSO and PSO.

Table 2. Control rate of change with different algorithms in Experiment 3.

Test Number	PSOμ_1/%	Improved PSOμ_1/%	PSOμ_2/%	Improved PSOμ_2/%
1	13.02	0.56	6.45	0.13
2	12.64	0.42	3.05	0.47
3	11.21	0.15	2.68	0.64
4	11.69	0.59	6.17	0.39
5	9.43	0.81	2.92	0.41
6	8.77	0.13	2.53	0.31
7	9.48	0.23	2.81	0.79
8	10.78	0.11	3.65	0.42
9	12.56	0.66	2.08	0.42
10	12.74	0.21	2.41	0.78
Average	10.73	0.41	3.52	0.51

5. **Conclusions.** This paper proposes an improved PSO algorithm based on Gaussian-grid search. It not only can control range-ability of variable, but also can ensure the controlling effect. This new algorithm can be used in nonlinear predictive controller. The simulation experiments results show that it can get satisfied control effect in controlling system. The controller has quick response speed and higher robustness. So this method will play an important role in practical engineering applications. The improved PSO is very effective for restraining controlled quantity. In the future, we will continue to study advanced PSO algorithms to improve controlling systems.

Acknowledgement. This work is supported by the Basic research project of Heilongjiang Provincial Department of Education (NO:2016-KYYWF-0547); Jiamusi University science and technology innovation team construction project (NO:CXTDPY-2016-03); Heilongjiang Province Science Fund for returnees (NO:LC2017027); School-level Project of Jiamusi University (JMSUJCMS2016-005).

REFERENCES

[1] T. Liu, S. Yin. An improved particle swarm optimization algorithm used for BP neural network and multimedia course-ware evaluation, *Journal of Multimedia Tools and Applications*, vol. 76, no. 9, pp. 11961–11974, 2017.

[2] S. Yin, J. Liu, L. Teng, An improved ant colony algorithm used for unscented kalman filter, *ICIC express letters. Part B, Applications: an international journal of research and surveys*, vol. 7, no. 11, pp. 2411-2417.

[3] Y. Delice, K. E. Aydoan, Uzcan, S. Mehmet. A modified particle swarm optimization algorithm to mixed-model two-sided assembly line balancing, *Journal of Journal of Intelligent Manufacturing*, pp. 1-14, 2014.

[4] Z. Zhang, Y. Jiang, S. Zhang, .S Geng, H. Wang, G. Q. Sang. An adaptive particle swarm optimization algorithm for reservoir operation optimization, *Journal of Applied Soft Computing,* , vol. 18, pp. 167-177, 2014.

[5] H. A. Kashan, B. Karimi, A Noktehdan. A novel discrete particle swarm optimization algorithm for the manufacturing cell formation problem, *Journal of The International Journal of Advanced Manufacturing Technology*, vol. 73, no. 9-12, pp. 1543-1556, 2014.

[6] L. Meng, S. Yin, An improved Mamdani Fuzzy Neural Networks Based on PSO Algorithm and New Parameter Optimization. *TELKOMNIKA Indonesian Journal of Electrical Engineering* , vol 17, no 1, pp:201-206, 2016.

[7] J. Liu, S. Yin, L. Teng. An Improved Multiple Extended Target Tracking Algorithm Based on Variational Bayesian Cardinality Equilibrium Multi-Objective Bernoulli Filtering, *ICIC Express Letters Part B: Applications,* vol. 7, no. 01,pp: 209-214, 2016.

[8] H. Wang, H. Sun, C. Li, S. Rahnamayan, J. Pan. Diversity enhanced particle swarm optimization with neighborhood search, *Journal of Information Sciences*, vol. 223, no. 2, pp. 119-135, 2013.

[9] K. Mcginnity ,R. Varbanov , E. Chicken, Cross-validated wavelet block thresholding for non-Gaussian errors, *Computational Statistics & Data Analysis,* vol. 106, pp. 127-137, 2017.

[10] S. Chen, G. Manalu, J. S. Pan, H. C. Liu. Fuzzy Forecasting Based on Two-Factors Second-Order Fuzzy-Trend Logical Relationship Groups and Particle Swarm Optimization Techniques., *Journal of IEEE Transactions on Cybernetics*, vol. 43, no. 3, pp. 1102-1117, 2013.

[11] C. Sun, J. Zeng, J. S. Pan, S. Xue, Y. Jin. A new fitness estimation strategy for particle swarm optimization, *Journal of Information Sciences*, vol. 221, no. 2, pp. 355-370, 2013.

[12] Q. Tang, Y. Shen, C. Hu, J. Zeng, W. Gong. Swarm Intelligence: Based Cooperation Optimization of Multi-Modal Functions, *Journal of Cognitive Computation*, vol. 5, no. 1, pp. pp 48-55,2013.