Jurnal Teknologi

Full paper

An Improved Global Particle Swarm Optimization for Faster Optimization Process

Mohamad Fadzli Haniff[a], Hazlina Selamat[a*], Salinda Buyamin[b]

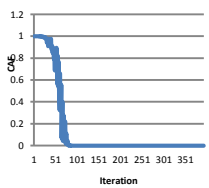
[a]*Centre for Artificial Intelligence & Robotics (CAIRO), Universiti Teknologi Malaysia, Jalan Semarak, Kuala Lumpur, Malaysia*
[b]*Control & Mechatronic Engineering Dpt. Faculty of Electrical Engineering, Universiti Teknologi Malaysia, UTM Johor Bahru, Johor, Malaysia*

*Corresponding author: hazlina@fke.utm.my

Article history

Received :15 June 2014
Received in revised form :
15 September 2014
Accepted :15 October 2014

Graphical abstract



Abstract

An efficient Global Particle Swarm Optimization (GPSO) is proposed in order to overcome the concern of trapping in the local optimal point especially in high dimensional while using ordinary Particle Swarm Optimization (PSO). GPSO is able to bring all the particles to be closely clumped together faster than PSO. In this paper, an improved GPSO is proposed in order to get a closely clumped particles group faster than using GPSO. The original GPSO is improved by taking into account the global best fitness error and particle fitness clumping size of every iteration. The improved GPSO is simulated by using several two dimension mathematical function and benchmarked with the original GPSO. The improved GPSO is shown to be able to obtain closely clumped particles much more faster than the original GPSO up to 62%. The performances are also evaluated by comparing the standard deviation, average, best particle and worst particles obtained through a 50 independent runs. In term of the four factors mentioned, the improved GPSO performance is shown to be as good of the original GPSO.

Keywords: Particle swarm optimization; global particle swarm optimization; optimization

© 2015 Penerbit UTM Press. All rights reserved.

1.0 INTRODUCTION

PSO is a popular optimization algorithm, which is proven to be robust and possess a well global exploration ability. It has been improved into many versions to be implemented in various kind of system such as Guided PSO for facial emotion detection [1,2], Gaussian PSO for bandwidth parameter determination [3] and genetic algorithm PSO for fault diagnosis [4]. PSO has tendency of getting trapped in local minima and slow convergence [5], most of the improved PSO are proposed due to these concerns.

In [6], an efficient GPSO algorithm is proposed in order to get an optimized point without getting trapped in the local optimal point as happened with ordinary PSO. A fast optimization method is essential in systems such as system scheduling [7,8] and human motion tracking [9,10]. The GPSO is said to be able to reach the optimal results in several iterations, which is faster than PSO. In [6], GPSO is compared to the ordinary PSO, iteration PSO and evolutionary PSO algorithms and has shown to have better performance.

In this paper, an improved GPSO is proposed in order to enhance the speed of optimization. The improved GPSO is taking into the account the global best fitness error and the particle fitness clumping size of every iteration. In Section 2, the modification done on the original PSO is discussed. In Section 3, the simulation results and discussions are presented. Lastly, Section 5 concludes this paper.

2.0 METHODOLOGY

2.1 An Improved Global Particle Swarm Optimization Algorithm

In ordinary PSO, the velocity algorithm is the most important part. Most of the modification of PSO revolves around this algorithm. The ordinary PSO velocity algorithm includes information such as local best position until k-th iteration for i-th particle (P_{best-i}^{k}) and global best position until k-th iteration (G_{best}^{k}) as shown below:

$$v_i^{k+1} = wv_i^k + c_1 r_{1i}\left(P_{best-i}^k - x_i^k\right) + c_2 r_{2i}\left(G_{best}^k - x_i^k\right) \quad (1)$$

where $k$is the current iteration, $w$is the inertia weight, $v_i$is the velocity of i-th particle, $c_1$and c_2 are the acceleration coefficients for cognitive and social components, lastly $r_1$and r_2 are random numbers between 0 and 1.

In GPSO, a fourth term is added into the velocity algorithm as below:

$$v_i^{k+1} = wv_i^{k+1} + c_1 r_{1i}\left(P_{best-i}^k - x_i^k\right) + c_2 r_{2i}\left(G_{best}^k - x_i^k\right) + c_3 r_{3i}\left(E_{best}^k - x_i^k\right) \quad (2)$$

where $c_3 r_{3i}\left(E_{best}^k - x_i^k\right)$ is named as the Improvement Factor (IF). c_3 is acceleration coefficent and r_3 is another random

numbers from 0 to 1. E_{best} is the random value selected from G_{best} for each iteration.

In the proposed improved GPSO, the forth term, IF is modified with the addition of an exponential equation, which taking into consideration the particle fitness clumping size and global best fitness error until k-th iteration as below:

$$v_i^{k+1} = wv_i^k + c_1 r_{1i}\left(P_{best-i}^k - x_i^k\right) + c_2 r_{2i}\left(G_{best}^k - x_i^k\right) + e^{\frac{-k}{s_k+e_k}} IF \quad (3)$$

The added exponential equation $e^{\frac{-k}{s_k+e_k}}$ is named correction acceleration factor (CAF) where s_k is the particle fitness clumping size of k-th iteration, which calculated using algorithm below:

$$s_k = abs\left(F_{max}^k - F_{min}^k\right) \quad (4)$$

where F_{max}^k is the maximum value of current iteration particle fitness and F_{min}^k is the minimum value of current iteration particle fitness. Meanwhile $e_k$is the global best fitness error of k-th iteration determined with the difference between the current iteration global best fitness (G_k) and the expected fitness (E). E is determined by the desire of the programmer.

$$e_k = abs(G_k - E) \quad (5)$$

The CAF is formed in order to control the acceleration of the particles depending on the difference between the local minimum and maximum fitness of current iteration an also the difference between global best fitness with the expected fitness that is to be determined.

During the start of optimization process, the value of Equations (4) and (5) are expected to be great, which means CAF value will also be great (not greater than 1). At this moment, the particles will accelerate following the determined IF value. As the iteration goes on, the value of Equations (4) and (5) will get smaller, causing the CAF value to be decrease, eventually to zero, render the IF to be null. At this time, particles start to slow down and eventually following the ordinary PSO velocity algorithm.

With CAF, it is expected that the number of iteration required for GPSO particles to be closely clumped will decrease. This makes the improved GPSO to be the latest fast speed optimizer in PSO family. Figure 1 shows the flow chart of the improved GPSO algorithm.

3.0 RESULTS AND DISCUSSIONS

Two simulations had been done on six benchmark functions based on [6] as shown in Table 1 in order to evaluate the performance of the original GSPO and the improved (IGPSO). The first simulation is done in order to determine how many iterations are required for both GPSO and IGPSO to fulfill the requirement based on Equation (6), where $\mathcal{E}$ is the acceptable value determined by the desire of the programmer.

$$s_k < \mathcal{E}. \quad (6)$$

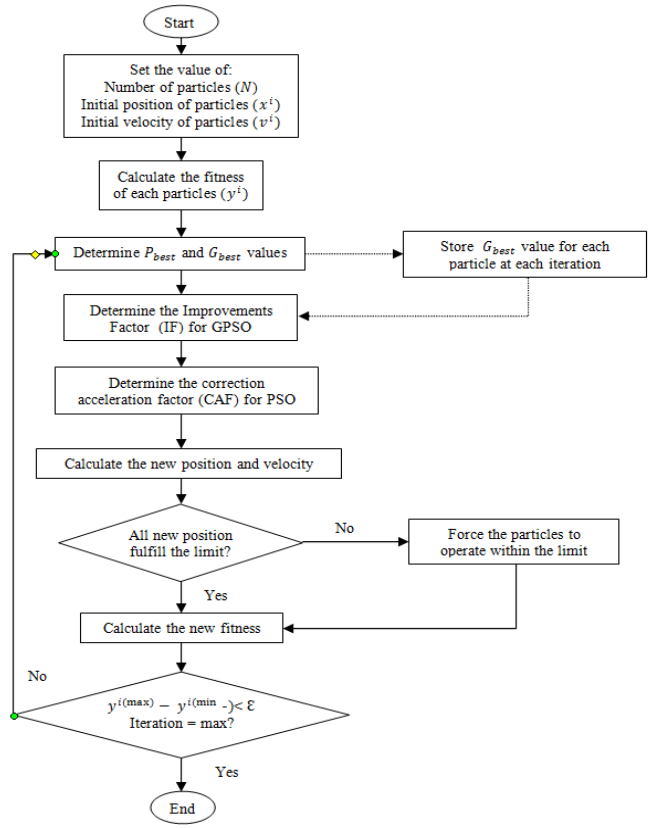


Figure 1 The process of improved GPSO Algorithm

The number of particles used are 200, maximum iteration is 1000 and $c_1 = c_2 = c_3 = 1$. The simulation is done so that the simulation will stop only when both of GPSO and IGPSO fulfilled the requirement. Fitness average is calculated from the final iteration particles. Table 2 shows the first simulation result.

Table 1 Bench mark test functions

Test Function n = 2
$f_1(x) = \sum_{i=1}^{n} x_i^2$
$f_2(x) = \sum_{i=1}^{n} (10^6)^{(i-1)/(n-1)} x_i^2$
$f_3(x) = \sum_{i=1}^{n} \lvert x_i \rvert^{(i+1)}$
$f_4(x) = \sum_{i=1}^{n} (x_i + 0.5)^2$
$f_5(x) = \frac{1}{n} \sum_{i=1}^{n} (x_i^4 - 16x_i^2 + 5x_i)$
$f_6(x) = \sum_{i=1}^{n} \lvert x_i \sin(x_i) + 0.1x_i \rvert$

From Table 2, it is shown that IGPSO is able to fulfill the clumping size requirement more faster than GPSO at up to 62% rate. If IGPSO is to be simulated as long as GPSO is, the fitness average for final iteration for IGPSO is much more smaller than GPSO, which means the particles of IGPSO is much more closer when compared to GPSO. Even when requirement is not fulfilled in 1000 iterations, IGPSO is still producing smaller fitness average.

Table 2 Iteration and average fitness of single simulation results

f	ε	Particle position limitation	Iteration required		Average Fitness after simulation stop	
			GPSO	*IGPSO*	*GPSO*	*IGPSO*
1	0.5	-100 to 100	392	146	0.0187	2.08e-6
2	0.05	-10 to 10	212	93	7.03e-4	1.60e-4
3	0.05	-10 to 10	158	87	0.0015	4.20e-5
4	0.5	-100 to 100	269	158	0.0156	4.54e-6
5	0.025	-5 to 5	>1000	>1000	-0.7850	-0.0228
6	0.05	-10 to 10	>1000	>1000	0.2199	0.2189

Figure 2 shows the sample of the CAF from Function 1. It can be seen that CAF never exceed the value of 1, so no over stepping of particles may happen. Over stepping of particles happens when particles moves to fast or too far. As the iteration increases, CAF decreases to zero, as expected. This figure shows that IF will only be applied at the beginning of the iteration, as the iteration goes on, the effective velocity value is obtained based on the original PSO algorithm.

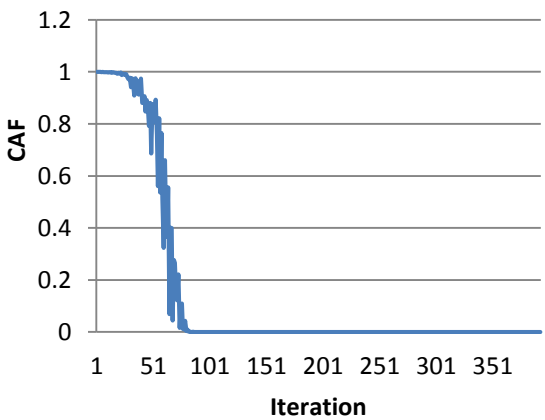


Figure 2 CAF sample from Function 1 optimization

Second simulation is done by executing 50 different runs of 50 iteration simulation. In each runs, the global best fitness particles is chosen. The number of particles used are 200, maximum iteration is 1000 and $c_1 = c_2 = c_3 = 1$. At the end of the 50 runs, the worst fitness, best fitness, fitness average and standard deviation are determined from the 50 collected global best fitness particles. The result of the second simulation is shown in Table 3.

Table 3 shows that the performance of IGPSO is as good as GPSO in term of the worst fitness, best fitness, fitness average and the standard deviation. Although the worst and best fitness of IGPSO is not consistently lower than GPSO, the IGPSO fitness average and standard deviation is always lower than GPSO. This shows that IGPSO particles performance as a group is much more better than GPSO.

Table 3 Best fitness, worst fitness, average fitness and standard deviation of 50 different simulations

f	Method	Worst	Best	Average	Std Dev
1	GPSO	0.0718	3.113e-5	0.0220	0.0219
	IGPSO	0.0964	1.400e-4	0.0157	0.0188
2	GPSO	0.0029	5.085e-6	4.994e-4	5.722e-4
	IGPSO	7.058e-4	2.743e-8	8.079e-5	1.317e-4
3	GPSO	0.0013	8.400e-7	1.482e-4	2.384e-4
	IGPSO	5.228e-4	1.542e-7	3.703e-5	8.343e-5
4	GPSO	0.1122	5.093e-4	0.0221	0.0228
	IGPSO	0.0909	5.504e-4	0.0200	0.0178
5	GPSO	-0.0047	-6.409e-6	-3.192e-4	0.0014
	IGPSO	-0.0062	-5.605e-5	-1.592e-4	0.0015
6	GPSO	0.1355	3.025e-4	0.0054	0.0192
	IGPSO	0.0157	7.492e-5	0.0011	0.0023

4.0 CONCLUSIONS

From the simulation results and discussions, it can be concluded that the IGPSO is a faster version of GPSO, which able to reduce the optimization process up to 62% faster. The proposed correction acceleration factor (CAF) is proven to function as acceleration control mechanism for the improvement factor (IF). With the CAF as the global best fitness approaching the expected fitness, which in this case is 0. As the particles are getting close to each other, the particles will eventually slow down and starts to move based on original PSO velocity algorithm. By controlling the IF with CAF, overstepping of particles can be avoided. Overstepping of particles may cause more iterations are required in order to fulfill the clumping requirement. As a future work, IGPSO is to be implemented on high dimension or complex low dimension functions such as predicted mean vote (PMV) equations.

Acknowledgement

This project is supported by the Ministry of Higher Education and Universiti Teknologi Malaysia under the Research University Grant 05H33.

References

[1] Ghandi, B. M., Nagarajan, R. and Desa, H. 2010. Real-time System for Facial Emotion Detection Using GPSO Algorithm. *Industrial Electronics & Applications (ISIEA), 2010 IEEE Symposium*. 40–45.

[2] Ghandi, B. M., Nagarajan, R. and Desa, H. 2010. Facial Emotion Detection Using GPSO and Lucas-Kanade Algorithms. *Computer and Communication Engineering (ICCCE), 2010 International Conference* 1–6.

[3] Liang, H. L. and Shen, X. M. 2011. Applying Particle Swarm Optimization to Determine the Bandwidth Parameter in Probability Density Estimation. *Machine Learning and Cybernetics (ICMLC), 2011 International Conference.* 3: 1362–1367.

[4] Qian Y. L., Zhang, H.; Peng, D. G., Huang, C. H. 2012. Fault Diagnosis for Generator Unit Based on RBF Neural Network Optimized by GA-PSO. *Natural Computation (ICNC), 2012 Eighth International Conference* . 233–236.

[5] Li, X., Liang, X., Ercan, M. F. and Zhou Y. 2009. A New Hybrid Algorithm Based on Collaborative Line Search and Particle Swarm Optimization. *Autonomous Robots and Agents, ICARA 2009. 4th International Conference*. 486, 489.

[6] Jamian, J. J., Abdullah, M. N., Mokhlis, H., Mustafa M. W. and Bakar , A. H. A. 2014. Global Particle Swarm Optimization for High Dimension Numerical Functions Analysis. *Journal of Applied Mathematics*. http://dx.doi.org/10.1155/2014/329193.

[7] Zhan, J. P., Yin, Y. J., Guo, C. X. and Wu, Q. H. 2011. Integrated maintenance Scheduling of Generators and Transmission Lines Based on Fast Group Searching Optimizer. *Power and Energy Society General* Meeting. 1–6.

[8] Minzu, V. and Beldiman, L. 2003. A Parallel Hybrid Metaheuristic for the Single Machine Scheduling Problem. *Assembly and Task Planning, 2003. Proceedings of the IEEE International Symposium on*. 134–139.

[9] Lee, J. K. and Park, E..J. 2008. A Fast Gauss-Newton Optimizer for Estimating Human Body Orientation. *Engineering in Medicine and Biology Society. 30th Annual International Conference of the IEEE*. 1679–1682.

[10] Lee, J. K. and Park, E. J. 2009. A Fast Quaternion-based Orientation Optimizer via Virtual Rotation for Human Motion Tracking. *Biomedical Engineering, IEEE Transactions on*. 56(5): 1574–1582.