An adaptive fuzzy weight PSO algorithm

Changxin Liu, Chunjuan Ouyang
College of Electronics and Information Engineering
Jinggangshan University
Ji'an, Jiangxi, China
{jalcx, oycj001}@163.com
Ping Zhu, Weidong Tang
{zhuping, tangwd}@jgsu.edu.cn

Chunjuan Ouyang
College of Information and Engineering
Shenzhen University
Shenzhen, Guangdong, China
oycj001@163.com

***Abstract*—In this paper ,we propose a novel adaptive fuzzy weight parameter PSO Algorithm（FPSO）. In the improved algorithm, the inertia weight reserves its decreasing property after fuzzy treatment ，and the position is controlled by fuzzy parameter. Simulations have been done to illustrate that the improved algorithm can regulate global search and local search, and has better search accuracy than the basic PSO and the linear decreasing inertia weight particle swarm optimization (WPSO).**

Keywords- fuzzy weight; global optimum; PSO

I. INTRODUCTION

PSO (Particle Swarm Optimization, PSO) is proposed by Eberhart and Kennedy[1] in 1995, which resembles a school of flying birds. Shi et al[2] introduce the inertia weight ω to control the convergence and exploration, and get the current standard PSO algorithm. PSO is simple, but it is easy to fall into local minimum. In recent years, various improved algorithm are proposed, mainly to increase the species diversity, the inertia coefficient optimum, the individual, and the global optimum ability[3-7]. There are some uncertain problems in the group activity, and the fuzzy theory is a good mathematical tools to solve the uncertain problem. In this paper, we will utilize the fuzzy theory to propose a novel fuzzy adaptive particle swarm optimization（FPSO）. The FPSO adopts a fuzzy inertia weight, and a fuzzy the particle position updata. Simulation results show that the algorithm has efficiency to complex function optimization. Sec. 2 begins with a review of the PSO. Fuzzy adaptive PSO is proposed in detail in Sec. 3. In Sec. 4, simulation results compare to the PSO and the linear descending inertia weight PSO are presented. Last, conclusion is gotten in Sec.5.

II. STANDARD PARTICLE SWARM OPTIMIZATION ALGORITHM

In the standard PSO algorithm, each individual is named as a "particle", which, in fact, represents a potential solution to a problem. Each particle is treated as a point in a D-dimensional space, which adjusts its flying according to its own flying experience and its companions' flying experience. The ith particle is represented as $x_i(t) = (x_{i1}(t), x_{i2}(t), \cdots, x_{id}(t))$, and its velocity is represented as $v_i(t) = (v_{i1}(t), v_{i2}(t), \cdots, v_{id}(t))$. In each iteration, the best individual position is represented as $pb_i(t) = (pb_{i1}(t), pb_{i2}(t), \cdots, pb_{id}(t))$, denoted by $pbest$, and the best global position is represented as $gb_i(t) = (gb_1(t), gb_2(t), \cdots, gb_d(t))$, denoted by $gbest$. The particles are manipulated according to the following equation:

$$v_{id} = \omega * v_{id}(t) + c_1 * r_1 * (pd_{id}(t) - x_{id}(t)) + c_2 * r_2 * (gd_{id}(t) - x_{id}(t)) \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (2)$$

Where $i = 1,2,\cdots,m; d = 1,2,\cdots,D$, c_1, c_2 are positive constants, r_1, r_2 are random value in the range [0,1], ω is inertia weight .In the equation (1), the first part is the momentum, expressing the affection of individual itself. The second part is individual cognition which represents the particle flying to its best position, and the last part is social part, which guides the particles flying to the best global position. The balance among the three parts determines the algorithms search capability.

III. FUZZY ADAPTIVE PARTICLE SWARM OPTIMIZATION (FPSO)

A. *The fuzzy inertia weight*

In the standard PSO, inertia weight ω is the most important parameter. When ω is large, it can increase space exploration ability, which is suitable for a wide-scale search problem, on the other hand, when ω is small, the particle local search ability is improved. According to linear decreasing weight [3] strategy and uncertain search factor, we adopt a fuzzy adaptive process to ω.

$$\omega' = \mu_1 \omega, \quad \mu_1 = \begin{cases} 1 & t \le t_{\max}/5 \\ \exp[\frac{-(t-20)^2}{\sigma_{ij}^2}] & t > t_{\max}/5 \end{cases} \quad (3)$$

In formula (3), the membership function is Gauss distribution, $t_{\max}$ is the maximum iteration number. Using the formula (3), decreasing property of ω parameter is reserved as

978-0-7695-4281-2/10 $26.00 © 2010 IEEE
DOI 10.1109/ICGEC.2010.10

well as ω is adaptively controlled, so it can regulate the global local search flexibly. So the formula (1) is changes to (5) by formula (3):

$$v_{id}(t+1) = \omega' * v_{id}(t) + c_1 * r_1 * (pd_{id}(t) - x_{id}(t)) + c_2 * r_2 * (gd_{id}(t)' - x_{id}(t)) \quad (4)$$

B. Fuzzy particle position update

In the standard PSO algorithm, the range of particle velocity is limited, but the particle search space isn't restricted. $x_{id}(t+1)$ changed greatly depends on the large change of $v_{id}(t+1)$. Thus, in the pursuit processing, when the particle is close to the optimal particle, the particle is easy to fall into local optimum. So, we use a fuzzy parameter to control the particle position change, when iteration number is small, μ_2 is 1,when iteration number is larger than a given threshold value. The particles position change slowly, so it can effectively avoid falling into local optimum. Equation (2) becomes:

$$x_{id}(t+1) = x_{id}(t) + \mu_2 * v_{id}(t+1) \quad (5)$$

Where $\mu_2 = \begin{cases} 1 & t \le T \\ [1 + (\frac{t-30}{a})^2]^{-1} & t > T \end{cases}$

a is a constant, generally obtained in the range [5,10] . T is a given threshold, related to the iterations maximum number $t_{\max}$.

C. Fuzzy Adaptive Particle Swarm Optimization (FPSO) Process

To get the improved algorithm, the fuzzy adaptive particle swarm optimization(FPSO), we use a fuzzy inertia weight , a fuzzy processing to location update, and a weighted average of all the particles instead of global optimization . The detailed process is described as follows:

Step1: Initialize all the particle's velocity and position;

Step2: Calculate each particle fitness value, so to determine the $pbest$ and $gbest$;

Step3: Using the fuzzy process on the inertia weight and the location update parameter, and update particle's velocity and position according to the formula (4) and (5);

Step4: Determine whether to reach the accuracy, or reach the evolutionary frequency. If reached, the program is over, and the $x_i(t+1)$ is the solution of the problem, otherwise, we turn to the step2.

IV. SIMULATION

To verify the effectiveness of the proposed algorithm, we have conducted a comparative experiments to linear descending inertia weight PSO in reference [3] (denoted by WPSO), and the standard particle swarm optimization (PSO). Using Rosenbrock, Rastrigin, Griewank and Ackley test functions in the simulation, and we want to get function minimum. The expressions of four test functions are expressed as follows:

(1)Rosenbrock function

$$f_1(x) = \sum_{i=1}^{n-1}\left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2\right]$$

(2) Rastrigin function

$$f_2(x) = \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i) + 10]$$

(3) Griewank function

$$f_3(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$$

(4) Ackley function

$$f_4(x) = -20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\frac{1}{n}\sum_{i=1}^{n}(2\pi x_i) + 2) + 20 + e$$

Experimental parameter settings are as follows: the particle size of three algorithms are all 40, the maximum evolutionary frequency is 1000, $c_1 = c_2 = 1.4962$,ω=0.7298, in the WPSO, the ω decreases to 0.5 linearly with iterative increasing. $v_{id} \in [-v_{\max}, v_{\max}]$, $v_{\max}$ is a constant.

TABLE I. PSO, WPSO AND FPSO OPTIMIZATION RESULTS

function	Search space	theory optimal	algorithm	average optimal
$f_1(x)$	[-100,100]	0	PSO WPSO FPSO	3.4356 1.8723 1.7562
$f_2(x)$	[-100,100]	0	PSO WPSO FPSO	3.4378 3.4368 3.4362
$f_3(x)$	[-512,512]	0	PSO WPSO FPSO	6.2334 2.8745 2.1123
$f_4(x)$	[-1024,1024]	0	PSO WPSO FPSO	8.2371 4.3482 4.3237

To eliminate random affection, each algorithm is run 20 times to each test function, and the average values are the optimal results. The initial $v_{\max}$ of each test function and the average optimization values of each algorithm are shown in Table 1. Rastrigin and Griewank function evolution curves of 1000 generation are shown in Figure 1 and Figure 2. As Shown in Table 1, we can see that the proposed fuzzy adaptive particle swarm algorithm(FPSO) have much better performance than the basic PSO algorithm ,and litter better compared with the

WPSO .By using the fuzzy inertia weight and the fuzzy particle position change strategy, FPSO escapes local minimum effectively. Figure 2 shows that when the evolution generations is 1000, the Rastrigin function has reached to global optimum after 600 generations in FPSO, 200 generations in WPSO, but less than 50 generations in the basic PSO .And we can obtain similar results in Griewank function shown in Figure 2. Therefore, FPSO algorithm not only improves the convergence accuracy but also avoid local convergence effectively.

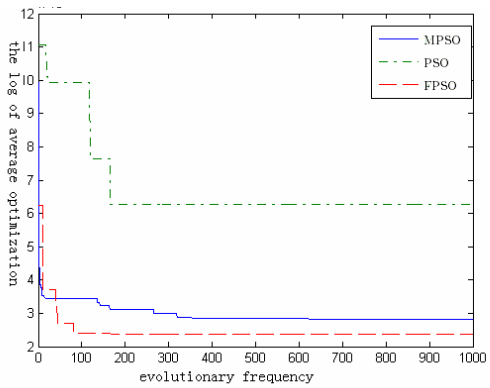

Figure 1. The evolution curve of 20-dimensional Griewank

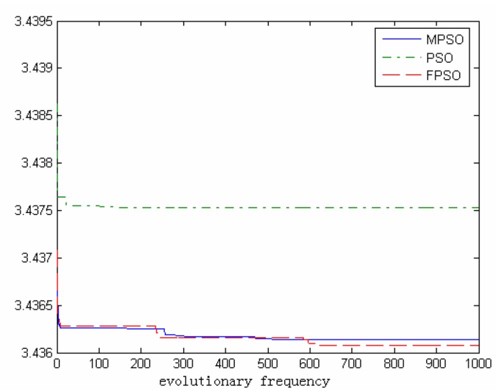

Figure 2. The evolution curve of 30-dimensional Rastrigin

V. CONCLUSIONS

The basic PSO algorithm is easy to fall into the local convergence. In this paper, we have proposed a fuzzy adaptive particle swarm algorithm (FPSO) by using a fuzzy inertia weight control and a fuzzy location updated control. Simulation results show that compared to the nearly decreasing inertia weight algorithm (WPSO) and basic PSO, the improved method not only avoids from falling into local convergence but also has better search capability and accuracy.

ACKNOWLEDGMENT

This work was supported by the Natural Science Foundation of Jiangxi Provincial Department of Education under the grant No.Gjj08417.

REFERENCES

[1] R. C. Eerhart, J.Kennedy, "A new optimizer using particle swarm theory", Proceedings of 6th International Symposium on Micro Machine and Human Science, Nagoya, pp. 39-43, 1995.

[2] Y. Shi, R. Eberhart, "Empirical study of particle swarm optimization", International Conference on Evolutionary Compution., Washington, pp. 1945-1950, 1999.

[3] Y. Shi Y, R. Ebethart, "A modified Particle swarm optimizer", IEEE World congress on Computational Intelligence, Anchonge , Alaska, pp. 69-73, 1998.

[4] A. Ratnaweeta, S. K. Halgamuge, H. C. Watson, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 240-255, 2004.

[5] J. J. Liang, P. N. Suganthan, "Dynamic muti-swarm particle swarm optimizer", IEEE Inernational Swarm Intelligence Stmposium, pp. 124-129, 2005.

[6] Jui-Fang Chang, Shu-Chuan Chu, John F. Roddick and Jeng-Shyang Pan, "A parallel particle swarm optimization algorithm with communication strategies", Journal of Information Science and Engineering, vol. 21, no. 4, pp. 809-818, 2005.

[7] Chao-Hsing Hsu, Wen-Jye Shyr, and Kun-Huang Kuo, "Optimizing multiple interference cancellations of linear phase array based on particle swarm optimization", Journal of Information Hiding and Multimedia Signal Processing, vol. 1, no. 4, pp. 292-300, October 2010.