American Journal of Operations Research, 2012, 2, 448-451
http://dx.doi.org/10.4236/ajor.2012.23053 Published Online September 2012 (http://www.SciRP.org/journal/ajor)

Dynamical Adaptive Particle Swarm Algorithm and Its Application to Optimization of PID Parameters

Jimin Li, Guolin Yu
Research Institute of Information and System Computation Science, The North University for Nationalities, Yinchuan, China
Email: guolin_yu@126.com

Received June 15, 2012; revised July 18, 2012; accepted August 5, 2012

ABSTRACT

Based on a new adaptive Particle Swarm Optimization algorithm with dynamically changing inertia weight (DAPSO), it is used to optimize parameters in PID controller. Compared to conventional PID methods, the simulation shows that this new method makes the optimization perfectly and convergence quickly.

Keywords: Particle Swarm Optimization; Dynamical Adaptive; PID Automatic Regulation System

1. Introduction

Particle Swarm Optimization is a kind of simulation group (Swarm) intelligent behavior of the Optimization of the algorithm proposed by Kennedy and Eberhart [1] and others. Its thought source from bird prey on behaveior research. Contrasting PSO with the genetic algorithm and the ant colony algorithm, the PSO method is simple and easy to implement, and it can be adjusted less parameters characteristics. So, it is widely applied in the structural design [2], electromagnetic field [3] task scheduling [4] engineering optimization problems.

In the particle swarm algorithm, the adjusted parameters are the most important part in the inertia weights. In order to find a inertia of weights selection method which can get the best balance between the global search and local search, the researchers have put forward to the linear decrease weights (LDIW) strategy [5], fuzzy inertia weights (FIW) strategy [6], and random inertia weights (RIW) strategy [7], and so on.

In the basic thought of diminishing inertia value guidance, this paper introduces a new adaptive self-adapting inertia, which is based on expectations of survival rate. When the expected survival rate gets smaller, it shows that the optimal particle distance to position is further, at this time, we should make particles into the local search rapidly, the authors shall propose a new dynamic adaptive particle swarm optimization algorithm (DAPSO), and using this algorithm to calculate optimizer in the PID regulator parameters of industrial control system.

This paper is divided into five sections. Section 2 includes preliminaries and related results which will be used in later sections. Section 3 is devoted to introduce dynamic adaptive particle swarm optimization algorithm. In Section 4, we shall use the dynamic adaptive particle swarm optimization algorithm to optimize the PID parameters. Section 5 is the conclusion and working direction in the further.

2. Basic PSO Algorithm and the Related Concept

Particle swarm optimization was firstly proposed in 1995 by Kennedy and Eberhart [1], using the following formula in the particle group of operation:

$$v_{id} = v_{id} + c_1 r_1 \left(p_{id} - x_{id} \right) + c_2 r_2 \left(p_{gd} - x_{id} \right) \quad (1)$$

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

which $i = 1, 2, \cdots, m$, $d = 1, 2, \cdots, D$; Learning factor c_1, c_2 is negative constants ; The parameters $r_1, r_2 \in [0,1]$ is between the random numbers $v_{id} = [-v_{\max}, v_{\max}]$; $v_{\max}$ is constant. The i-th particles in a D-dementional vector is $x_i = (x_{i1}, x_{i2}, \cdots, x_{iD})$, In the space of flight speed is v_i and $v_i = (v_{i1}, v_{i2}, \cdots, v_{iD})$; The i-th known so far to search the optimum position is $p_i = (p_{i1}, p_{i2}, \cdots, p_{iD})$, the particle swarm so far to search the optimal position is $p_g = (p_{g1}, p_{g2}, \cdots, p_{gD})$. In the literature [7], the speed Equation (1) have made the following changes:

$$v_{id} = w v_{id} + c_1 r_1 \left(p_{id} - x_{id} \right) + c_2 r_2 \left(p_{gd} - x_{id} \right) \quad (3)$$

where w is inertia coefficient, c_1, c_2 are nonnegative, Equations (1) and (3) are basic PSO iterative formula.

Experimental results show the particle swarm optimization algorithm whether premature convergence or global convergence, particle swarm of particles will appear "together" phenomenon. Therefore, we study the change of the affinity between particles, particle swarm state can be track, the affinity between particles reflect the similarity between the particle, when affinity is large, and the number particles are more similar. Conversely, similar degree is poor. We shall make the parts thought of immune algorithm to fuse the particle swarm optimization algorithm, and give the new definition of affinity and the concept of antibody concentration, the construction method of nonlinear adaptive inertia the weights, and based on this, it improves the dynamic adaptive particle swarm optimization algorithm in [8].

3. Dynamic Adaptive Particle Swarm Optimization Algorithm

Now, we introduce the dynamic adaptive particle swarm optimization algorithm:

DAPSO

Step 1: The particle position and speed of Random initial particle swarm.

Step 2: Make the particles, p_b to be the current position, set p_g to the best position of particles in initial population.

Step 3: Judg algorithm convergence criteria whether meet, if meet, true to Step 5; or, execute Step 4.

Step 4: To the all particles of particle swarm, execute the following operations:

1) According to the type (1), (2), (3) update the particle's position and speed;

2) According to the type of above, deduce a dynamic adaptive inertia weights, turn to Step 2.

Step 5: output p_g, the algorithm to run is end.

4. Using the Dynamic Adaptive Particle Swarm Optimization Algorithm to Optimize the PID Parameters

4.1. PID Regulator Parameters' Optimization Problem

In the industrial control system, lots of control objects under the action of step representation the output signal is *S* form rises curve, right now, can use a second order inertia and delayed model to describe it. Its transfer function is:

$$W(S)=\frac{Ke^{-T_3S}}{(T_1S+1)(T_2S+1)}$$

In order to make the control object's output y in disturbance action maintain, we usually use proportion, integral, differential (*P*, *I*, *D*) regulator to form a constant adjustment system.

When the production process is stable, which object characteristics are stable, *K*, *T*1, *T*2 and *T*3 are basically constant, at this time, PID parameters which are once adjusted would not be charged, but, when the production process is in a constantly changed situation, the reaction conditions such as chemical engineering change, power plant and the change of load, so some object *K*, *T*1, *T*2 and *T*3 will have a corresponding changed, at this time, if we still keep PID parameters unchanged, it will be difficult to achieve the optimal regulation effect. In the condition of constant fluctuation, because the computer has strong ability of calculation and control on flexibility, we can achieve the adjustment type DDC control. Therefore, using the computer to carry on the PID regulator parameters adjustment and control system has the very strong superiority [9].

Now using a computer to achieve P, I, D adjustment, then we control u and deviation between e to satisfy the

$$u(n)=K_P\left[e(n)+\frac{1}{T_I}\sum_{K=0}^{n}e(K)T+T_d\frac{e(n)-e(n-1)}{T}\right]$$

or

$$u(n)=u(n-1)+K_P\left\{\left[e(n)-e(n-1)\right]+\frac{T}{T_I}e(n)+\frac{T_d}{T}\left[e(n)-2e(n-1)+e(n-2)\right]\right\}$$

Among them, $u(n)$ for the control effect; $u(n-1)$ for the previous control function; $e(n)$ for the deviation; and $e(n-1)$ for the previous deviation; and $e(n-2)$ for again the first deviation.

T is the sampling period, K_P is gain coefficient ratio, T_I is integral time constant, T_d is differential time constant.

Now the task is to search the best: seek the PID three parameters K_P, T_I, T_d, to make the objective function is the smallest

It belongs to nonlinear planning in the multivariate function optimization problem, and can not use a mathematical

$$Q=\int_0^\infty |e|t\mathrm{d}t$$

expression to express the relations between the objective function and variables K_P, T_I, T_d. In this situation, this paper uses the dynamic adaptive particle swarm optimization algorithm and obtains the better results.

4.2. The Results of the Simulation Experiment

If we input the following data:

Variable number N = 3;

the calculation precision

E = 0.01;

Compression factor is 0.618; Expansion factor is 1.5.

The parameters of the controlled objects

$T_1 = 0.44$ s; $T_2 = 0.44$ s; $T_3 = 0.12$ s;

The total number of print L3 = 30; a given number of control system's input value R = 10;

PID parameters K_P, T_I, T_d initial value $X(1, 0) = 1.5$; $X(2, 0) = 0.88$; $X(3, 0) = 0.11$.

For DAPSO, the size of the group set to 30, the terminate algebra is 50, set for this d range of 15%, c_1, c_2 are all 2, a maximum of self-adapting inertia is 0.8, minimum number is 0.2.

The calculated parameters correspond to the control object T_1, T_2 and T_3 is with a given number, the optimal number of PID regulator parameters K_P, T_I, T_d: $K_P = 1.69764$, $T_I = 0.772662$ s, $T_d = 0.229209$ s.

In addition, when K_P, T_I, T_d is the best number, we still obtain all sampling moments of output X_1, and deviation quantity X_5, as is shown in **Table 1**.

The system output overshoots is E = 1.42%, the transition process time is TP = 1.1 s.

Table 1. The output results of system simulation.

T	X_1	X_5
0.1	0.1011220	9.898878
0.2	0.8997235	9.100277
0.3	3.5021150	6.497885
0.4	5.5186300	4.481378
0.5	7.8396510	2.160349
0.6	8.6678680	1.332132
0.7	9.9175360	0.082464
0.8	10.0164380	–0.016438
0.9	10.1245500	–0.124550
1.0	10.1423600	–0.142360
1.1	10.1318500	–0.131850
1.2	10.0348000	–0.034799
1.3	9.9862700	0.013728
1.4	9.9598700	0.040128
1.5	9.9487600	0.051248
1.6	9.9873600	0.012639
1.7	9.9963200	0.003678
1.8	10.0110	–0.011012
1.9	10.0106	–0.010585
2.0	10.0173	–0.017387
2.1	10.0186	–0.018595
2.2	10.0182	–0.018186
2.3	10.0172	–0.017153
2.4	10.0162	–0.016168
2.5	10.0154	–0.015366
2.6	10.0142	–0.014176
2.7	10.0136	–0.013582
2.8	10.0121	–0.012085
2.9	10.0092	–0.009192
3.0	10.0086	–0.008568

According to the standard of the engineering design method I system, the system of the output overshoot is E = 4.3%, the transition process time is TP = 1.76 s. Contrast with the results, it shows the parameters which get by using this algorithm are much superior

5. Conclusion and Working Direction

This paper puts forward a dynamic changing the inertia weights of the adaptive particle swarm algorithm (DAPSO), and using the algorithm to optimize the PID parameters, the simulation results show that this method, contrasted with a relatively routine of the simplex method, has a lot of advantages, such as the fast speed, accuracy, the optimal speed, accurate, to improve the dynamic system, etc. Having the prospect of the good application and the further research, however, for the function, which is very complex and exists many local optimizations near the optimal solution, the results are not very ideal. Therefore, further work direction is: expanding the scope of the algorithm testing, seeking the cooperate ways between particles, and to a certain extent, analysis, the convergence and robustness of PAPSO.

6. Acknowledgements

This work is supported by Natural science foundation of the State Ethnic Affairs Commission of PRC (No. 10BF07); Zizhu Natural Science Foundation of Beifang University for Nationalities (No. 2011ZQY024). Ministry of Education Science and technology key projects (No. 212204); Natural Science Foundation for the Youth (No. 10901004).

REFERENCES

[1] J. Kennedy and R. Eberhert, "Particle Swarm Optimization," *IEEE International Conference on Neural Networks*, IEEE Service Center Press, IV. Piscataway, New Jersey, 1995, pp. 1942-1948.

[2] C. Elegbede, "Structural Reliability Assessment Based on Particles Swarm Optimization," *Structural Safety*, Vol. 27, No. 2, 2005, pp. 171-186. doi:10.1016/j.strusafe.2004.10.003

[3] J. Pobinson and Y. Rahmat-Samii, "Particle Swarm Optimization in Electromagnetics," *IEEE Transactions on Antennas and Propagation*, Vol. 52, No. 2, 2004, pp. 397-406. doi:10.1109/TAP.2004.823969

[4] A. Salman, I. Ahmad and S. Al-Madani, "Particle Swarm Optimization for Task Assignment Problem," *Microprocessors and Microsystems*, Vol. 26, No. 8, 2002, pp. 363-371. doi:10.1016/S0141-9331(02)00053-4

[5] Y. Shi and R. Eberhart, "Empirical Study of Particle Swarm Optimization," *International Conference on Evolutionary Computation*, IEEE Service Center Press, Washington DC, 1999, pp. 1945-1950.

[6] Y. Shi and R. Eberhart, "Fuzzy Adaptive Particle Swarm

Optimization," *The IEEE Congress on Evolutionary Compution*, IEEE Service Center Press, San Francisco, 2001, pp. 101-106.

[7] R. Eberhart and Y. Shi, "Tracking and Optimizing Dynamic Systems with Particle Swarm," *The IEEE Congress on Evolutionary Computation*, IEEE Service Center Press, San Francisco, 2001, pp. 94-100.

[8] J. M. Li, C. M. Lei and Y. Qiao, "Based on Expectations of Survival Rate Dynamic Adaptive Particle Swarm Algorithm," *Journal of Ningxia University*, Vol. 12, 2009, pp. 347-350.

[9] Y. X. Yuan and W. Y. Sun, "Optimization Theory and Method," Science Press, Beijing, 1999, pp. 69-75.