ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A FAST QUANTUM-BEHAVED PARTICLE SWARM OPTIMIZATION BASED ON ACCELERATING FACTOR

[1]ZHEHUANG HUANG
[1]School of Mathematics Sciences, Huaqiao University, Quanzhou, 362021, China
Cognitive Science Department, Xiamen University, Xiamen, 361005, China
E-mail: hzh98@stu.xmu.edu.cn

ABSTRACT

Quantum-behaved particle swarm optimization (QPSO) is a good optimization technique which has been successfully applied in many research and application areas. But traditional QPSO algorithm is easy to fall into local optimum and the rate of convergence is slow. To solve these problems, an improved algorithm based on dynamic adjustment of the acceleration factor is proposed. The experiments on high dimensional function optimization showed that the improved algorithm has more powerful global exploration ability and faster convergence speed.

Keywords: *QPSO; Accelerating Factor; Function Optimization*

1. INTRODUCTION

Modern heuristic algorithms are considered as practical tools for nonlinear optimization problems. Particle swarm optimization (PSO) is a population-based, self-adaptive search optimization technique first introduced by Kennedy and Eberhart [1, 2] in 1995. The method has been developed through a simulation of simplified social models.

PSO is a kind of algorithm based on social psychology. Like GA [3], PSO must also have a fitness evaluation function. Compared with GA, PSO has some attractive characteristics. It has memory, so knowledge of good solutions is retained by all particles [4]. But in GA, previous information is destroyed once the population changes.

The PSO method is becoming very popular due to its simplicity of implementation and ability to quickly converge to a reasonably good solution. In recent years, there has been increasing interest in developing the PSO algorithm. PSO has been applied successfully to all kinds of optimization problems such as nonlinear functions [5], neural networks [6-8], power and voltage control [9], and task assignment problem [10], etc.

This paper is organized as follows. In section 2 the background of PSO and QPSO is presented. The improved algorithm is presented in section 3. In Section 4 some experimental tests, results and conclusions are given. Section 5 concludes the paper.

2. INTRODUCTION TO PSO

2.1. Background

PSO is based on swarms such as fish schooling and bird flocking. According to the research results for bird flocking, birds find food by implementing an "information sharing" approach.

Each particle moves over the search space according to the historical behaviors of the particle and its companions. Suppose that the location of the ith particle is represented as $X_i = (x_{i1}, x_{i2,} \cdots x_{id})$. The best previous position of the ith particle is represented as $P_i = (p_{i1}, p_{i2,} \cdots p_{id})$, which is also called pbest. The location p_g is also called gbest. The velocity for the ith particle is represented as $V_i = (v_{i1}, v_{i2,} \cdots v_{id})$.

If g is the index of the particle visited the best position in the swarm, and then p_g becomes the best solution found so far, and the velocity of the particle and its new position will be determined according to the following equations (1a) and (1b), respectively:

$$vid = w \times vid + c1 \times rand() \times (pid - xid)$$
$$+ c2 \times rand() \times (pgd - xid) \quad \text{(1a)}$$

$$xid = xid + vid \quad \text{(1b)}$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Where the function $rand()$ can generate a random number between 0 and 1. w is called the inertia weight. $c1$ and $c2$ are two constant numbers, which are often called the cognitive confidence coefficients.

The procedure for implementing the PSO is given by the following steps:

Initialize the population x_i
for each particle x_i do
  update the particle's best position
  if f(x_i) < f(pbi) then
  pbi = x_i
  end if
  update the global best position
  if f(pbi) < f(gb) then
  gb = pbi
  end if
  end for
update particle's velocity and position
for each particle x_i do
  for each dimension d in D do
  update particle's velocity by equation (1a)
  update particle's position by equation (1b)
end for
end for
it = it + 1
until it < max iterations

Algorithm 1. The PSO algorithm

2.2. Quantum Model of PSO

The development of quantum mechanics [11, 12] forced the scientists to rethink the applicability of classical mechanics and the traditional understanding of the nature.

In classical PSO algorithm, a particle is depicted by its position vector and velocity vector, which determine the trajectory of the particle. In quantum world, the term trajectory is meaningless, because position vector and velocity vector of a particle can not be determined simultaneously according to uncertainty principle. Some researchers consider a social organism is a system far more complex than that formulated by particle swarm optimization, and a linear evolvement equation is not sufficient to depict it at all. Quantum-behaved Particle Swarm Optimization (QPSO) was proposed [13, 14] in 2004. QPSO is inspired by quantum mechanics and fundamental theory of particle swarm. Wave function of position [15-17]depicts the state of the particle in quantized search space, not informing us of any certain information about the position of a particle that is vital to evaluate the fitness of a particle.

A global point denoted as mbest is introduced into PSO:

$$mbest = \frac{1}{M}\sum_{i=1}^{M} P_i$$

$$= (\frac{1}{M}\sum_{i=1}^{M} P_{i1}, \frac{1}{M}\sum_{i=1}^{M} P_{i2}, \cdots \frac{1}{M}\sum_{i=1}^{M} P_{id}) \quad (2)$$

The procedure of QPSO is given by the following steps:

Initialize the population x_i
Do
find out the mbest of the swarm by equation (2);
  for I=1 to population size M;
  if f(p_i)<f(x_i) then $p_i = x_i$
  p_g =min(p_i)
  for d=1 to dimension D
  f_{i1} =rand(0,1), f_{i2} =rand(0,1)
  P=(f_{i1} *pid+ f_{i2} *pgd)/(f_{i1} + f_{i2})
  L=beta*abs(mbestd- x_{id})
  u=rand(0,1);
  if rand(0,1)>0.5;
  x_{id} =p-abs(mbestd- x_{id})×log(1/u);
  else
  x_{id} =p+abs(mbestd- x_{id})×log(1/u);
  until the termination criterion is met.

Algorithm 2 The OPSO algorithm

3. THE IMPROVED ALGORITHM BASED ON ACCELERATING FACTOR (AQPSO)

From the QPSO algorithm, we can know local attractor q is determined by local optimum $pbest$ and global optimum $gbest$:

$$q = \varphi \bullet pbest + (1-\varphi) \bullet gbest \ ,$$

$$\varphi = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2}, r_1 \sim U(0,1), r_2 \sim U(0,1)$$

From standard PSO algorithm, it is known that c_1 and c_2 as two accelerating factor, will not only affect the algorithm convergence speed, at the

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

same time, also may lead to premature phenomenon occurs. In the early stage, if the local optimal has faster convergence speed and the global optimal has slower convergence speed, or in the late stage, the local optimal has slower convergence speed and the global optimal has faster convergence speed, can effectively avoid the happening of premature. In the early stages, the particles and the optimal location are far away. At later stage, the position of the particle and the optimal location are closer. We can use the error function represented the closer degree between particles and the global optimal position, so as to select the corresponding acceleration factor.

The particles closer to the global optimal position, the acceleration factor. Let error function as follows:

$$\Delta F = \frac{ABS(F_i - F_{gbest})}{MIN(ABS(F_i), ABS(F_{gbest}))}$$

Where F_i is the fitness of the ith particle, F_{gbest} is the fitness of global optimal, $ABS(F_i)$ is absolute value of F, thus acceleration factor function is:

$$c1 = \begin{cases} (Mc-mc)\times\dfrac{MAXITER-t}{MAXITER}+mc, & \Delta F > F_0 \\ (Mc-mc)\times\sin(\dfrac{MAXITER-t}{MAXITER})+mc, & \Delta F < F_0 \end{cases}$$

$$c2 = \begin{cases} (Mc-mc)\times\sin(\dfrac{MAXITER-t}{MAXITER})+mc, & \Delta F > F_0 \\ (Mc-mc)\times\dfrac{MAXITER-t}{MAXITER}+mc, & \Delta F < F_0 \end{cases}$$

(3)

Where $F_0 = 0.3$.

The procedure for implementing the AQPSO is given by the following steps:

Step 1 . Initialize all particles;

Step 2. Calculation of the value of the variable of β :

$\beta = (1-0.5)*(MAXITER-t)/MAXITER + 0.5$

, where t is the number of iterations ;

Step3 . Calculating the fitness of each particle;

Step 4. Calculated error values according to the error function $\Delta F = \dfrac{f(x_i) - f(P_{gj})}{MIN(ABS(f(x_i)), ABS(f(P_{gj})))}$;

Step5. Calculated accelerating factor based on the acceleration factor function;

Step 6. Obtain the average optimal position:

$$mbest_j = (\frac{1}{N}\sum_{i=1}^{N}P_{i1}, \frac{1}{N}\sum_{i=1}^{N}P_{i2}, \frac{1}{N}\sum_{i=1}^{N}P_{i3}, \cdots, \frac{1}{N}\sum_{i=1}^{N}P_{ij})$$;

Step 7. Obtain the local attractor:

Update $c1$ and $c2$ by equation (3) ;

$$\varphi = \frac{c_1 r_1}{c_1 r_1 + c_2 r_2}, r_1 \sim U(0,1), r_2 \sim U(0,1)$$

$$p_{ij}(t) = \varphi_{ij}(t)\cdot P_{ij}(t) + (1-\varphi_{ij}(t))\cdot P_{gj}(t)$$

Step 8. update the particle's velocity and position

$$x_{ij}(t+1) = p_{ij}(t) \pm \beta\left|mbest_j(t) - x_{ij}(t)\right|\cdot \ln(1/\mu_{ij}(t))$$

Step 9. If iteration terminated, output optimal value; otherwise return 3 and continue.

4. EXPERIMENTAL RESULTS

A set of unconstrained real-valued benchmark functions was used to investigate the effect of the improved algorithm. The functions are shown in Table 1.

The results are shown in Table 2. Each point is made from average values of over 30 repetitions.

For sphere function and Ackey function, AQPSO algorithm can effectively improve the accuracy such that the optimal value obtained is much closer to the theoretical one and the accuracy is improved compared with the standard QPSO algorithm. Shaffer function is a multimodal function, from the results of iteration, the convergence effect of this algorithm is not as good as the standard QPSO algorithm, but the difference is small and acceptable. On the convergence time, there is a significant improvement as expected. These experimental results show that improving algorithm can effectively improve the convergence speed with

Journal of Theoretical and Applied Information Technology
20th February 2013. Vol. 48 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

excellent convergence effect.

Table 1 Functions Used To Test The Effects Of AQPSO

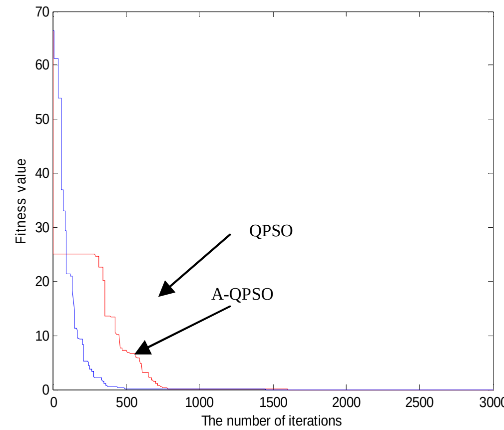
Function	Function expression
Sphere function	$f_1(x)=\sum_{t=1}^{n} x_t^2$
Rastrigrin function	$f_3(x)=\sum_{t=1}^{n}(x_t^2-10\cos(2\pi x_t)+10)$
Griewank function	$f_2(x)=\frac{1}{4000}\sum_{t=1}^{n}(x_t-100)^2-\prod_{t=1}^{n}\cos(\frac{x_t-100}{\sqrt{t}})+1$
Ackey function	$f_3(x)=20+e-20e^{-0.2\sqrt{\frac{\sum_{t=1}^{n}x_t^2}{n}}}-e^{\frac{\sum_{t=1}^{n}\cos(2\pi x_t)}{n}}$
Shaffer's function	$f_4(x)=0.5+\frac{(\sin\sqrt{x_1^2+x_2^2})^2-0.5}{(1+0.001(x_1^2+x_2^2))^2}$

Table 2 The Performances Of AQPSO

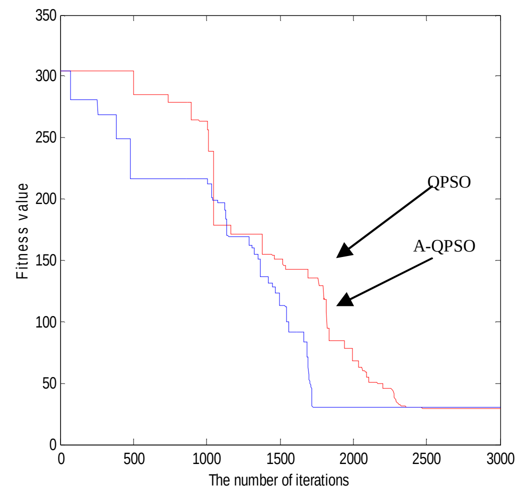
algorithm	*OPSO*		*AQPSO*	
	optimal	time(s)	optimal	time(s)
Sphere	5.29e-29	3.67	3.12e-30	3.1
Rastrigrin	23	4.08	23.8	3.27
Griewank	0	3.93	0	3.81
Ackey	2.79e-3	4.13	2.66e-3	3.84
shaffer	0	3.67	1.2e-9	3.02

The comparison of two methods with convergent curves is shown in Figs.1. to 5.

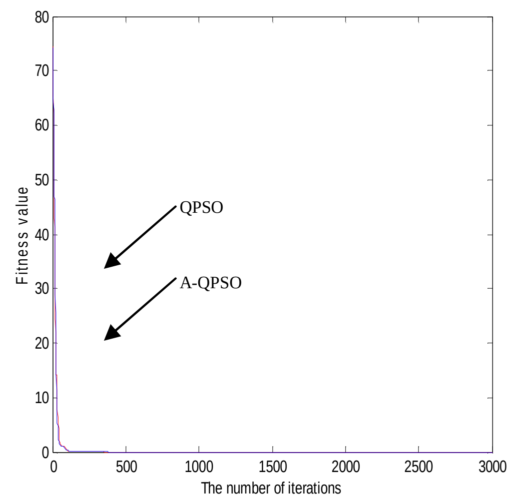
The experiment results show the AQPSO algorithm has better result. Comparisons with QPSO, the AQPSO algorithm has both global search ability and fast convergence speed.



Figs.1. Sphere Function

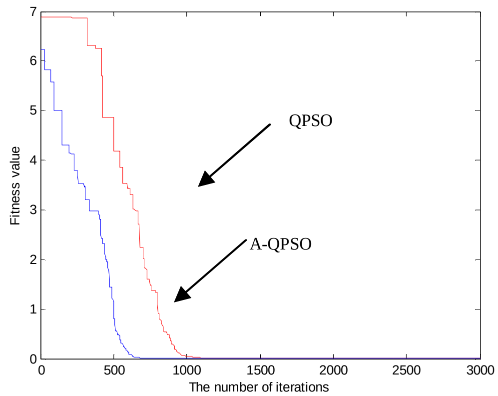


Figs.2. Rastrigrin Function

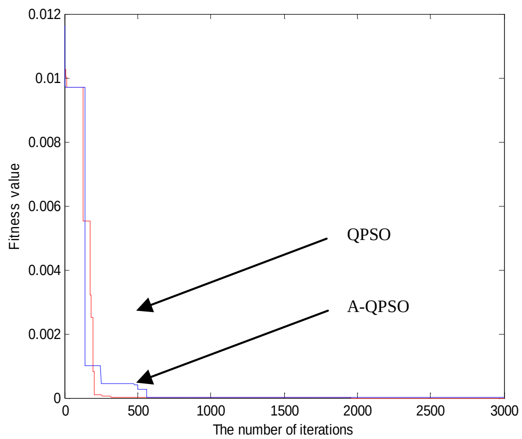


Figs.3. Griewank Function

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195



Figs.4. Ackey Function



Figs.5. Shaffer's Function

5. CONCLUSION

From above, we can know the improved algorithm have more powerful global exploration ability and faster convergence speed and can be widely used in other optimization tasks.

REFRENCES:

[1] R. C. Eberhard and J. Kennedy, "A new optimizer using particle swarm theory", In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Japan, 1995,pp. 39-43.

[2] J. Kennedy and R. C. Eberhard, "Particle swarm optimization", In Proceedings of IEEE Int'l Conf. on Neural Networks, USA, 1995 ,pp. 1942-1948.

[3] M. Clerc and J.Kennedy,"The particle swarm—Explosion, stability, and convergence in a multi-dimensional complex space", IEEE Trans. Evol. Comput., Vol. 6, 2002, pp. 58–73.

[4] Russell C. Eberhart, Yuhui Shi,"Particle Swarm Optimization: Developments, pplications and Resources", In Proceedings of the IEEE Congress on Evolutionary Compufation, Korea, 2001.

[5] J. Kennedy,"The particle swarm: Social adaptation of knowledge", In Proceedings of IEEE Int. Conf. Evolutionary Computation, 1997,pp. 303–308.

[6] Jian Zhang,"Experimental Parameter Investigations on Particle Swarm Optimization Acceleration Coefficients",International Journal of Advancements in Computing Technology, Vol. 4, No. 5, 2012,pp. 99- 105.

[7] Zengfu Xi, Yunxin Zhang, Changjun Zhu,"Application of PSO-Neural Network Model in Prediction of Groundwater Level in Handan City", Advances in Information Sciences and Service Sciences, Vol. 4, No. 6, 2012,pp. 177-183.

[8] C. Zhang, H. Shao, and Y. Li,"Particle swarm optimization for evolving artifical network", In Proceedings of IEEE Int. Conf. Syst., Man, Cyber., Vol. 4,2000, pp. 2487–2490.

[9] H. Yoshida, K. Kawata, Y. Fukuyama, and Y. Nakanishi,"A particle swarm optimization for reactive power and voltage control considering voltage security assessment", IEEE Transactions on Power Systems, Vol. 15, 2000,pp.1232-1239.

[10] A. Salman, I. Ahmad, S. Al-Madani,"Particle swarm optimization for task assignment problem.Microprocessors and Microsystems", Vol. 26, 2003, pp. 363-371.

[11] Roos, C. F., Chwalla, M., Kim, K., Riebe, M. & Blatt, R, "Designer atoms' for quantum metrology", Nature, Vol. 443, 2006, pp. 316–319.

[12] Milburn, G. J., Schneider, S., James, D. F. V,"Ion trap quantum computing with warm ions",Fortschr.Phys. Vol. 48, 2000, pp. 801–810.

[13] J.. Sun, B. Feng, and W. Xu,"Particle swarm optimization with particles having quantum behavior",In Proceedings of theCongress on Evolutionary Computation, Vol. 1, 2004 ,pp.325–331,.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[14] J. Sun, W. Xu, and B. Feng,"A global search strategy of quantum-behaved particle swarm optimization",In Proceedings of IEEE Conference on Cybernetics and Intelligent Systems,2004,pp. 111–116.

[15] L. D. S. Coelho ,"A quantum particle swarm optimizer with chaotic mutation operator", Chaos, Solitons and Fractals, Vol. 37, No. 5, 2008, pp. 1409–1418.

[16] S. Yang, M. Wang and L. Jiao,"A Quantum Particle Swarm Optimization", IEEE Congress on Evolutionary Computation, Vol. 1, 2004, pp. 320-324.

[17] Yin Guofu,"An Improved Quantum Particle Swarm Optimization Algorithm Based on Real Coding Method n", International Journal of Advancements in Computing Technology, Vol. 4, No. 3, 2012,pp. 181-188.