A Novel Dynamic Particle Swarm Optimization Algorithm Based on Chaotic Mutation

Min Yang
College of Information Engineering
Xiangtan University
Xiangtan, Hunan Province, China
ym1984xqp@163.com

Huixian Huang *
College of Information Engineering
Xiangtan University
Xiangtan, Hunan Province, China
huanghx0618@hotmail.com

Guizhi Xiao
College of Information Engineering
Xiangtan University
Xiangtan, Hunan Province, China
greedy304@hotmail.com

***Abstract*—A novel dynamic particle swarm optimization algorithm based on chaotic mutation (DCPSO) is proposed to solve the problem of the premature and low precision of the common PSO. Combined with linear decreasing inertia weight, a kind of convergence factor is proposed based on the variance of the population's fitness in order to adjust ability of the local search and global search; The chaotic mutation operator is introduced to enhance the performance of the local search ability and to improve the search precision of the new algorithm. The experimental results show finally that the new algorithm is not only of greater advantage of convergence precision, but also of much faster convergent speed than those of common PSO (CPSO) and linear inertia weight PSO (LPSO).**

Keywords- particle swarm optimization; chaotic mutation; dynamic inertia weight; convergence factor

I. Introduction

Particle Swarm Optimization (PSO) is a population-based random search strategy and an adaptive optimization algorithm developed by Dr. Eberhart and Dr. Kennedy in 1995[1]. Because of the unique search mechanism, excellent convergence performance and easy realization, the algorithm has obtained rapid development and has been widely used in many fields since it was proposed [2-4].

Although particle swarm optimization algorithm has been developing rapidly, it is also of the problem of the premature and low precision. In order to solve the problem, a novel dynamic particle swarm optimization algorithm based on chaotic mutation is proposed in the paper. Combined with linear decreasing inertia weight, a kind of convergence factor is proposed based on the variance of the population's fitness in order to adjust the ability of local search and global search; the chaotic mutation operator is introduced to enhance the performance of the local search ability and to improve the search accuracy of the new algorithm. The simulation results show the new algorithm has much better performance on the convergence rate and convergence accuracy.

II. Common PSO Algorithm (CPSO) and Some Related Concept

In the PSO algorithm, the locations of particles present the potential solution. Each particle searches the optimal solution in the problem space according to its own experience and the experience of the other particles in the search space.

Let D be the dimension of the search space, $X_i = (x_1^i, x_2^i, \cdots, x_D^i)$ is noted as the current position of i^{th} particle of swarm, $V_i = (v_1^i, v_2^i, \cdots, v_D^i)$ is noted as the rate of the velocity for the i^{th} particle of swarm, and $P_i = (p_1^i, p_2^i, \cdots, p_D^i)$ is noted as the best position by which it has ever visited. g is noted as the index of the best particle among the particles in the population. $P_g = (p_1^g, p_2^g, \cdots, p_D^g)$ is noted as the best position by which the swarm have never visited. In basic PSO model, the particles update their velocities and positions with following formulas:

$$v_d^i(t+1) = wv_d^i(t) + c_1 r_1 (p_d^i - x_d^i(t)) + c_2 r_2 (p_d^g - x_d^i(t)). \tag{1}$$

$$x_d^i(t+1) = x_d^i(t) + v_d^i(t+1). \tag{2}$$

where $i = 1, 2, \ldots, \mathrm{N}$, $d = 1, 2, \cdots, \mathrm{D}$. c_1 and c_2 are positive constants, called cognitive and social coefficient respectively, r_1 and r_2 are two random numbers, t is the iteration number. The inertia weight w is considered critical for the convergence of the PSO. A large w facilitates the global search while a small w facilitates the local search. Through a great amount of experiments, literature [5] proposed a linear decreasing inertia weight approach (LDIW), which significantly improved the algorithm performance.

In the search processing of PSO, if some particles find a current optimal location, other particles will move close to it rapidly. But if the current optimal location is local optimal solution, particle swarm will not search the global optimal again, the algorithm will fall into the local best, this phenomena is called premature convergence. In order to track the state of particle swarm, we give the definition of the variance of the population's fitness:

*corresponding author

978-0-7695-3543-2/09 $25.00 © 2009 IEEE
DOI 10.1109/WKDD.2009.142

Definition 1[6] Let n be the number of particle swarm, f_i is the fitness of i^{th} particle, f_{avg} is the current average fitness of particle swarm, σ^2 is the variance of the population's fitness, then σ^2 can be defined as

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(\frac{f_i - f_{avg}}{f})^2 . \quad (3)$$

where $f_{avg} = \frac{1}{n}\sum_{i=1}^{n} f_i$, f is the normalization factor, which can limit the magnitude of σ^2, the value of f is calculated as follows:

$$f = \max\{1, \max\{|f_i - f_{avg}|\}\},\ i \in [1, n] . \quad (4)$$

From definition 1, it can be seen that the variance σ^2 of the population's fitness reflects the convergence degree of all particles in the population. The smaller the variance σ^2 is, the more convergent the particle swarm is. On the contrary, the particles are at the random search stage. Thus we can judge the convergence rate of particle swarm based on the change of the variance of the population's fitness between adjacent two generations. Next in this paper, we give a new definition of convergence ratio of the population:

Definition 2 Let σ_{i+1}^2 be the current variance of the population's fitness, σ_i^2 be the variance of the population's fitness of last generation. ρ is noted as the convergence ratio of the population, then ρ can be defined as follows:

$$\rho = \sqrt{\frac{\sigma_{i+1}^2}{\sigma_i^2}} . \quad (5)$$

From the definition 1, the variance of the population's fitness σ^2 reflects the convergence degree of all particles in the population. Therefore, the convergence ratio reflects the change of the convergence degree, in other words, it reflects the convergence speed. The smaller σ^2 is, the higher the convergence degree is. If $\rho < 1$, particles tend to convergence, otherwise, particles are at the random search stage.

III. NOVEL DYNAMIC PSO ALGORITHM BASED ON CHAOTIC MUTATION (DCPSO)

A. *Dynamic Inertia Weight Based on Convergence Factor*

PSO algorithm searches the optimal solution in the problem space relying on the mutual cooperation and competition between groups. When the particles aggregate around a local extremum, it is difficult to search the other region in the problem space, which leads to the so-called "premature convergence" phenomenon. Although the LDIW strategy has improved the convergence performance of PSO algorithm effectively, the change of w is only linear correlation with the iteration times, it can not adapt the complex and nonlinear characteristic of the operation process of algorithm.

Considering the convergence ratio reflect the speed of convergence, in order to make the particle swarm be convergent fast and accuracy, we design a random factor in this paper. While the convergence ratio is less than a certain value, the particles converges very fast, algorithm is easy getting in local optimum. At this situation, we increase the size of inertia weight, allowing the particles have the opportunity to search the other region. Contrarily, if the convergence ratio is greater than a certain value, particles converge slowly, even tend to become divergent. At this time, by decreasing inertia weight, the local search ability and global search ability of algorithm can be re-balanced to ensure the convergence rate. At the same time, algorithm adopts random factor to adjust the linear inertia weight in order to ensure that the particles will not get into local optimum. Accordingly, we propose a novel dynamic inertia weight. The formula of w is as follows:

$$w = randNum[w_{\max} - iter\frac{w_{\max} - w_{\min}}{iter_{\max}}] . \quad (6)$$

$$randNum = \begin{cases} rand\left(\frac{1}{2\rho}, \frac{1}{\rho}\right) & if \quad \rho > \rho_{\max} \\ rand\left(1.0, \frac{1}{\rho}\right) & if \quad \rho < \rho_{\min} \\ 1 & else \end{cases} . \quad (7)$$

where $\rho_{\max}$ and $\rho_{\min}$ are given values, $rand\left(\frac{1}{2\rho}, \frac{1}{\rho}\right)$ denotes a random number among the area $\left(\frac{1}{2\rho}, \frac{1}{\rho}\right)$. It is clear from the equation (6) ~ (7) when $\rho > \rho_{\max}$, $randNum$ is less than one and w is decreased, thus the local search ability of particles is enhanced. But, when $\rho < \rho_{\min}$, the convergence rate of particles is fast, $randNum$ is a random number more than one, w is increased, particles search area is expanded to avoid local optimum. The algorithm performance is improved through the adjustment of $randNum$.

B. *Chaotic Mutation Operator*

Chaos is the essential characteristic of the nonlinear system with randomness, ergodicity, regularity, as well as a series of special properties[7]. By introducing chaotic mutation, there is a wider distribution of particles to help particles jump out the local minimum point and speed up to find the global optimum. Chaotic mutation to the particles can reduce the evolutionary generations of algorithm, and find the optimal solution as soon as possible. Further more, the particles may be better after mutation operation. It can avoid local convergence and premature and it can improve search accuracy of the algorithm.

One-dimensional Logistic map is adopted as chaos model in mutation operator to improve the performance of the algorithm:

$$\gamma_d(t+1) = u\gamma_d(t)\left(1-\gamma_d(t)\right) \qquad t=1,2,\cdots \quad \gamma_d \in (0,1) . \tag{8}$$

The chaotic mutation form is designed as follows:

$$x_d'(t) = x_d(t) + \sigma_d(t)\gamma_d(t) . \tag{9}$$

where $\sigma_d(t) = \sigma_0 \exp(-ut)$, σ_0, u are constants.

C. The Description of DCPSO

Step1: Initialize a population of particles with random positions and velocities in D dimensions of the problem space.

Step2: Evaluate the fitness of each particle in the swarm.

Step3: Compare each particle's fitness and its personal best position $pbest$. If its fitness is better, replace $pbest$ with its fitness.

Step4: Compare each particle's fitness and the global best position $gbest$. If its fitness is better, replace $gbest$ with its fitness.

Step5: Calculate the variance of the population's fitness and convergence ratio, and then calculate the inertia weight according to expression (6).

Step6: Execute chaotic mutation to each particle according to expression (9).

Step7: Updating the positions and velocities of particles by using (1) and (2).

Step8: Stop the iteration process if a given stopping criterion is met. Otherwise, go to step2.

IV. Performance Test

A. Parameter Setting

Four non-linear functions appeared in [8] are selected here to test the optimization effect of the given algorithm, which are usually used to check the performance of some algorithm. The four functions are the Sphere function, Rosenbrock function, Rastrigrin function and Griewank function described respectively by equation (10) - (13):

$$f_1(x) = \sum_{i=1}^{n} x_i^2 . \tag{10}$$

$$f_2(x) = \sum_{i=1}^{n-1} [100 \times (x_{i+1} - x_i^2)^2 + (x_i - 1)^2] . \tag{11}$$

$$f_3(x) = \sum_{i=1}^{n} (x_i^2 - 10\cos(2\pi x_i) + 10) . \tag{12}$$

$$f_4(x) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 . \tag{13}$$

Three types of algorithms CPSO, LPSO and DCPSO have been adopted to get the optimal solution of the above-mentioned four functions. Experimental parameters are given as follows: $c_1 = c_2 = 2$, $w_{\max}$ and $w_{\min}$ are 0.9 and 0.4 respectively, population size n =50, the functional dimensions of all experiments are D=10, the max iteration number is $IterMax$ =100. The error limits of objective functions are 1.0e-18. The search spaces of variables are [-4, 4]. In this paper, the function value is the fitness value. Each case of the processes has been repeated for fifty times.

B. Test Results and Discussion

Use the following appraising indexes to analyze experiment result: Mean Best Fitness (MBF) and Standard Deviation (SD). MBF indicates the precision that the algorithm can get within given iterated times, and it reflects the algorithm's convergence velocity. SD reflects the algorithm's stability and robustness. The experiment result is shown at TABLE Ⅰ.

It is easy to see that both MBF and SD obtained from DCPSO is much better than the results which are obtained from the common PSO (CPSO) and from linear inertia weight PSO (LPSO). MBF indicates that the new algorithm can get better optimum result and have preferable convergence precision. SD indicates that the deviation of the optimum fitness result of the new algorithm is comparatively less. It indicates the new algorithm have better stability and robustness.

Fig.1-4 show the evolutionary curves of the mean function values of the four cases respectively. From the

TABLE I. Mean Best Value and Standard Deviation of the Optimal Value for Fifty Trials

Function	Method	MBF	SD
Sphere	*CPSO*	4.748642	5.485242
	LPSO	0.789054	0.326184
	DCPSO	1.7152e-007	8.1654e-008
Rosenbrock	*CPSO*	372.7252	1135.844
	LPSO	236.0047	425.1348
	DCPSO	0.610096	0.156843
Rastrigrin	*CPSO*	60.93297	40.21548
	LPSO	50.05334	28.15627
	DCPSO	0.594582	0.292514
Griewank	*CPSO*	0.300323	0.258467
	LPSO	0.162782	0.095215
	DCPSO	0.003456	0.000158

Fig.1-4, it is clear that not only the DCPSO has great advantage of convergence precision over CPSO and LPSO, but also it can be convergent much faster than CPSO and LPSO.

We can see that in the earlier stage, both CPSO and LPSO trend to be convergent. But in the later, CPSO is easy to get in local optimal points. LPSO have the trend to be convergent, but because of the inefficient iteration in the latter, it can hardly get the global optimum in the given iterated times. Because of the regulation of the convergence factor and chaotic mutation, DCPSO can get convergence quickly and reach the global optimum through its own regulatory mechanism.

V. Conclusion

A kind of convergence factor is proposed based on the variance of the population's fitness in order to adjust the local search and global search ability combined with linear decreasing inertia weight. The chaotic mutation operator is introduced to enhance the performance of the local search ability and to improve the search precision of the new algorithm. The experimental results show that the DCPSO algorithm has higher convergence precision and faster convergence rate, and can avoid premature convergence effectively, compared with the CPSO and LPSO algorithm.

Acknowledgment

This work is supported by Natural Science Foundation of Hunan Province (No.06JJ50112). The authors are grateful for the anonymous reviewers who made constructive comments.

References

[1] J. Kennedy, R.C. Eberhart, "Particle Swarm Optimization", Proceedings of the IEEE International Conference on Neural Networks, Piscataway, NJ, IV,1995, pp.1941-1948.

[2] Shinn-Ying Ho, Hung-Sui Lin, Weei-Hurng Liauh, Shinn-Jang Ho, "OPSO: Orthogonal Particle Swarm Optimization and Its Application to Task Assignment Problems", IEEE Transactions on Systems, Man and Cybernetics, Part A, Volume 38, Issue 2,2008, pp.288 – 298.

[3] Nanbo Jin, Rahmat-Samii, Y., "Advances in Particle Swarm Optimization for Antenna Designs: Real-Number, Binary, Single-Objective and Multiobjective Implementa- tions", IEEE Transactions on Antennas and Propagation, Volume 55, Issue 3, Part 1,2007, pp.556-567.

[4] Vlachogiannis, J.G., Lee, K.Y., "A Comparative Study on Particle Swarm Optimization for Optimal Steady-State Performance of Power Systems", IEEE Transactions on Power Systems, Volume 21, Issue 4, 2006, pp.1718 – 1728.

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

[6] Lü Zhensu, Hou Zhurong, "Particle swarm optimiza- tion with adaptive mutation", ACTA ELECTRONICA SINICA, Vol.32, No.3, 2004, pp.416-420.

[7] Lam, H.K., Wing-Kuen Ling, Lu, H.H.-C., Ling, S.S.H.,"Synchronization of Chaotic Systems Using Time-Delayed Fuzzy State-Feedback Controller", IEEE Transactions on Circuits and Systems I: Regular Papers, Volume 55, Issue 3, 2008, pp.893 – 903.

[8] Stefan Janson, Martin Middendorf, "A Hierarchical Particle Swarm Optimizer and Its Adaptive Variant", IEEE Transactions on SYSTEMS, Volume 35, Issue 6, 2005, pp. 1272-1282.

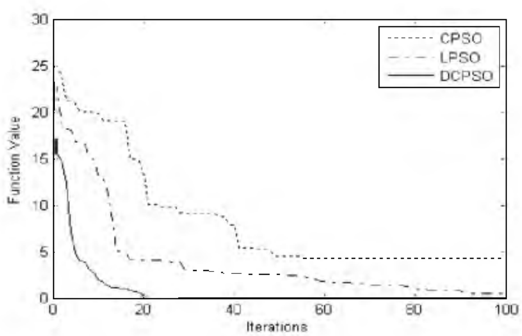

Figure 1. The mean function value of Sphere function versus the number of iterations

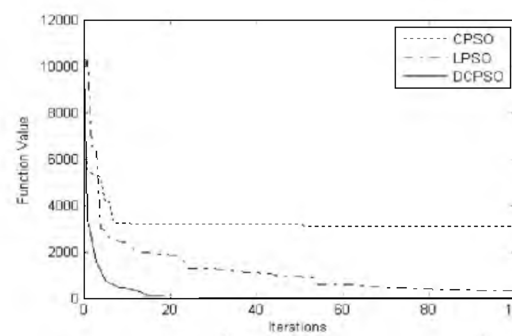

Figure 2. The mean function value of Rosenbrock function versus the number of iterations

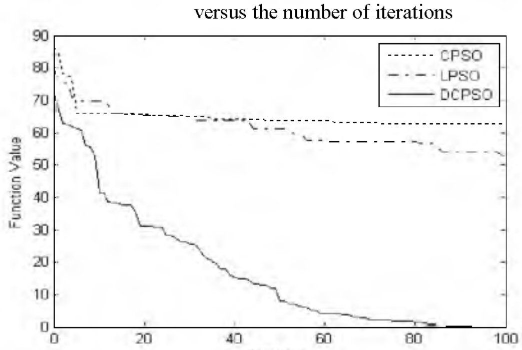

Figure 3. The mean function value of Rastrigrin function versus the number of iterations

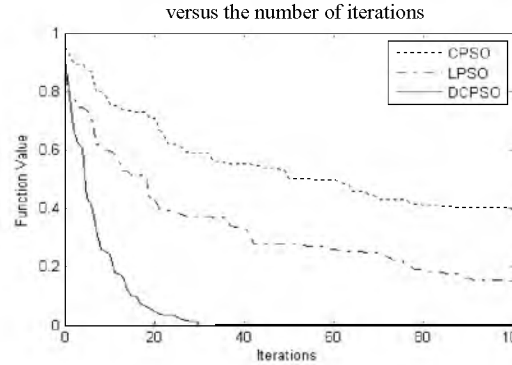

Figure 4. The mean function value of Griewank function versus the number of iterations