TELKOMNIKA, Vol.13, No.1, March 2015, pp. 331~340
ISSN: 1693-6930, accredited A by DIKTI, Decree No: 58/DIKTI/Kep/2013
DOI: 10.12928/TELKOMNIKA.v13i1.1267

A Self-Adaptive Chaos Particle Swarm Optimization Algorithm

Yalin Wu*, Shuiping Zhang
Faculty of Information Engineering, Jiangxi University of Science and Technology,
Ganzhou, Jiangxi, China
*Corresponding author, e-mail: 1159376855 @qq.com

Abstract

As a new evolutionary algorithm, particle swarm optimization (PSO) achieves integrated evolution through the information between the individuals. All the particles have the ability to adjust their own speed and remember the optimal positions they have experienced. This algorithm has solved many practical engineering problems and achieved better optimization effect. However, PSO can easily get trapped in local extremum, making it fail to get the global optimal solution and reducing its convergence speed. To settle these deficiencies, this paper has proposed an adaptive chaos particle swarm optimization (ACPSO) based on the idea of chaos optimization after analyzing the basic principles of PSO. This algorithm can improve the population diversity and the ergodicity of particle search through the property of chaos; adjust the inertia weight according to the premature convergence of the population and the individual fitness; consider the global optimization and local optimization; effectively avoid premature convergence and improve algorithm efficiency. The experimental simulation has verified its effectiveness and superiority.

Keywords: *chaotic theory, particle swarm optimization, self-adaption*

1. Introduction

PSO is a new evolutionary algorithm developed in recent years and as a novel swarm intelligent search technique, it has attracted the attention of a great number of researchers who have applied it in every field actively because of the features of simple computation form, few parameter setting and quick search speed since it was proposed. As a kind of evolutionary algorithm, PSO starts from a random solution; searches the optimal solutionvia iterations and it evaluates the solution quality through fitness, but it is simpler than the rules of genetic algorithm (GA). It doesn’t have theoperationsof “crossover” and “mutation”of GA and it searches the global optimum by following the current optimal value [1]. This kind of algorithm has drawn the attention of the academic circle for the advantages of ease to realize, high accuracy and fast convergence and it has demonstrated its superiority in solving practical problems.

In practical applications, since the initialization of the basic PSO is random, it has certain blindness. Although the random initialization can basically guarantee the uniform distribution of the initial population, it can’t guarantee the quality of every particle and it may cause some particles get far away from the optimal solution and affect the convergence speed of the algorithm [2]. PSO can easily get trapped in local extremum and it can’t have global optimal solution. The convergence speed of PSO is quite slow. It usually takes some time to reach the corresponding accuracy in solving practical problems. Sometimes it is not worthy to spend a long time to get a feasible solution[3]. The reason to cause this problem is that PSO doesn’t make full advantage of the information from the computation; instead, it only uses the information of global optimum and individual optimum in every iteration. Besides, the algorithm itself doesn’t have an optimization mechanism to eliminate the bad candidate solution so that it has a slow convergence speed [4],[5].

To solve the above-mentioned shortcomings of basic PSO, this paper has proposed adaptive chaos particle swarm optimization (ACPSO) based on the theory of chaos optimization. Generally, chaos refers to a random motion state to get from a deterministic formula and it is a common phenomenon in non-linear system. It is complicated and it has the characteristic similar to randomness. However, the seemingly chaotic chaos change is not totally chaos, on the contrary, it has certain inherent law. ACPSO performs chaotic initialization to the basic particle swarm with the unique ergodicity of chaos optimization; generates a series of initial solution groups and select the initial population from them, in this way, it effectively

Received October 18, 2014; Revised December 21, 2014; Accepted January 10, 2015

improves the diversity of PSO and the ergodicity of particle search.The inertia weight in PSO adjusts itself according to the fitness value of every particle individual so as to enhance its “global coarse search” and “local coarse search” capacities and improve its convergence speed and accuracy. Determine the good or bad of the current particle according to the fitness value and conduct chaotic operation on some selected particles to help “inertia”particle jump out from the local solution region and quickly search the global optimal solution. The last part of this paper has verified the effectiveness and advancement of the algorithm proposed in this paper through experiment simulation.

2. Overview of Standard Particle Swarm Optimization

The initial purpose of Doctor Ebethart and Doctor Kennedy was to establish a model in a 2D space to schematize the motion of the bird flock. Assuming such a scenario that a group of birds randomly search food in a region where there is only a piece of food. However, all of the birds don’t know where the food is. All they know is how far they are from the food. Then what is the optimal strategy to find the food? The simplest and the most effective strategy is the search the surrounding area of the bird which is the closest to the food. PSO is inspired from this kind of model and it is used to settle optimization problems. The bird has been abstracted as the particle with no quality and volume and it has been extended to *N*-dimensional space where the position of particle i is expressed as vector $X_i = \left(x_{i1}, x_{i2}, \ldots, x_{iN}\right)$and its flying speed as vector $V_i = (V_{i1}, V_{i2}, \ldots, \; V_{iN})$. Every particle has a fitness value determined by the objective function and it knows the current optimal position $\left(p_i\right)$and the current position, which can be seen as its own flying experience. In addition, every particle also knows the optimal position $\left(p_g\right)$(p_g is the optimal value of p_i) all particles in the group has found. The particle decides its next motion according to its own experience and the optimal experience of its companions.

For the kth iteration, every particle in the PSO changes according to the formula below:

$$v_{id}^{k+1} = v_{id}^{k} + c_1 rand\left(\;\;\right) \times \left(p_{id} - x_{id}^{k}\right) + c_2 rand\left(\;\;\right) \times \left(p_{gd} - x_{id}^{k}\right) \tag{1}$$

$$x_{id}^{k+1} = x_{id}^{k} + v_{id}^{k+1} \tag{2}$$

In Formulas (1) and (2), $i = 1, 2, \ldots, M, M$ is the total particles in the group. v_{id}^{k} is the dth sub-vector of the position vector of the iteration particle i in the kth iteration. p_{id} is $thedth$ component of the optimal position p_i of the particle i. p_{gd} is the dth component of the optimal position p_g in the group. $c_1 \& c_2$ are weight factors and $rand()$ is the random function to generate a random number within the range of (0,1). Formula (1) mainly computes the new speed of particle i through 3 parts: the speed of particle i at a former moment; the distance between the current position of particle i and its optimal position as well as the distance between the current position of particle i and the optimal position of the group. Particle i computes the coordinate of its new position through Formula (2). Particle i decides its next motion position through Formulas (1) and (2). From the angle of sociology, the first part of Formula (1) is called memory item, standing for the influence of the last speed and direction, the second part is the self cognition item, a vector pointing from the current point to the optimal point of the particle, representing that the particle motion comes from its own experience and the third part is called group cognition item, namely a vector from the current point to the optimal point of the group, reflecting the coordination and information sharing between the particles. Therefore, the particle determines its next motion through its own experience and the best experience of its partners [6],[7]. In Figure 1, the example of 2D space describes the principle that the particle moves from position X^{K} to X^{K+1} according to Formulas (1) and (2).

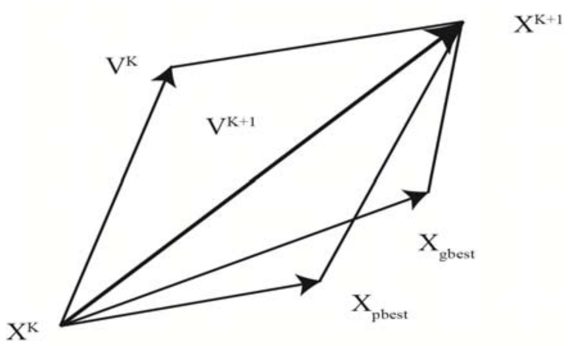

Figure 1. Schematic diagram of particle motion

In 1998, Shi and other people revised Formula (1) and introduced inertia weight factor w.

$$v_{id}^{k+1} = w \times v_{id}^{k} + c_1 rand(\ ) \times \left(p_{id} - x_{id}^{k}\right) + c_2 rand(\ ) \times \left(p_{gd} - x_{ik}^{k}\right) \tag{3}$$

w, non-negative, is called inertia factor. The bigger w is, the stronger the global optimization ability is and the local optimization ability is, if w is small, and it is good for local search. Initially, Shi took w as constant, but in the later experiment, it can be found that the dynamic w can obtain an optimization result better than the fixed value. Dynamic w can change linearly in the search process of PSO and it can also change according to certain measure function dynamically of the performance of PSO. At present, the linearly decreasing weight strategy (LDW) proposed by Shi has been used frequently, namely

$$w^k = w_{\min} + \frac{\left(w_{\max} - w_{\min}\right)}{N} \times \left(N - k\right) \tag{4}$$

In this formula, N is the maximum iterations and $w_{\max}$ & $w_{\min}$ are the maximum and minimum inertia weights respectively.

The introduction of inertia weight factor can greatly improve the performance of PSO and it can also adjust the global and local search capacities according to different search problems, making PSO solve many practical problems. Formulas (2) and (3) are called standard PSO [8].

3. Chaos Optimization Algorithm and Its Ergodic Property

In non-linear system, chaos is a common motion phenomenon with such excellent characteristics as ergodicity, randomness and “regularity”.Chaotic motion can experience all the states in the state space without repetition according to certain “rule” within certain motion range. Even a tiny change in the initial value can cause huge changes in the post motion, which is called the strong sensitivity of the initial value. The researches have used these features of chaos and proposed chaos optimization algorithm, which can easily jump out of the local solution region and which can have high computation efficiency [9].

The most typical chaotic system is sought from Logistic formula and its iterative formula is as follows:

$$x_{n+1} = S\left(x_n\right) = \mu\left(1 - x_n\right), n = 0, 1\cdots, N, 0 < x_0 < 1, \mu \in \left[0, 4\right] \tag{5}$$

When $\mu = 4$, the orbit $x_n\left(n = 0, 1, ..., N\right)$ is chaotic. It can be seen from Fig.2 that the two orbit points close to each other in the initial state diverge after only 3 iterations. After several iterations, since the errors of these two orbits increase continuously, it is difficult to identify which state the system is in and Logistic map is totally in the chaotic state [10].

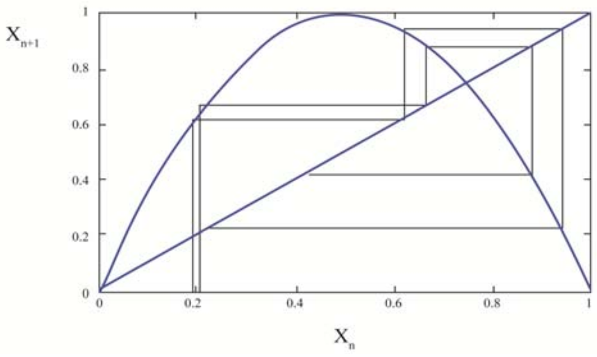

Figure 2. The logistic map when $\mu = 4$

Randomly take an initial point $x_0 \in (0,1)$, the orbit of Logistic map is chaotic among (0,1), that is to say, regardless of N formerly known iterative points, the position of the $(N+1)\mathrm{th}$ iterative point can't be predicted. In fact, for $x_0 \in (0,1)$, there is a sub-orbit $\{x_{nk}\}$ of $\{x_n\}$ to make $x_{nk} \to x$.

Although the chaotic dynamic system has a complicated "chaotic" state, it can be found that the certain chaos has regularity and ergodicity in statistics by observing this system with the perspective of statistic. As indicated in Figure 3, perform 1500 iterations with $x_{10} = 0.41795$ and $x_{20} = 0.81042$ as the initial values and get two chaotic sequences $\{x_{1n}\}$ and $\{x_{2n}\}$. A dot in the figure stands for a point in the space with its coordinate as $(x_{1j},\ x_{2j})$.

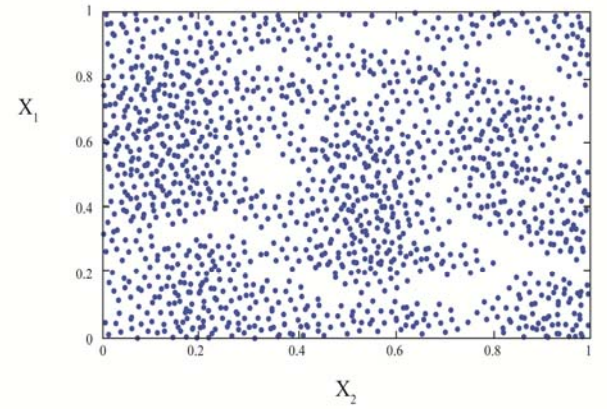

Figure 3. Ergodicity of logistic map

From the above figure, it is clear that chaotic variable is sensitive to the initial value, as evidenced by the fact that the chaotic trajectories of these two close initial values in the state space are completely different. Chaotic variable can experience all the states in the state space without repetition according to its own "rule". Chaotic variable is as chaotic as random variable. When $\mu = 4$, the total chaotic iterative formula map of Logistic moves unstably in the range of [0,1] and when it experiences a long-time dynamic motion, it will reflect the characteristic of randomness [11],[12]. Although chaotic variable is random, with the initial value determined, its chaotic variable has also been determined and the highlighted randomness is the inherent regularity of chaotic motion. The specific steps of chaos optimization algorithm can be indicated by Figure 4:

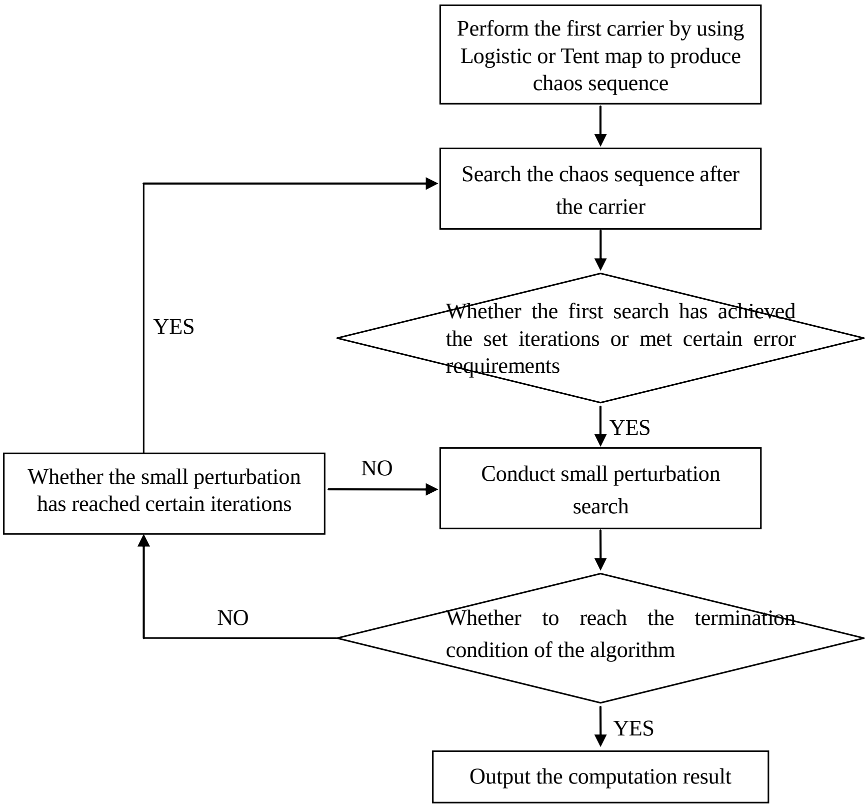
Perform the first carrier by using Logistic or Tent map to produce chaos sequence
Search the chaos sequence after the carrier
Whether the first search has achieved the set iterations or met certain error requirements
YES
YES
Conduct small perturbation search
NO
Whether the small perturbation has reached certain iterations
Whether to reach the termination condition of the algorithm
NO
YES
Output the computation result

Figure 4. Procedures of chaos optimization algorithm

4. Adaptive Chaos Particle Swarm Optimization & Example Test

Because PSO uses the randomly-generated particle, it may have unsearched dead zones for multimodal function. Since chaos optimization algorithm has the excellent feature of ergodicity, this paper initializes by using chaos optimization in the early phase of the optimization of the particle swarm and selects the initial particle population. The specific process includes two parts: the 1st part is to perform global search with the basic particle swarm optimization while the 2nd part is to implement local search according to the result of PSO by using chaos optimization.

4.1. Chaotic Local Search

Chaotic local search (CLS) is mainly to improve the search performance of the particle and avoid the particle to get trapped in local extremum. Take Logistic map as example. CLS firsts maps the particle variable into chaotic variable and ten transforms the chaotic variable into particle variable after the iteration. The basic formulas of these two transformations are as follows:

$$\phi_i^{(k)} = \frac{x_i^{(k)} - x_{\min,i}}{x_{\max,i} - x_{\min,i}} \tag{6}$$

$$x_i^{(k+1)} = x_{\min,i} + \phi_i^{(k+1)} \left(x_{\max,i} - x_{\min,x} \right) \tag{7}$$

Here, $x_{min,i}$ and $x_{max,i}$ represent the upper and lower bounds of the ith partical respectively. $x_i^{(k)}$ is the decision variable (namely the individual extremum of the particle swarm) within the range of $\left(x_{min,i,} x_{max,i} \right)$ while $\phi_i^{(k)}$ is the chaotic variable with its value in the range of (0,1).

The specific steps of CLS are as follows:

(1) Assume that $k = 0$ and transform $x_i^{(k)}$ into $\phi_i^{(k)}$ through Formula (6).

(2) According to the current $\phi_i^{(k)}$, determine the chaotic variable $\phi_i^{(k+1)}$ of the next iteration.

(3) According to Formula (7), transform $\phi_i^{(k+1)}$ into the decision variable $x_i^{(k+1)}$.

(4) Evaluate the new $x_i^{(k+1)}$.

(5) If the new solution is superior to the optimal solution before the local search of the particle swarm or it reaches the preset maximum iterations, output the solution which is found from chaotic local search; otherwise, make $k = k + 1$ and return to Step (2) [13],[14].

4.2. The Steps and Procedures of Adaptive Particle Swarm Optimization

By integrating the search process of the two phases of chaos particle swarm optimization, the overall search steps of this algorithm are as follows:

(1) Set the size of the particle population *N* and the maximum number of iterations and initialize the position and speed of the particle randomly within the feasible value range.

(2) Evaluate the fitness of every particle; set the objective fitness value of the 1st particle as the global optimal value and its initial position is its own individual extremum value.

(3) Update the particle speed and position according to Formulas (2) and (3).

(4) Evaluate the particle fitness; compare the particle fitness value with the previous value and update the superior objective function fitness value as the current individual extremum value; compare the current optimal particle fitness with the previous and update the superior objective fitness value as the current global optimal value.

(5) Keep the former *N*/5 particles of the population.

(6) Update these particle positions by using chaos local search and the CLS results. If it meets the termination standard, output the currently optimal solution.

(7) Narrow down the search space and randomly generate 4*N*/5 new particles in the narrowed search space.

(8) Form a new population with the particles through CLS update and the new particles of 4*N*/5.

(9) Make $k = k + 1$ and return to Step (3).

4.3. Algorithm Testing & Result Analysis

This paper has used four classical testing functions in the experiments so as to test the performance of ACPSO and it also compares the testing result of ACPSO with those of DE and PSO.

In order to investigate the algorithm expandability, the variable dimensions for every function to be used in the testing are 20 dimensions. In the experiment, operates every circumstance 30 times and seek the mean optimal value as the basis of performance comparison. The parameters selection in these experiments are for PSO and ACPSO, w reduces from 0.8 to 0.3 with the evolutionary algebra, the scaling factor and crossover factor of DE are 0.5 and 0.9 respectively.

(1) Function f_1 (Griewank Function)

$$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1,|x_i|\le 300$$

$$\min f(x^*)= f(0,0,\cdots 0)=0$$

Griewank function reaches its global minimal point when $x_{\text{i}}=0$ and it reaches the local minimal points when $x_i=\pm k\pi\sqrt{i}, i=1,2,\cdots,n, k=1,2,\cdots,n$. The computation results of these algorithms are demonstrated in Table 1.

Table 1. The Mean Fitness Value for Griewank Function

Algorithm	PSO	DE	ACPSO
Variable Dimension	n=20	n=20	n=20
Mean Optimal Value	3.36	4.52	0.81

When the variable dimension $n=20$, the changing curve of the mean fitness value for Griewank function with the iterations is as indicated in Figure 5.

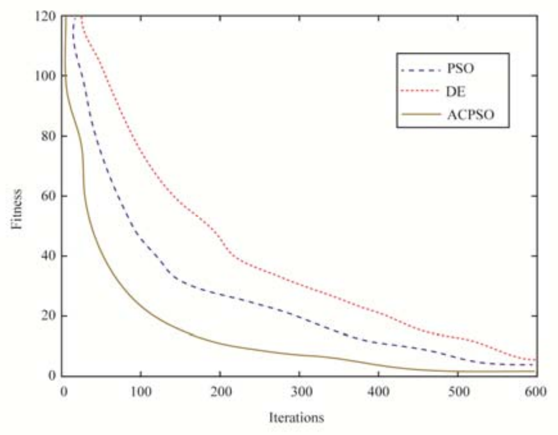

Figure 5. Iterations on Griewank function

(2) Function f_2 (Rastrigrin Function)

$$f(x)=\sum_{i=1}^{n}\left[x_i^2-10\cos(2\pi x_i)+10\right],|x_i|\le 5.12$$

$$\min f(x^*)= f(0,0,\cdots,0)=0$$

As a multimodal function, Rastrigrin function reaches the global minimal point when $x_{\text{i}}=0$ .There are about 10n local minimal points within $S=\{x_i\in[-5.12,5.12], i=1,2,\cdots,n\}$. See the computation results of these algorithms in Table 2.

Table 2. The Mean Fitness Value for Rastrigrin Function

Algorithm	PSO	DE	ACPSO
Variable Dimension	n=20	n=20	n=20
Mean Optimal Value	25.36	100.62	3.98

When the variable dimension $n = 20$, the changing curve of the mean fitness value for Rastrigrin function with the iterations is as indicated in Figure 6.

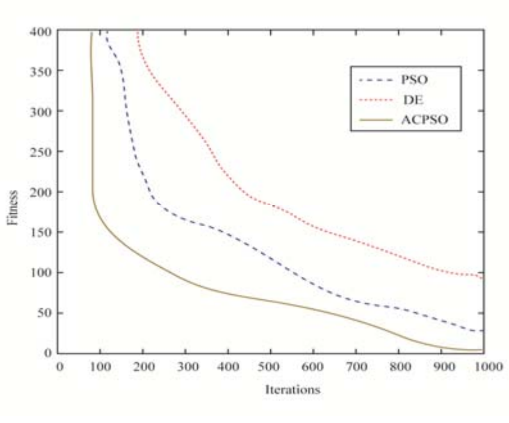

Figure 6. Iterations on Rastrigrin function

(3) Function f_3 (Ackley Function)

$$f\left(x\right) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos\left(2\pi x_i\right)\right) + 20 + e, \left|x_i\right| \le 32$$

$$\min f\left(x^*\right) = f\left(0,0,\cdots 0\right) = 0$$

Ackley function reaches the global minimal point when $x_i = 0$. The computation results of these algorithms are shown in Table 3.

Table 3. The Mean Fitness Value for Ackley Function

Algorithm	PSO	DE	ACPSO
Variable Dimension	n=20	n=20	n=20
Mean Optimal Value	1.18	2.03	1.21

When the variable dimension $n = 20$, the changing curve of the mean fitness value for Ackley function with the iterations is as indicated in Figure 7.

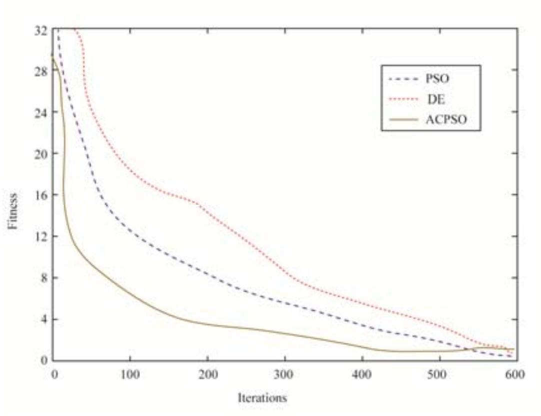

Figure 7. Iterations on Ackley function

(4) Function f_4 (Rosenbrock Function)

$$f(x)=\sum_{i=1}^{n}\left(100\left(x_{i+1}-x_i^2\right)^2+\left(x_i-1\right)^2\right),\left|x_i\right|\le 50$$

$$\min f\left(x*\right)=f\left(1,1,\cdots 1\right)=0$$

Rosenbrock is a non-convex pathological quadratic function and its minimal point is easy to find, but it is greatly difficult to converge to global minimum. The computation results of the several algorithms can be seen in Table 4.

Table 4. The Mean Fitness Value for Rosenbrock Function

Algorithm	PSO	DE	ACPSO
Variable Dimension	n=20	n=20	n=20
Mean Optimal Value	3.14	4.21	3.5e-3

When the variable dimension $n=20$, the changing curve of the mean fitness value for Rosenbrock function with the iterations is as indicated in Figure 8.

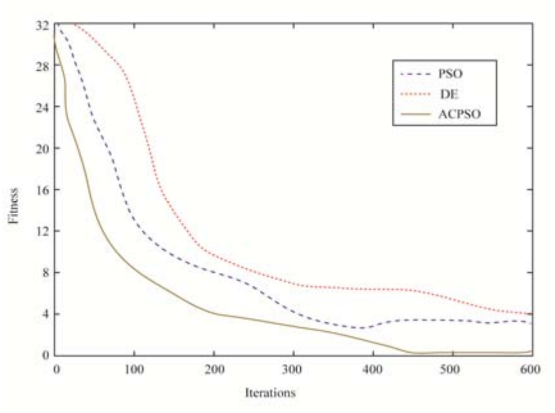

Figure 8. Iterations on Rosenbrock function

It can be seen from the above experimental test that ACPSO is superior to PSO and DE in all test function performance. It not only has fast convergence speed, but also has strong global search ability. In the four testing functions, the mean optimal fitness of ACPSO is the minimum, therefore, it has higher accuracy and stability than PSO and DE and its advantage is more obvious in the cases of higher dimensions and relatively sharp function value changes while the effect of PSO falls significantly. From the mean experimental result, it is clear that the mean fitness and standard deviation of DE are big, therefore, this algorithm has bigger fluctuations, the bigger the dimensions, the more unstable it is. It can be seen from the above experimental result that ACPSO is a suitable tool to solve the global optimization problems of complicated functions. For the high-dimensional and multi-extreme-point functions, the global minimum of certain accuracy can be obtained at fewer computation cost.

5. Conclusion

This paper has analyzed the constitution and optimization principle of particle swarm optimization and it has pointed out that the basic particle swarm optimization can easily get trapped in local optimal solution in the optimum iteration and that its convergence speed is very slow in the late stage since it generates the particles representing variable value randomly. Therefore, this paper integrates chaos optimization algorithm in the particle swarm optimization and proposes adaptive particle swarm optimization to solve objective optimization problems. This algorithms uses the chaos principle, enhances the diversity of variable value; computes the corresponding fitness value to every particle , namely the objective function value of the problem, through particle swarm optimization; selects variable value to perform chaos optimization according to its value in order to help the variable to jump out from the local extremal region and self-adaptively adjusts its inertia weight coefficient according to the objective function value of the problem to improve the global and local search capacities of the algorithm.

References

[1] Masoud JS, Ramin V, Mehrdad B, Hassan S, Aria A. Application of Particle Swarm Optimization in Chaos Synchronization in Noisy Environment in Presence of Unknown Parameter Uncertainty. *Communications in Nonlinear Science and Numerical Simulation*. 2012; 17(2): 742-753.

[2] Amir HG, Gun JY, Xin SY, Siamak T. Chaos-enhanced Accelerated Particle Swarm Optimization. *Communications in Nonlinear Science and Numerical Simulation.* 2013; 18(2): 327-340.

[3] Wei ZR, Jing C. A Chaos Particle Swarm Optimization Ranging Correction Location in Complex Environment. *The Journal of China Universities of Posts and Telecommunications.* 2012; 19(1): 1-5.

[4] Na D, Chun HW, Wai HI, Zeng QC, Kai LY. Chaotic Species Based Particle Swarm Optimization Algorithms and Its Application in PCB Components Detection. *Expert Systems with Applications*. 2012; 39(15): 12501-12511.

[5] Shijun Z, Ting J. A Novel Particle Swarm Optimization Trained Support Vector Machine for Automatic Sense-through-foliage Target Recognition System. *Knowledge-Based Systems.* 2014; 65: 50-59.

[6] MR Tanweer, S Suresh, N Sundararajan. Self Regulating Particle Swarm Optimization Algorithm. *Information Sciences*. 2015; 294(10): 182-202.

[7] Oguz ET, Mert ST, Mustafa TC. Chaotic Quantum Behaved Particle Swarm Optimization Algorithm for Solving Nonlinear System of Equations. *Computers & Mathematics with Applications.* 2014; 68(4): 508-530.

[8] Keiji T, TakeruIbuki, Tetsuzo T. A Chaotic Particle Swarm Optimization Exploiting A Virtual Quartic Objective Function Based on The Personal and Global Best Solutions. *Applied Mathematics and Computation.* 2013; 219(17): 8991-9011.

[9] Yaoyao H, Qifa X, Shanlin Y, Li L. Reservoir Flood Control Operation Based on Chaotic Particle Swarm Optimization Algorithm. *Applied Mathematical Modelling.* 2014; 38(17): 4480-4492.

[10] Shen X, Liu J, Liu Y. A New Particle Swarm Optimization Solution to Power Flow Regulation for Unsolvable Case. *Energy Procedia.* 2012; 14: 775-781.

[11] Wei FG, San YL, Ling LH. Particle Swarm Optimization with Chaotic Opposition-based Population Initialization and Stochastic Search Technique. *Communications in Nonlinear Science and Numerical Simulation*. 2012; 17(11): 4316-4327.

[12] Huimin J, CK Kwong, Zengqiang C, YC Ysim. Chaos Particle Swarm Optimization and Fuzzy Modeling Approaches to Constrained Predictive Control. *Expert Systems with Applications*. 2012; 39(1): 194-201.

[13] Omid MN, Anvar B, Paria J. Dynamic Diversity Enhancement in Particle Swarm Optimization (DDEPSO) Algorithm for Preventing from Premature Convergence. *Procedia Computer Science.* 2013; 24: 54-65.

[14] Mehdi S, Hassan S, Aria A. Minimum Entropy Control of Chaos Via Online Particle Swarm Optimization Method. *Applied Mathematical Modelling.* 2012; 36(8): 3931-3940.