International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 8, No. 4 (2015), pp. 41-48
http://dx.doi.org/10.14257/ijsip.2015.8.4.04

# A New Chaos Particle Swarm Optimization Combining the Chaotic Perturbation

[1]Li Mengxia and [2]Liao Ruiquan

*[1]School of Information and Mathematics, Yangtze University, Jingzhou Hubei 434023, China and Petroleum Engineering College, Yangtze University, Jingzhou Hubei 434023, China and The Branch of Key Laboratory of CNPC for Oil and Gas Production, Yangtze University, Jingzhou Hubei 434023, China*
*[2]The Branch of Key Laboratory of CNPC for Oil and Gas Production, Yangtze University, Jingzhou Hubei 434023, China*
*limengxia81@126.com, liaoruiquan@163.com*

### *Abstract*

*Aiming at the drawback of the local search ability of weakness of adaptive chaos particle swarm optimization (ACPSO), which is based on the variance of population's fitness, this paper presents that introduces the chaos mutation and chaos search to the ACPSO. By using the An chaos mapping to proceed the chaos mutation for some particles and take chaos search for the global optimal particle, it proposes a rule which takes into account the positions of particles and adaptive mutative scale of optimizing space. The results of numerical simulation show that the convergence, and the global and local search ability of the new method are improved, and can effectively avoid premature convergence.*

***Keywords:*** *chaos mapping, particle swarm optimization (PSO), fitness variance, convergent percentage*

## 1. Introduction

Because the particle swarm optimization has many advantages as followed, it is simple and easy to implement. It can carry out the parallel processing, and has good robustness. It can converge to the global optimal solution with greater probability. PSO has been widely concerned and successfully applied to several fields [1] which are constrained optimization, dynamics process optimization and traffic control.

Being similar to the genetic algorithm, there also exists the phenomenon of premature convergence for PSO. The premature convergence performs particularly obviously in the more complex high dimensional and multimodal search problems. People had given several improved forms [2-3]. One of these forms is combined with chaotic system, and disturb on the particle using the pseudo-randomness and ergodicity of the chaotic system to improve the probability and velocity of the global convergence [4]. These forms are all using the Logistic chaos mapping to produce the chaotic sequence [4-5], but the uniformity of chaotic sequence is poor and it influences the superiority of the algorithm.

It discusses the CAPSO based on the group fitness variance in the reference [6] which is constructed based on An chaos mapping. It initializes the positions and velocity of particle swarm using An chaos mapping. It is similar to the reference [7]. It proceeds the chaos update using the changes in the fitness variance which adaptively controls parts of particles. Its optimal performance is better than the chaos particle swarm optimization with the similar structure which is based on the Logistic chaos mapping. The reason is that the chaos sequence which is produced by An chaos mapping is superior to the one which is produced by Logistic chaos mapping.

ISSN: 2005-4254 IJSIP

But the local search ability of the algorithm performs big dent in the final stage of the iteration just as this paper stated. In order to improve the performance of the algorithm, this paper introduces two types of modification. The first one is that relaxes the triggering condition of chaos mapping and performs the chaotic mutation to the non-global optimal particles at a certain probability. The second one is that performs chaos search to the global optimal particles. This paper uses the pseudo-randomness and ergodicity of the chaos system to prove the global and local optimizing abilities.

## 2. An Chaos Mapping

The behavior of chaotic motion is complex and similar to the random, but it has the intrinsic regularity. The chaotic motion has pseudo-randomness, ergodicity, regularity, and so on. It is between the definite phenomena and completely random phenomenon.

The recursive formula [8] of random number generation of chaos mapping which is introduced by An is:

$$y_{n+1} = \begin{cases} \frac{3}{2} y_n + \frac{1}{4} & 0 \le y_n < \frac{1}{2} \\ \frac{1}{2} y_n - \frac{1}{4} & \frac{1}{2} \le y_n < 1 \end{cases} \quad (n = 0,1,2,\cdots) \tag{1}$$

This recursive formula can generate the sequence with endless cycle, its limit distribution of empirical distribution is:

$$F(y) = (Ln(y + 1/2) + \ln(2))/\ln(3) \tag{2}$$

From the following expression:

$$cx_i = (Ln(y_i + 1/2) + \ln(2))/\ln(3) \tag{3}$$

We can consider the sequence obtained as the random number sequence which is the uniform distribution over the interval [0, 1].

## 3 The Modified PSO

Consider the following global optimal model:

$$min \quad f(x) = f(x_1, x_2, \cdots, x_D) \tag{4}$$

Where, $D$ is the dimension of the variable $x$ . This paper uses the real number encoding.

### 3.1. Standard PSO

The iterative formula of the velocity and position of the standard PSO is:

$$v(t+1) = w \times v(t) + c_1 \times rand1 \times (pbest - x(t)) + +c_2 \times rand2 \times (gbest - x(t)) \tag{5}$$

$$x(t+1) = x(t) + v(t+1) \tag{6}$$

In the above two formulas, $v(t)$ and $x(t)$ are the particle velocity and position of the t-th iteration respectively. The rand1 and rand2 are the uniform random numbers over the interval [0, 1]. $c_1$ and $c_2$ are learning factors with the value 2. $pbest$ is the individual extreme position, that is the optimal solution which the particle had ever experienced itself. $gbest$ is the global extreme position, that is the optimal solution which the particle swarm had ever experienced. $w$ is the inertia weight whose value is between 0.1 and 0.9 [8]. Reference [8] indicates that if $w$ decreases linearly as the iterations increase, the convergence of the algorithm can be improved dramatically. That is

$$w = w_{max} - t \times (w_{max} - w_{min}) / MaxDT \tag{7}$$

Where, $w_{max}$ and $w_{min}$ are the maximum and minimum of the weight respectively. $MaxDT$ is the maximum iterations, and $t$ is the current iterations.

 

Restrict the range of the velocity value and position value. Generally, if $|x| \le x_{max}$, take $v_{max} = k \cdot x_{max}$, where $0.1 \le k \le 1.0$ [9].

### 3.2. Initial Population Produced by Chaos Mapping

Being same to the reference [6], there produces a $D$ -dimensional vector $cx_1$. The value of each element is between 0 and 1. Uses An chaos mapping to produce $N-1$ s $D$ -dimensional points with $N-1$ iterations to each element of $cx_1$, and denote these points as $cx_2, cx_3, \cdots, cx_N$. Convert the formula (9) to the optimization space, and consider it as the initial population.

As to the variable $x$ which is not in the interval $(0,1)$, we suppose $x \in (a,b)$. We can convert it through the following expressions:

$$cx = (x-a)/(b-a) \quad (8)$$

$$x = a + cx(b-a) \quad (9)$$

### 3.3. Improvement of the Iterative Update

(1) Chaotic Mutation

In order to improve the global search capability, but have strong local search capability in the later stage, the interval length of the chaotic mutation decreases as the increase of the iterations. Denote the length of the corresponding interval of the mutation in the $t$ -th iteration as $CD(t)$, and denote $k_t = \frac{CD(t)}{CD(t-1)}$ and $CD(0) = ub - lb$. Might as well take $k_t = \frac{MaxDT - t + 1}{MaxDT}$, where $t$ is the current iterations, $MaxDT$ is the maximum iterations, $ub$ is the upper limit of the initial search space, and $lb$ is the lower limit of the initial search space. Then

$$CD(t) = k_1 \times k_2 \times \cdots \times k_t \times CD(0) \quad (10)$$

The mutation interval of the particle $x(t)$ is:

$$lb(t) = (1 - k_1 k_2 \cdots k_t) \times x(t) + k_1 k_2 \cdots k_t \times lb \quad (11)$$

$$ub(t) = (1 - k_1 k_2 \cdots k_t) \times x(t) + k_1 k_2 \cdots k_t \times ub \quad (12)$$

Produce the uniform random number $p$ over the interval $[0,1]$ for the particle $x(t)$. If $p > 0.5$, mutate the particle according to the formulas (8), (1), (3), and (9),and the mutation points locate in the interval $[lb(t+1), ub(t+1)]$. If the fitness value becomes small we except the mutation, or refuse it. If $p \le 0.5$, the particle doesn't be mutated.

(2) Chaos Search

To the global extreme value $gbest$, repeatedly use the formulas (8), (1), (3), and (9) to perform the chaos iteration. If the fitness value doesn't become small until the chaos iterations obtains the setting upper limit, stop the chaos search. If when the fitness value becomes small, stop the chaos search and update $gbest$.

The interval of chaos search shrinks gradually as the preceding formulas from (10) to (12).

(3) Chaos Disturbance

Similar to the reference [6], this paper determines that whether the premature convergence occurs or not using the changes of $\sigma^2$ which are produced in the adjacent iterations. If the difference of the two $\sigma^2$ which are produced in the adjacent iterations is less than some given fixed value such as $eps = 10^{-6}$, it considers that the current particle swarm is needed to perform the perturbation. Firstly, determine the particle number $s$ which is needed to be replaced in the current particle swarm. $s$ is about 61.8% of the total particle number. Start with the particle position which is produced randomly, and use An chaos mapping to produce $s-1$ s particle positions. Use the $s$ s particle positions to replace the $s$ s particle positions which have the worst fitness among the current particle swarm. And then perform the PSO iteration.

### 3.4. Terminal Condition of Iteration

Similar to the reference [6], be able to use the given maximum iterations as the terminal condition. When test the efficiency of algorithm using the standard functions, we can use the fitness values to obtain the standard convergence to be the convergence criterion. These papers use the maximum iterations as the terminal condition.

### 3.5. The Entire Process of the Improved PSO

Step 1 Initialize the inertia weight $w_{\max}$ and $w_{\min}$, the learning factor $c_1$ and $c_2$, the group size $N$, the maximum iterations $MaxDT$, the problem dimension $D$, and the accuracy control $eps = 10^{-6}$. Give the optimal space $[lb, ub]$ and the speed limit $v_{\max}$. Give the maximum iterations of chaos search $HDT$.

Step 2 Randomly produce a $D$-dimensional particle over the interval $[0,1)$. According to the statement of the section 2.2, get $N$ s $D$-dimensional particles which are denoted by $x_i, i = 1, 2, \cdots, N$. A random $D$-dimensional space vector is selected from $[0,1)$. Use the expressions (1) and (3) of An chaos mapping to obtain $N-1$ s particles, and obtain the $N$ s $D$-dimensional vector which are the initial particle speed. Let the iterations be 0, and turn to step 3.

Step 3 Substitute $x_i$ into the objective function to calculate the fitness $f_i$, and determine the global optimal positions of the particle swarm $gbest$, the experienced optimal positions of articles $pbest_i$ $\cdot i = 1, 2, \cdots, N$. Turn to step 4.

Step 4 $w$ decreases according to the expression (7). The positions and speed of the particles are updated according to the expression (5) and (6). The iterations increase by 1. Take $k_1 = \frac{MaxDT - t + 1}{MaxDT}$, and determine the chaotic mutation and the chaos search interval $[lb(t), ub(t)]$ according to the expressions (10) to (12). Perform chaotic mutation to update $gbest$ and $pbest_i$. Perform chaotic search to $gbest$, and update $gbest$ and $pbest_i$. Compute the fitness variance of the current particle swarm. If the absolute value of the difference between the fitness variance of the current particle swarm and the one of the pre-iteration is less than $eps$, turn to step 5, or turn to step 6.

Step 5 Compute the numbers of the particles $s$ which is needed to replace. Produce $s$ s new particles using the manipulation which is similar to produce the initial particle positions. Use the new particles to replace the current $s$ s particles which have the worst fitness. And then turn to step 4.

Step 6 If the iterations are less than $MaxDT$, turn to step 4. Otherwise, turn to step 7.

Step 7 Output the final results: $gbest$ and $fbest$.

## 4. Numerical Simulation

In order to test the properties of the improved algorithm and compare with the results of the numerical simulation of the reference [6] and [9], choose five nonlinear standard functions which show in the table 1. Where, $f_1$ and $f_2$ are unimodal high-dimensional functions. $f_3$ and $f_4$ are multimodal high-dimensional functions.

**Table 1. Standard Test Functions**

| function | dimension | domain | Convergence criteria | optimization |
|---|---|---|---|---|
| $f_1 = \sum_{i=1}^{n} x_i^2$ | 30 | $[-100,100]^n$ | E-6 | 0 |
| $f_2 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | 30 | $[-30,30]^n$ | 100 | 0 |
| $f_3 = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 30 | $[-5.12,5.12]^n$ | 100 | 0 |
| $f_4 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}}) + 1$ | 30 | $[-600,600]^n$ | E-6 | 0 |
| $f_5 = \frac{\sin^2\sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2} + 0.5$ | 2 | $[-100,100]^n$ | 0 | 0 |

Indicated as the reference [9], the theoretical optimal solutions of part test functions are difficult to obtain. So give the convergence criterion. In practice, determine the convergence by the convergence criterion. The convergence criterion in the table 1 and the one in the reference [9] are consistent.

According to the reference [9], the learning factors $c_1 = c_2 = 2$, dimension of the variables, domains, convergence criterions are showed in the table 1. The inertia weight $w$ decreases linearly from 0.9 to 0.2. According to the reference [6], replace the 61.8% of the particles using the chaos, $eps = 10^{-6}$, and the maximum iterations is $MaxDT = 10000$. But the iterations of this paper are $MaxDT = 1000$, and the iterations of chaotic search are 100. It is observed that the total iterations of this paper is less than the ones of the reference [6]. Take the particle swarm size 100. Randomly run 20 times. In order to compare the search efficiency of the algorithm, adopt the criterion of the reference [9] as below:

1) Denote the optimal average value $mB$ on the premise that search successfully.

2) Denote the rate of successful search $Ir$. Show the results in the Table 2. The data of the reference [9] are quoted from the reference itself. The data of the reference [6] and this paper are obtained from the programming.

**Table 2. The Results of Search of Functions**

| function | source | $mB$ | $Ir$ |
|---|---|---|---|
| $f_1$ | reference [6] | 2.58E-16 | 1.0 |
| | reference [9] | 0 | 1.0 |
| | this paper | 3.92e-29 | 1.0 |
| $f_2$ | reference [6] | 35.0677 | 1.0 |
| | reference [9] | 29.3308 | 0.83 |
| | this paper | 26.7778 | 1.0 |
| $f_3$ | reference [6] | 38.1145 | 0.9 |
| | reference [9] | 19.4780 | 0.99 |
| | this paper | 0 | 1.0 |
| $f_4$ | reference [6] | 3.96E-14 | 1.0 |
| | reference [9] | 0 | 1.0 |
| | this paper | 0 | 1.0 |
| $f_5$ | reference [6] | 0 | 0.85 |
| | reference [9] | 0 | 1.0 |
| | this paper | 0 | 0.95 |

To the high-dimensional unimodal functions $f_1$ and $f_2$, the rate of convergence of this paper is 1.0, and the average optimization is also the optimization. Because the uniformity of the sequence which is produced by An chaos mapping in this paper is good and benefits to jump out of the local minimum. And chaotic search strengthens the utilization of the chaotic characteristic. To the multimodal high-dimensional functions $f_3$, the convergence rate of this algorithm is 1, and the average optimization is theoretical optimization. The results of this paper are superior to the existed results. To $f_4$, the algorithm in this paper directly obtained the theoretical optimization with the convergence rate 1, and achieved the effect in the reference [9].

To $f_5$, the rate which the algorithm converges to 0 in this paper is 0.95 which is larger than the rate in the reference [6] and slightly less than the one in the reference [9]. The main reason is that the ratio of the area of the valley shape which contains the global optimal points to the one of the search range is less than 0.000256. When use the algorithm to perform 20 times optimization to the function $f_5$, only obtains two optimal results with 0 and 0.0097159. The two optimizations are very close. It shows that improving the local search ability is needed.

Run 20 times to each test function, and iterate 1000 times in each running. 1000 iterations correspond to 1000 global optimizations. Consider the logarithm of the average value of the 20s results with the base 10 as the y-coordinate, and the iterations as the x-coordinate. The speeds of convergence of the standard PSO, ACPSO in the reference [6] and MACPSO in this paper are showed in the Figure 1.

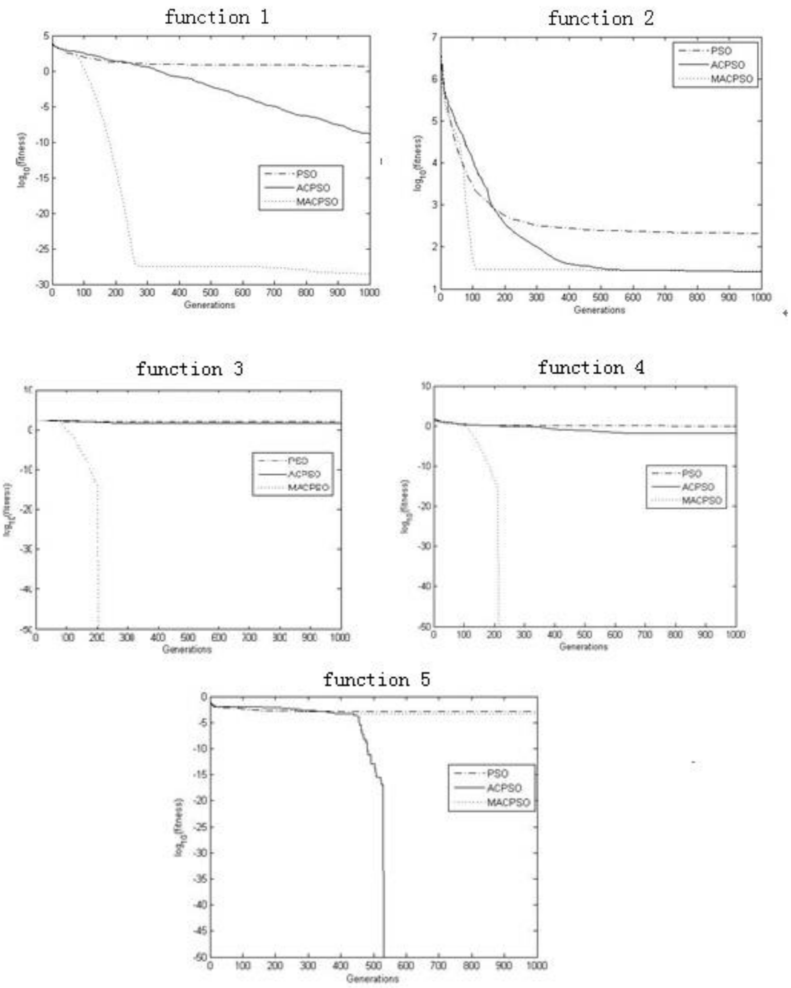


**Figure 1. Comparison between the Average Global Optimization and the Iterations**

## 4. Conclusion

In this paper, to the above standard functions, initialize the particles by introducing An chaos mapping, and use the Changes of the fitness variance to control the chaotic update of parts of particles. Perform chaotic mutation to the non-global optimal particles, and do chaotic search to the global optimal particles. Dynamically justify the range of mutation and search. This paper gives the good constringency effect.

## Acknowledgements

The authors will thank people in the Branch of Key Laboratory of CNPC for Oil and Gas Production for their great help.

This paper is supported by National Natural Science Foundation of China (61170031 and 60873021). It is also supported by Research Fund for the Doctoral Program of Higher Education of China (20114220110001).

## References

[1] R. Zi-Hui and W. Jian, Journal of Computer Applications, vol. 28, **(2008)**, pp. 2652-2654.
[2] Y. del Valle, G. K. Venayagamoorthy, S. Mohaghegh, J.-C. Hernandez and R. G. Harley, IEEE Transactions on Evolution Computation, vol. 12, **(2008)**, pp. 171-195.
[3] K. Kameyama, IEICE Transactions on Information and Systems. E92D, **(2009)**, pp. 1354-1361.

[4] F. Bin, W. Zhang and S. Jun, Computer Application and Software, vol. 26, **(2009)**, pp. 50-52.
[5] L. Huaying, L. Yu'e and Z. Junshi, "Computer Engineering and Applications, vol. 42, **(2006)**, pp. 77-79.
[6] D. Yong and G. Hai-Min, Application Research of Computers, vol. 28, **(2011)**, pp. 854-856.
[7] L. V. Zhen-Su and H. Zhi-Rong, ACTA ELECTRONICA SINICA, vol. 32, **(2004)**, pp. 416-420.
[8] Y. Shi and R. C. Eberhart, "A modified swarm optimizer", Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, IEEE Press, **(1998)** May, Anchorage, Alaska.
[9] L. Rong-Jun and C. Xian-Ying, Computer Engineering, vol. 35, **(2009)**, pp. 192-194.

## Authors

**Li Mengxia**, She received his M.Sc. in Mathematics (2007) from Yangtzeu University. Now she is a doctoral student majoring in Oil and gas field development engineering, Yangtzeu University. Since 2007 she is a teacher of Yangtzeu University. His current research interests include Oil and gas field development and optimization algorithm.

**Liao Ruiquan**, He received his M.Sc. in oil-gas field development engineering (1994) from Southwest Petroleum University. And he received his PhD in control theory and control engineering (2002) from HuaZhong University of Science and Technology. Now he is full professor of petroleum engineering college, Yangtzeu University. Since 2008 he is the doctoral tutor. His current research interests include process theory and technology of oil and gas exploitation, multiphase pipe flow, and systems engineering.

 