The Open Cybernetics & Systemics Journal, 2014, *8*, 1252-1256

Particle Swarm Optimization Algorithm with Chaotic Mapping Model

Songhao Jia[1], Cai Yang[1], Yan Tian[1], Changwang Liu[2,*] and Yihua Lan[1]

[1]*School of Computer and Information Technology, Nanyang Normal University, Nanyang, 473061, China;* [2]*School of Software, Nanyang Normal University, Nanyang, 473061, China*

Abstract: Particle swarm optimization algorithm is easy to reach premature convergence in the solution process, and fall into the local optimal solution. Aiming at the problem, this paper proposes a particle swarm optimization algorithm with chaotic mapping (CM-PSO). The algorithms uses chaotic mapping function to optimize the initial state of population, improve the probability of obtain optimal solution. Then, CM-PSO algorithm introduces nonlinear decreasing strategy on the inertia weight to avoid local optimal solution. In the experimental stage, four different functions are used to validate the performance of the algorithm. The experimental results show that, compared with the standard particle swarm algorithm, CM-PSO algorithm has strong global searching ability, can effectively avoid the premature convergence problem, and enhance the ability of the algorithm to escape from local optima. Although the algorithm consumes time is slightly increased, it is worth for getting the global optimal solution with such cost.

Keywords: Chaotic mapping, chebyshev, inertia weight, initial state of population, particle swarm optimization algorithm.

1. INTRODUCTION

Particle swarm optimization algorithm (PSO) is a global optimization algorithm proposed in 1995 by J.Kennedy and R.C. Eberhart, used for complex optimization problem of nonlinear and multi peak. Because the PSO algorithm operation and implementation is simple and easy to realize, it is widely used in system identification, neural network and other fields. The algorithm has good operability and optimization performance has been verified.

In the application process, PSO algorithm also has some shortcomings, such as premature convergence. This makes that the final solution is a local optimum, but not the desired results. At present, there are many experts and scholars have conducted a lot of research on it. A new general form of velocity update rule for PSO algorithm that contains a user-definable function f was proposed. It was proven that the proposed velocity update rule guarantees to address all of these issues [1]. Ching-Shih Tsou *et al.* incorporated a local search and clustering mechanism into the multi-objective particle swarm optimization (MOPSO) algorithm to solve two bi-objective inventory planning models, both having a cost minimization objective along with the stock out occasions minimization objective (named as N-model) and the number of items stocked out minimization objective (named as B-model) [2]. Hsing-Hung Lin proposed a particle swarm optimization to address open-shop scheduling problems with multiple objectives. The particle position representation, particle velocity and particle movement were modified due to the discrete essence of the scheduling problem [3]. The Random PSO was used which utilizes the weighted particle to guide the search direction for both explorative and exploitative searches. The Random PSO and DE were efficiently combined so as to overcome the disadvantages faced by both the algorithms individually and were used for the design of linear phase low pass and high pass FIR filters [4]. A new population heuristic based on the PSO technique was presented to solve the single machine early/tardy scheduling problem against a restrictive common due date [5].

These algorithms mainly focus on particle swarm of inertia weight, acceleration coefficients and other aspects of the optimization. To a certain extent, these algorithms improve the performance of particle swarm algorithm and accuracy, but ignore the particle initial position and other factors on the impact of the final solution. This paper presents a kind of particle swarm optimization algorithm with chaotic mapping. In the initial stage of particles, CM-PSO algorithm optimizes the position of particles to get the optimal solution. In the later stage of the algorithm, the impact factor values are no longer used with the linear decreasing mode, but for the global optimal solution by nonlinear decreasing mode. Experiments show that the CM-PSO algorithm has global search ability and the success rate is increased.

2. BASIC KNOWLEDGE

2.1. The Tilt Angle

In the PSO algorithm, each particle is represented a potential feasible solution in the search space. The feasible solution is calculated by the fitness function. Each particle has a speed variable to determine its flying direction and distance [6, 7]. The standard particle swarm algorithm is described as follows: the total particle number is N in the D dimension search space. The position vector of i particles is expressed as $x_i = (x_{i1}, x_{i2} \dots x_{iD})$. The velocity vector of i

1874-110X/14 2014 Bentham Open

particles is expressed as $v_i=(v_{i1},v_{i2},\ldots,v_{iD})$. In the search space, each particle moves as a certain velocity and direction. When the particle reached the new position, it is evaluated by the fitness function. The future direction and velocity of the particle is decided by the fitness value[8].

The history optimal solution of a particle x_i is expressed as $pBest_i=(pBest_{i1}, pBest_{i2}, \ldots, pBest_{iD})$. The optimal solution of all the particles are expressed as gBest(the value is optimal for all values in the $pBest_i$). According to the formula (1), the velocity and position of each particle in the population is updated.

$$\begin{cases} v_{id}^{t+1} = \omega v_{id}^{t} + c_1 r_1 (pBest_{id}^{t} - x_{id}^{t}) + c_2 r_2 (gBest^{t} - x_{id}^{t}) \\ x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1} \end{cases} \quad (1)$$

In formula (1), i=1,2…N, d=1,2…D. The symbol t is said the number of iterations.The symbol x_i^t is said the position of the i particle in t iterations. v_{id}^t is said the velocity of the i particle in t iterations. ω is said the inertia weight. The symbol c_1 and c_2 are represented as learning factor. They are two positive real number, usually taken as $c_1=c_2=2$. r_1 and r_2 is a random number between [0,1].

2.2. The Tilt Angle

Chebyshev chaotic mapping is a typical one-dimensional chaotic mapping, the mapping equation is shown as formula (2).

$$x_{n+1} = \cos(\beta \times \arccos(x_n)) \quad (2)$$

In formula (2), β represents the control parameters, when β is greater than or equal to 2, the mapping is in a chaotic state[9].

3. THE DESIGN OF ALGORITHM

3.1. Chaos Initialization

In the PSO algorithm, the distribution of initial population is one of the important factors to influence the final convergence result. From equation (1), it can be seen that the influence of the position to the final result is very big in the initial population. The behind calculation data are based on the initial population. In the selection of the initial population, if the initial value are selected in the vicinity of the global optimum solution, the iterative searching algorithm can search the global optimal solution in a very short period of time [10, 11]. On the contrary, the improper selection of initial population is likely to cause the algorithm in local optimum. Finally, the global convergence of the algorithm will have a very big impact. According to the characteristics of randomness and sensitivity to initial conditions, chaotic mapping is used to generate the initial population, which reduce the impact on the final results of the initial population.

In this paper, the Chebyshev chaotic mapping in the formula (2) is used to generate the initial population. The initial parameter settings for the Chebyshev chaotic mapping are: β=π, x_0=0.234567. The operating steps are shown as follows.

Step 1: The Chebyshev chaotic mapping is used to generate the first particle position x_i and velocity v_i, which are the 1*D of random vector. Vector value should be between (0,1). They are shown as formula (3).

$$\begin{cases} x(i,d) = |x(i,d)| & x(i,d) < 0 \\ v(i,d) = |v(i,d)| & v(i,d) < 0 \end{cases} \quad (3)$$

x(i,d) and v(i,d) indicate the Dth dimension vector of particle x_i, which respectively represent the position and velocity. If the value is less than 0, the absolute value is taken.

Step 2: 2* (N-1) vectors are iteratively generated, which used to represent the initial population in the rest of the N-1 particle position $x_2,x_3\ldots x_N$ and velocity $v_2,v_3\ldots v_N$. The iterative methods are shown in the formula (4).

$$\begin{cases} x(i+1,d) = 1 - [1 - x(i,d)] \times x(i,d) \times 4 \\ v(i+1,d) = 1 - [1 - v(i,d)] \times x(i,d) \times 4 \end{cases} \quad (4)$$

Step 3: Because the search space is not the same in different optimization problems, the generated value in Step2 should be expressed as another way. The generated value is mapped to the defined scope of the search space by the formula (5).

$$\begin{cases} x(i,d) = (2 \times x(i,d) - 1) \times S_{id} \\ v(i,d) = (2 \times v(i,d) - 1) \times S_{id} \end{cases} \quad (5)$$

In formula (5), $(-S_{id}, S_{id})$ is said the search scope.

Through the above steps, the initial population in PSO algorithm are generated by using one dimension Chebyshev chaotic mapping.

3.2. Chaos Initialization

Inertia value ω is used to control the effect of previous iteration of this iteration. Ajusting ω value can keep the best relationship between the global search and local search, so as to improve the performance of the proposed algorithm. If ω value is larger, the exploration ability of particle is enhanced [12-14]. This is conducive to the global search, and can avoid the local extremum, but not easy to get the accurate solution. It is diffcult to obtain the accurate solution. On the contrary, if the ω value is smaller, the global exploration ability will be weakened, and is more inclined to local search. The speed of convergence is slow and sometimes fall into the local extremum. Therefore, selection of appropriate ω for algorithm is essential. The right values can improve the optimization performance, but also can reduce the number of iterations. In the current PSO algorithm, ω value is fixed as a value or decreased with the iteration number linear value [15, 16]. This method has shortcomings, and can not be adaptive with dynamic changement. In order to overcome this problem, this paper adopts the chaotic mapping is used to adjust the ω value.

The dynamic change of inertia weight is as shown in formula (6).

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{\omega_{\max} + \omega_{\min}} \times \frac{t}{t_{\max}} \times rand \quad (6)$$

Table 1. The number of iterations and the successful convergence rate based on four test functions.

	Test Function			
Algorithm	f_1	f_2	f_3	f_4
SPSO	1863/42%	1685/96%	1253/100%	1536/51%
CM-PSO	536/76%	635/100%	433/100%	685/73%

Among them, the ω_{max} and ω_{min} said range of ω, general value (0.4,0.9). The symbol t said the current iteration number, t_{max} represents the maximum number of iterations. rand said the random number which are generated by Chebyshev chaotic mapping collection. Its range is (0,1). And the initial parameter of the mapping equation is shown as: β=3, x_0=0.123456.

4. ALGORITHM PERFORMANCE ANALYSIS

4.1. The Test Function

In order to test the performance of the CM-PSO algorithm, four typical benchmark functions are used for the simulation[17-18]. Compared with the standard particle swarm optimization algorithm(SPSO), it is confirmed whether CM-PSO algorithm performance meets the requirements.

4.1.1. Rosenbrock Function

It is a unimodal ill posed two times function, the solution is not easy. Because the function provides very few messages for search, and has a strong correlation between variables, which makes it difficult to identify the direction of search. It is very difficult to find the global optimal point, so this function is used to evaluate the optimization algorithm implementation ability. The function has global minimal value 0 in the x= (1,1,... 1). The function expression is shown as formula (7).

$$f_1(\mathrm{x}) = \sum_{i=1}^{n-1} (100(\mathrm{x}_{i+1}^2 - \mathrm{x}_i)^2 + (1 - x_i)^2) \tag{7}$$

4.1.2. Griewank Function

The function is a complex multi peak function. With the increasing function dimension, local optimal area will become more and more narrow, and the process of finding the global optimal value becomes relatively easy. Its expression is shown as formula (8).

$$f_2(\mathrm{x}) = \frac{1}{4000} \sum_{i=1}^{n} \mathrm{x}_i^2 - \prod_{i=1}^{n} \cos \frac{x}{\sqrt{i}} + 1 \tag{8}$$

4.1.3. Sphere Function

It is a unimodal function. This algorithm can easily find its optimal solution, which can complete the numerical optimization. It is helpful to study the optimization algorithm effect in dimension. The function expression is shown as formula (9).

$$f_3(\mathrm{x}) = \sum_{i=1}^{n} \mathrm{x}_i^2 \tag{9}$$

4.1.4. Rastrigrin Function

On the basis of Sphere function, the function generates a large number of local optimal value by cosine function. It is easy to make the optimization algorithm falling into local optimum in the optimization process, thus unable to get the global optimal solution. Its expression is shown as formula (10).

$$f_4(\mathrm{x}) = \sum_{i=1}^{n} (\mathrm{x}_i^2 - 100\cos(2\pi\,\mathrm{x}_i) + 10) \tag{10}$$

4.2. Experimental Results and Analysis

In order to get objectively result, the same parameters is used in the experiment. Learning factor c_1 and c_2 are 2. ω_{max} and ω_{min} are 0.95 and 0.35. The initial value of ω is set to 0.95. The particle dimension is D=5, the population size is N=100. The maximum number of iterations of t_{max} is 2000.

For each function, the algorithm is run repeatedly 100 times. The average iteration number and success rate of convergence are calculated as shown in Table **1**.

Because the function of f1 is very difficult to optimize to the global optimal solution, the CM-PSO algorithm also failed to obtain the global minimum every time. But the experimental results show that, compared with SPSO, the CM-PSO algorithm has greatly improved the success rate. As the function of f2 and f3, the two algorithms both search basically the global optimal value. But the CM-PSO algorithm has fewer iterations, which shows its effect is better. As the f4 function is very difficult to minimize, the success of the CM- PSO algorithm rate hasn't reached 100%

It can be seen from the experimental results that the optimization effect of the CM-PSO algorithm is obviously higher than that of the SPSO algorithm.

BF said the best result in 100 experiments, used to compare two algorithms running result. MBF said the arithmetic average of 100 tests of theoptimal value, and it can reflect the precision of algorithm. The operation results are shown as Table **2**.

It can be seen from Table **2** that BF and MBF of CM-PSO algorithm are better than the SPSO algorithm in four different test functions. This shows that the performance of the CM-PSO algorithm has been greatly improved.

In the group of the initialization phase, the CM-PSO algorithm is introduced to optimize population by the chaotic mapping. In the inertia weight stage, the CM-PSO algorithm has abandoned the linear change, and introduces chaos mapping function in nonlinear decreasing inertia weight. So each

Table 2. The operation results of four test function.

Function	Algorithm	BF	MBF
f_1	SPSO	4.3568	9.1538
	CM-PSO	0.6259	1.5326
f_2	SPSO	$0.0866e^{-5}$	$0.1324\ e^{-5}$
	CM-PSO	$1.0012e^{-8}$	$1.2586\ e^{-8}$
f_3	SPSO	$7.8563e^{-7}$	$4.2563\ e^{-6}$
	CM-PSO	$1.0526e^{-9}$	$5.5263e^{-8}$
f_4	SPSO	0.0072	0.0468
	CM-PSO	$3.6257e^{-5}$	$8.2512e^{-4}$

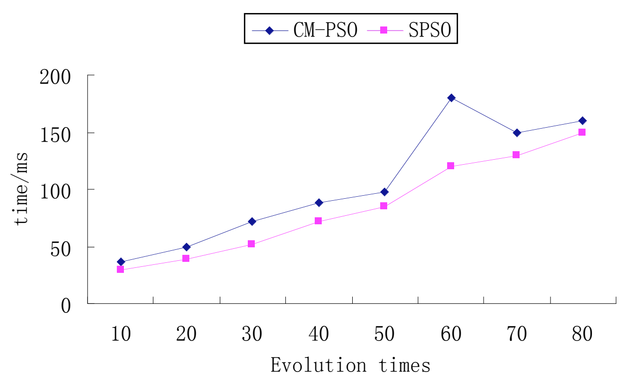

Fig. (1). Comparison of solving the problem of time.

population evolutionary time of the CM-PSO algorithm is also higher than that of the SPSO algorithm. The evolutionary time of two algorithms are shown as Fig. (**1**).

In short, the CM-PSO algorithm optimization results is better, and has good reliability and fast convergence speed. Although consumption time slightly of the CM-PSO algorithm is longer than SPSO, but in the acceptable range.

CONCLUSION

In order to solve the problem of premature convergence occurs in PSO algorithm, a particle swarm optimization algorithm with chaotic mapping has been proposed . The algorithm optimized the initial state of the population by Chebyshev chaotic mapping, which improved the probability for calculating optimal solution of the particle swarm. In order to avoid the disadvantages of linear decline, the inertia weight is dynamicly adjusted at the same time. In the simulation experiment, the performance of CM-PSO and SPSO algorithms are compared by using four kinds of typical function. The results show that the CMPSO algorithm has strong global search ability. Although the calculation time is slightly added, it is worth for improving the probability of getting global optimal solution. All in all, the algorithm is simple and easy to implement and can effectively avoid the premature convergence problem of particle swarm optimization algorithm.

CONFLICT OF INTEREST

The authors confirm that this article content has no conflict of interest.

ACKNOWLEDGEMENTS

This work is supported by the Research on Henan Province Natural Science Fund Project (142300410414, 142300410413), the Education Department of Henan province science and technology research project (12A520033), by the National Natural Science Foundation of China (Grant No. 61401242), the Research Foundation for Advanced Talents of Nanyang Normal University (ZX2014058), Technology Research and Development Program of Lianyungang (Grant No.SH1223).

REFERENCES

[1] M. Z. Bonyadi, and Z. Michalewicz, “A locally convergent rotationally invariant particle swarm optimization algorithm,” *Swarm Intelligence*, vol. 8, no. 3, pp. 159-198, 2011.

[2] C. S. Tsou, K. J. Chung, and C. H. Hsu, “Multi-objective analysis of the inventory planning problem using particle swarm optimization,” *Journal of Statistics and Management Systems*, vol. 11, no. 1, pp. 131-140, 2008.

[3] H. Lin, and D. Y. Sha, “A new particle swarm optimization for multi-objective open shop scheduling,” *Journal of Information and Optimization Sciences*, vol. 32, no. 6, pp. 1269-1287, 2011.

[4] D. M. Vasundhara, and S. Ghoshal, “Digital FIR Filter Design Using Hybrid Random Particle Swarm Optimization with Differential Evolution,” *International Journal of Computational Intelligence Systems*, vol. 6, no. 5, pp. 911-927, 2013.

[5] A. C. Nearchou, and S. Omirou, “A Particle Swarm Optimization Algorithm for Scheduling Against Restrictive Common Due Dates,” *International Journal of Computational Intelligence Systems*, vol. 6, no. 4, pp. 584-699, 2013.

[6] R. Nasimi, M. Shahbazian, and R. Irani, “Permeability Estimation of the Reservoir Based on Particle Swarm Optimization Coupled with Artificial Neural Networks,” *Petroleum Science and Technology*, vol. 29, no. 22, pp. 2329-2337, 2011.

[7] C. Boonchuay, and W. Ongsakul, “Risk-constrained optimal bidding strategy for a generation company using self-organizing hierarchical particle swarm optimization,” *Applied Artificial Intelligence*, vol. 26, no. 3, pp. 246-260, 2012.

[8] C. C. Chiu, and W. C. Hsiao, “Comparison of asynchronous particle swarm optimization and dynamic differential evolution for partially immersed conductor,” *Waves in Random and Complex Media*, vol. 21, no. 3, pp. 485-500, 2011.

[9] N. Khaji, and M. Mirzajani, “Frequency domain analysis of elastic bounded domains using a new semi-analytical method,” *Acta Mechanica*, vol. 224, no. 7, pp. 1555-1570, 2013.

[10] L. C. Cagnina, S. C. Esquivel, and C. C. Coello, "Solving constrained optimization problems with a hybrid particle swarm optimization algorithm," *Engineering Optimization*, vol. 43, no. 8, pp. 843-866, 2011.
[11] S. Vakili, and M. S. Gadala, "A Modified Sequential Particle Swarm Optimization Algorithm with Future Time Data for Solving Transient Inverse Heat Conduction Problems," *Numerical Heat Transfer, Part A: Applications*, vol. 59, no. 12, pp. 911-933, 2011.
[12] P. Somasundaram, and N. M. Jothi Swaroopan, "Fuzzified Particle Swarm Optimization Algorithm for Multi-area Security Constrained Economic Dispatch," *Electric Power Components and Systems*, vol. 39, no. 10, pp. 979-990, 2011.
[13] H. H. Zeineldin, and A. F. Zobaa, "Particle Swarm Optimization of Passive Filters for Industrial Plants in Distribution Networks," *Electric Power Components and Systems*, vol. 39, no. 16, pp. 1795-1808, 2011.
[14] E. Assareh, M. A. Behrang, and A. Ghanbarzdeh, "Forecasting Energy Demand in Iran Using Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) Methods," *Energy Sources, Part B: Economics, Planning, and Policy*, vol. 7, no. 4, pp. 411-422, 2012.
[15] T. Niknam, E. Farsani, and M. Nayeripour, "Hybrid Fuzzy Adaptive Particle Swarm Optimization and Differential Evolution Algorithm for Distribution Feeder Reconfiguration," *Electric Power Components and Systems*, vol. 39, no. 2, pp. 158-175, 2011.
[16] V. Aristidis, P. Maria, and L. Christos, "Particle Swarm Optimization (PSO) algorithm for wind farm optimal design," *International Journal of Management Science and Engineering Management*, vol. 5, no. 1, pp. 53-58, 2010.
[17] F. J. Ruiz-Rodriguez, M. Gomez-Gonzalez, and F. Jurado, "Binary Particle Swarm Optimization for Optimization of Photovoltaic Generators in Radial Distribution Systems Using Probabilistic Load Flow," *Electric Power Components and Systems*, vol. 39, no. 15, pp. 1667-1684, 2011.
[18] H. C. Tsai, Y. Y. Tyan, and Y. W. Wu, "Isolated particle swarm optimization with particle migration and global best adoption," *Engineering Optimization*, vol. 44, no. 12, pp. 1405-1424, 2012.

Received: September 16, 2014 Revised: December 23, 2014 Accepted: December 31, 2014