



Research on Improved Chaotic Particle Optimization Algorithm Based on Complex Function

Xiangli Xia¹ and Shijin Li^{2*}

¹Chongqing Key Laboratory of Spatial Data Mining and Big Data Integration for Ecology and Environment, Rongzhi College of Chongqing Technology and Business University, Chongqing, China, ²Academic Affairs Office, Yunnan University of Finance and Economics, Kunming, China

In order to improve the performance of Particle Swarm Optimization (PSO) algorithm in solving continuous function optimization problems, a chaotic particle optimization algorithm for complex functions is proposed. Firstly, the algorithm uses qubit Bloch spherical coordinate coding scheme to initialize the initial position of the population. This coding method can expand the ergodicity of the search space, increase the diversity of the population, and further accelerate the convergence speed of the algorithm. Secondly, Logistic chaos is used to search the elite individuals of the population, which effectively prevents the PSO algorithm from falling into local optimization, thus obtaining higher quality optimal solution. Finally, complex functions are used to improve chaotic particles to further improve the convergence speed and optimization accuracy of PSO algorithm. Through the optimization tests of four complex high-dimensional functions, the simulation results show that the improved algorithm is more competitive and its overall performance is better, especially suitable for the optimization of complex high-dimensional functions.

Keywords: PSO algorithm, complex function, chaos search, convergence rate, improve chaotic particles

OPEN ACCESS

Edited by:

Jia-Bao Liu,
Anhui Jianzhu University, China

Reviewed by:

Ndolane Sene,
Cheikh Anta Diop University, Senegal
Hanliang Fu,
Illinois State University, United States

*Correspondence:

Shijin Li
lee_shijin@sina.com

Specialty section:

This article was submitted to
Mathematical and Statistical Physics,
a section of the journal
Frontiers in Physics

Received: 19 June 2020

Accepted: 30 July 2020

Published: 09 September 2020

Citation:

Xia X and Li S (2020) Research on
Improved Chaotic Particle
Optimization Algorithm Based on
Complex Function.
Front. Phys. 8:368.
doi: 10.3389/fphy.2020.00368

INTRODUCTION

Complex function optimization is an important research direction of optimization problems. Generally speaking, the solving methods of optimization problems can be divided into analytical method and numerical calculation [1]. The analytical method solves the problem according to the relationship between the derivative of the objective function and the extreme value of the function. This method is only suitable for optimization problems with relatively simple objective function. According to the variation rule of objective function value, in appropriate steps along the direction that optimizes the value of the objective function, an approximate calculation method that approaches the optimal point of the objective function step by step. This method is good at solving continuous differentiated convex optimization problems. With the continuous expansion of engineering optimization problems, most of the objective functions are non-convex optimization problems. The emergence of group intelligent optimization algorithms provides a limited way for complex function optimization problems [2, 3].

Particle Swarm Optimization (PSO) algorithm is a kind of bionic intelligent optimization algorithm based on population, which is proposed by Kennedy et al. [4]. Each particle in the PSO algorithm represents a feasible solution; the location of food source is the global optimal location point. PSO has strong search diversity, simple operation and few adjustment parameters. As soon as it was proposed, it was widely used [5, 6], especially PSO has shown excellent optimization ability

in complex optimization problems [7]. In reference [8], Eberhart and Shi found that when the maximum velocity of the particle is not too small ($v_{\max} > 3$), The inertia weight $\omega=0.8$ is best. In addition, this conclusion has been confirmed on many subsequent issues. Clerf [9] when carefully studying a general PSO system, the change of speed can be controlled by controlling ϕ_1 and ϕ_2 . In order to improve the calculation speed of particle motion trajectory, Clerf introduced contraction factor χ to improve the basic model of PSO algorithm. PSO algorithm has the following advantages:

First, the algorithm is easy to describe.

Second, there are few parameters to be adjusted in the algorithm.

Third, the number of functions to be evaluated in the algorithm is small.

Fourth, the number of populations required by the algorithm in the process of solving the problem is small.

Fifth, the algorithm converges quickly.

Because there are few parameters in PSO, PSO is easy to Realize, there is also less demand for computing resources, the gradient information of fitness function is not needed, only the value of fitness function is needed. Although PSO algorithm has various advantages, but the PSO algorithm itself also has several limitations, its performance is as follows. First, PSO is a probabilistic algorithm, without systematic and standardized theoretical support, It is still difficult to verify the correctness of PSO algorithm from a mathematical point of view so far. Moreover, based on the theory of random events, it is an extremely difficult task to analyze the particle trajectory quantitatively in the search process of PSO algorithm. However, this is also related to the key issues of convergence and parameter selection of PSO algorithm. Although most scholars are currently verifying the convergence of its improved PSO algorithm, But none of them has produced a set of mature and universal theories. Second, the behavior and characteristics of complex systems are the emergence of behaviors that are continuously superimposed through interaction between individual individuals in the system. Although the control of individual behaviors is relatively simple, however, this does not mean that the control of the whole system is an easy task. Third, as far as the whole algorithm is concerned, due to the lack of balance mechanism, when solving some complex or special problems, the algorithm is easy to lose population diversity and fall into local extreme. The structure and contents of the paper are as follows: (1) Introduce the basic particle swarm optimization algorithm and the algorithm flow; (2) Introduce the quantum chaotic adaptive particle swarm optimization algorithm, explaining the quantum Bloch coordinate coding, chaos optimization method, quantum particle swarm optimization algorithm, adaptive inertia weight and the improved algorithm flow, respectively; (3) Comparing the convergence test through experiments.

BASIC PSO

The mathematical description of PSO algorithm is follow that the population with dimension D and scale N can be expressed as

$X = \{X_1, X_2, \dots, X_D\}$, then at time t, the position of the i^{th} particle is $X_i(t) = \{X_{i1}(t), X_{i2}(t), \dots, X_{iD}(t)\}$ and its velocity is $V_i(t) = \{V_{i1}(t), V_{i2}(t), \dots, V_{iD}(t)\}$. The algorithm always maintains two optimal positions: evolution process, the individual best position $pbest_i(t)$ of particle i, expressed as $p_i(t) = \{p_{i1}(t), p_{i1}(t), \dots, p_{iD}(t)\}$, and $gbest(t)$ of population a best location, expressed as $p_g(t) = \{p_{g1}(t), p_{g1}(t), \dots, p_{gD}(t)\}$. If the optimization model is $\max f(X)$, the update formulas of $p_i(t)$ and $p_g(t)$ are as follows:

$$p_i^{(t+1)} = \begin{cases} X_i^{(t+1)} & f(X_i^{(t+1)}) > f(p_i^{(t)}); \\ P_i^{(t)} & f(X_i^{(t+1)}) \leq f(p_i^{(t)}) \end{cases} \quad (1)$$

$$p_g^{(t+1)} = \max_{1 \leq i \leq N} p_i^{(t+1)} \quad (2)$$

In the t-th iteration, p_{best} and $gbest$ represent historical and global optimal positions. Then the calculation formula for particle flight update is expressed as follows:

$$v_{i,d}^{t+1} = wv_{i,d}^t + c_1r_1(pbest_{i,d}^t - x_{i,d}^t) + c_2r_2(gbest_d^t - x_{i,d}^t) \quad (3)$$

$$x_{i,d}^{t+1} = x_{i,d}^t + v_{i,d}^{t+1} \quad (4)$$

Where $v_{i,d}^{t+1}$ represents the flight speed of i^{th} particle iterations $t + 1$; X_{iD} represents the position of i^{th} particle with iterations t. w represents the inertia weight, which is taken here as 0.6; c_1 and c_2 represent learning factors, generally taking $c_1 = 2, c_2 = 2; r_1 \in [0, 1] r_2 \in [0, 1]$.

The flow of the basic PSO algorithm is as follows:

- (1) Set the parameters of PSO algorithm, such as population size, problem dimension, inertia weight, maximum range and maximum speed, etc. Randomly initialize group position and speed.
- (2) Judge whether the particle is beyond the search range, and correct the position if it is beyond the range.
- (3) According to the state of each particle, calculate the corresponding fitness value.
- (4) Update p_{best} according to the current fitness value.
- (5) Update $gbest$ according to the current fitness value.
- (6) According to formulas (3) and (4), update the speed and position.
- (7) Judge the termination condition, and return to (2) if it is not terminated, otherwise it will end.

QUANTUM CHAOS ADAPTIVE PSO ALGORITHM

Initial Population of Quantum Bloch Coordinate Coding

In quantum computation, the smallest information unit is expressed by qubits, which are also called qubits. The state of a qubit can be expressed as [10]:

$$|\varphi\rangle = \cos(\theta/2)|0\rangle + e^{i\varphi} \sin(\theta/2)|1\rangle \tag{5}$$

In Equation (5), numbers φ and θ define a point, Qubits establish correlation with Bloch spherical points, and the conversion formula is as follows:

$$|\varphi\rangle = [\cos\varphi \sin\theta \quad \sin\varphi \sin\theta \quad \cos\theta]^T \tag{6}$$

Let p_i be the i -th candidate solution in the group, and its coding scheme is as follows:

$$p_i = \begin{bmatrix} \cos\varphi_{i1} \sin\theta_{i1} & \cdots & \cos\varphi_{in} \sin\theta_{id} \\ \sin\varphi_{i1} \sin\theta_{i1} & \cdots & \sin\varphi_{in} \sin\theta_{id} \\ \cos\theta_{i1} & \cdots & \cos\theta_{id} \end{bmatrix} \tag{7}$$

Where $\varphi_{ij} = 2\pi \times rnd$, $\theta_{ij} = \pi \times rnd$ and $rnd \in [0, 1]$; Each candidate solution occupies three positions in the space, i.e., represents the following three optimization solutions:

$$\begin{aligned} P_{ix} &= (\cos\varphi_{i1} \sin\theta_{i1}, \dots, \cos\varphi_{id} \sin\theta_{id}); \\ P_{iy} &= (\sin\varphi_{i1} \sin\theta_{i1}, \dots, \sin\varphi_{id} \sin\theta_{id}); \\ P_{iz} &= (\cos\theta_{i1}, \dots, \cos\theta_{id}) \end{aligned} \tag{8}$$

Note that the feasible solutions corresponding to P_{ix}, P_{iy} and P_{iz} are as follows:

$$\begin{aligned} x_{ix} &= (x_{i1}^1, x_{i2}^1, \dots, x_{in}^1); \\ x_{iy} &= (x_{i1}^2, x_{i2}^2, \dots, x_{in}^2); \\ x_{iz} &= (x_{i1}^3, x_{i2}^3, \dots, x_{in}^3) \end{aligned} \tag{9}$$

Transformation of solution space: Bloch coordinates of the i -th qubit on candidate solution p_i are $[x_{ix}, x_{iy}, x_{iz}]^T$, and the value of solution space is $[a_j, b_j]$, then the transformation formula mapping $I^d = [-1, 1]^d$ is follow:

$$\begin{aligned} x_{ij}^1 &= \frac{1}{2} [b_j (1 + x_{ix}) + a_j (1 - x_{ix})]; \\ x_{ij}^2 &= \frac{1}{2} [b_j (1 + x_{iy}) + a_j (1 - x_{iy})]; \\ x_{ij}^3 &= \frac{1}{2} [b_j (1 + x_{iz}) + a_j (1 - x_{iz})] \end{aligned} \tag{10}$$

Therefore, individuals with smaller fitness values are selected as the initial population among all candidate solutions. Bloch coding can enhance the ergodicity of the optimization space, improve the population, and further improve the optimization performance.

Chaos Optimization Method

Chaos has the characteristics of randomness, ergodicity and regularity. In the field of optimization design, the ergodicity of chaos phenomenon can be used as an optimization mechanism to avoid falling into local minima in the search process. Chaotic variables are used to search and this method is applied to the optimization of continuous complex objects. The steps of chaos optimization algorithm using Logistic mapping are as follows:

(1) Let $k = 0, x_j^k, j = 1, 2, \dots, n$ to chaotic variables $x_j^k \in [0, 1]$.

$$s_j^k = \frac{x_j^k - x_{\min,j}}{x_{\max,j} - x_{\min,j}} \quad j = 1, 2, \dots, n \tag{11}$$

In the formula, $x_{\max,j}$ is max bounds and $x_{\min,j}$ is the search min bounds of the j -dimensional variable, respectively.

(2) Calculate next value s_j^{k+1} .

$$S_j^{k+1} = 4s_j^k(1 - s_j^k) \tag{12}$$

(3) The chaotic variable s_j^k is transformed into the decision variable x_j^{k+1} . Using a certain chaotic mapping structure to generate chaotic sequences, m is the length of chaotic sequences. These sequences are inversely transformed to the original search space through equation (13).

$$x_j^{k+1} = x_{\min,j} + s_j^{k+1}(x_{\max,j} - x_{\min,j}) \tag{13}$$

The new solution is taken as the result of chaotic optimization $K=K+1$, otherwise, the new solution is transferred to (2) and the iteration is continued.

- (4) Calculate the fitness value $F(X_i)$ of X_i and compare it with the fitness value $F(X_i)$ of X_i to retain the best solution;
- (5) Updating Chaotic Search Space $(x_{\min,j}, x_{\max,j})$.

Quantum PSO Algorithm

In 2004, Sun studied the convergence behavior of absorption-related particles and proposed a quantum PSO algorithm based on the model of quantum mechanics. In quantum mechanics, when each particle moves in the search space, there is a DELTA potential well-centered on p . The properties of particles in quantum space satisfying aggregated states are completely different from those in classical mechanics. The particle with quantum behavior has no definite trajectory when moving, i.e., The velocity and position are uncertain, and this uncertainty makes the possible position of the particle "everywhere" (i.e., In the whole feasible solution region) full of possibilities, and the particle has the possibility to get rid of the local optimal value point with large interference. It can ensure the global convergence of the algorithm and has only position vector, no speed vector, few control parameters and strong optimization ability in the optimization model.

Quantum PSO algorithm is described as follows: in D -dimensional space, there are m particles, the individual extreme point $ispBest_i = (pBest_{i1}, pBest_{i2}, \dots, pBest_{iD})$, and the potential center point is p ; The current global extreme point searched by the whole particle swarm is $gBest=(gBest_1, gBest_2, \dots, gBest_D)$. Then, The position update operation for this particle is as follows:

$$p = \frac{rand1() \times pBest_{id} + rand2() \times gBest_d}{rand1() + rand2()} \tag{14}$$

$$l = z \times |X_{id}(t) - p| \quad 0 < z < \ln \sqrt{2} \quad (15)$$

$$X_{id}(t + 1) \begin{cases} p - l \times \ln[1/\text{rand3}()] & \text{rand3}() > 0.5; \\ p + l \times \ln[1/\text{rand3}()] & \text{rand3}() \leq 0.5 \end{cases} \quad (16)$$

Where $i = 1, 2, \dots, m$ and $Z \leq \ln \sqrt{2}$, usually 0.5–1.0; The range are (0, 1).

Adaptive Inertia Weights

The global exploration ability and local mining ability of PSO algorithm are contradictory to each other, and it is hard to seek a balance point. To balance the global exploration and local development capabilities of PSO algorithm, an improved chaotic particle optimization Algorithm is proposed to further modify the optimization algorithm effect.

Accord to the particle position update formula (3), two adaptive inertia weights w_j and w'_j are introduced. Among them, w_j is used to control the influence degree of the original particle position on the new particle position, and w'_j is used to balance the influence weight of the particle flight speed on the new position. The improved particle position update formula is expressed as follows:

$$x_{i,d}^{t+1} = w_j x_{i,d}^t + w'_j v_{i,d}^{t+1} \quad (17)$$

From above formula 17, w_j and w'_j can keep the fault tolerance of particle population, and enhance robustness of the algorithm to quit the local optimization in the optimization process.

The mathematical expressions for w_j and w'_j are as follows:

$$w_j = 1 / (1 + \exp(-f(j)/u)^{iter}) \quad (18)$$

$$w'_j = 1 - w_j \quad (19)$$

In the formula, $f(j)$ express adaptive value of the j -th particle, u express the best value in the particle population in the first iteration calculation, and $iter$ represents the current iteration number.

Quantum Chaos Adaptive PSO Algorithm Steps

The specific steps of QCPSO algorithm search are as follows:

- (1) Initialize all parameters including $N, c_1, c_2, M, v, D, t = 0$;
- (2) The current fitness value of each particle is calculated by the optimization function and compared with the fitness value corresponding to the individual historical optimal solution. If the current fitness value is better than the fitness value corresponding to the individual historical optimal solution, the current solution is replaced by the individual optimal solution $pbest$, otherwise it is not replaced.

(3)The optimal solution $gbest$ of the current population is determined by comparing the optimal fitness values of all particles;

(4) Updating the flight speed of particles;

(5) Updating the weights w_j and w'_j ;

(6) Updating the position of particles;

(7) If $t < M$ and not converge

$t = t + 1$

Go to step (3);

Otherwise

Find the global optimal solution and go to step (8);

(8) Output the best value.

CONVERGENCE TEST COMPARISONS

The PSO algorithm is improved by parameter adjustment strategy, and the search process is optimized by re-search and reverse learning. In order to embody the effectiveness of the improved PSO algorithm put forward in the research of test case generation, it is relative to other algorithms, and has the best effect. Based on existing problems that need to be studied, this paper uses Matlab 2016a programming to implement the above-mentioned algorithm, and evaluates the advantages and disadvantages through fitness value, average coverage rate and iteration times. To ensure the fairness and scientificity of the performance comparison of all algorithms, each group of experiments is run 100 times to obtain the average value.

The basic parameters of the whole experiment are seen as follows: $M = 1,000, N = 30, D = 30$, and the range of values is [0.4, 0.9]. To verify the superiority of QCPSO algorithm put forward, four typical test functions Sphere, Rosenbrock, Rastrigrin and Griewank are compared and tested. These four functions include unimodal function, multimodal function and trigonometric function, which are relatively comprehensive. The specific formulas of the test functions are as follows.

$$f_1(x) = \sum_{i=1}^n x_i^2 \quad x \in [-100, 100] \quad (20)$$

$$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2) \quad x \in [-30, 30] \quad (21)$$

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad x \in [-5, 5] \quad (22)$$

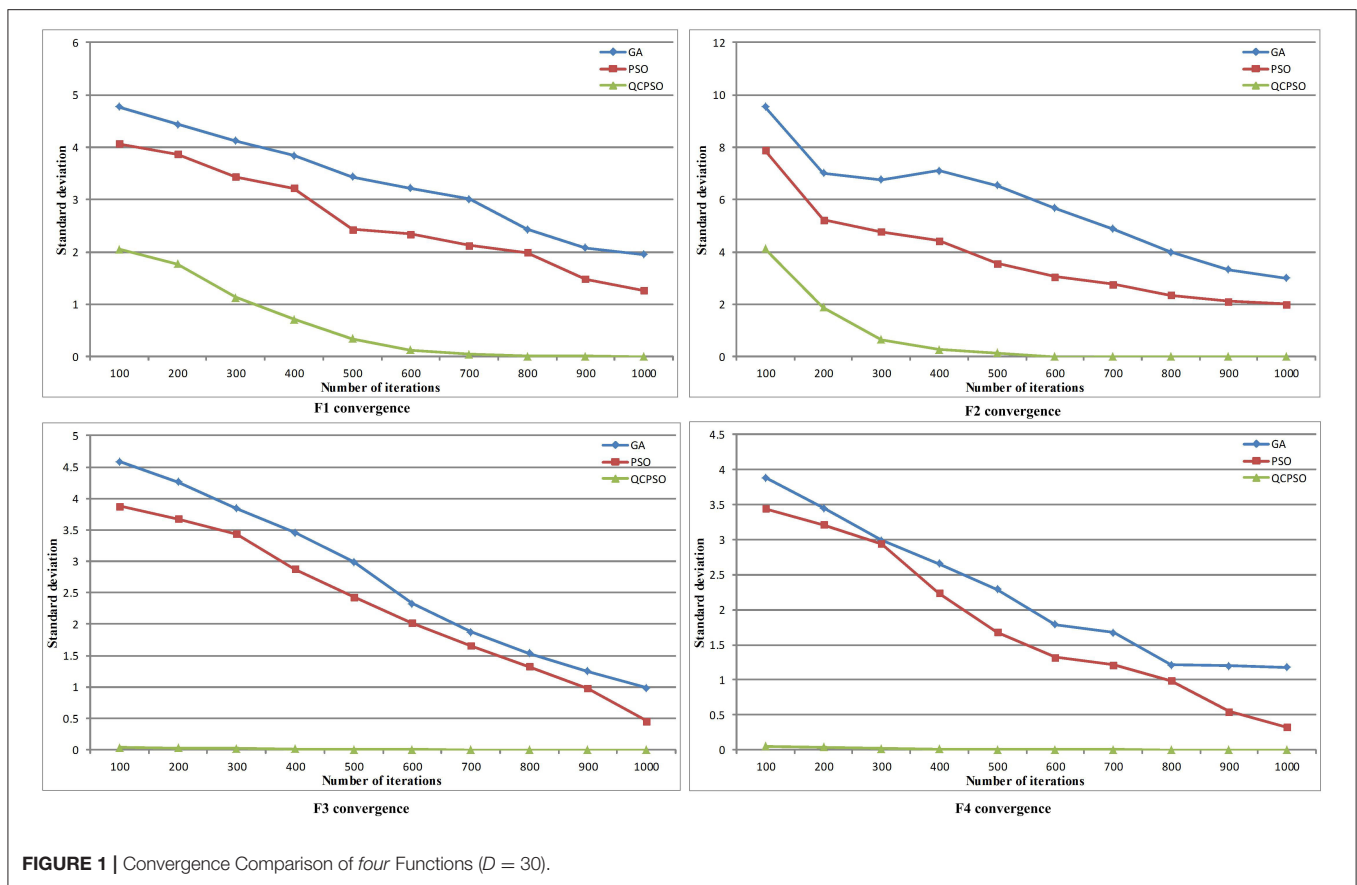
$$f_4(x) = (\sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1) / 4000 \quad x \in [-600, 600] \quad (23)$$

In order to verify the performance advantages of the improved PSO algorithm, it is necessary to judge the fitness values of four typical test functions and compare them with the results of other

TABLE 1 | Value for three Algorithms.

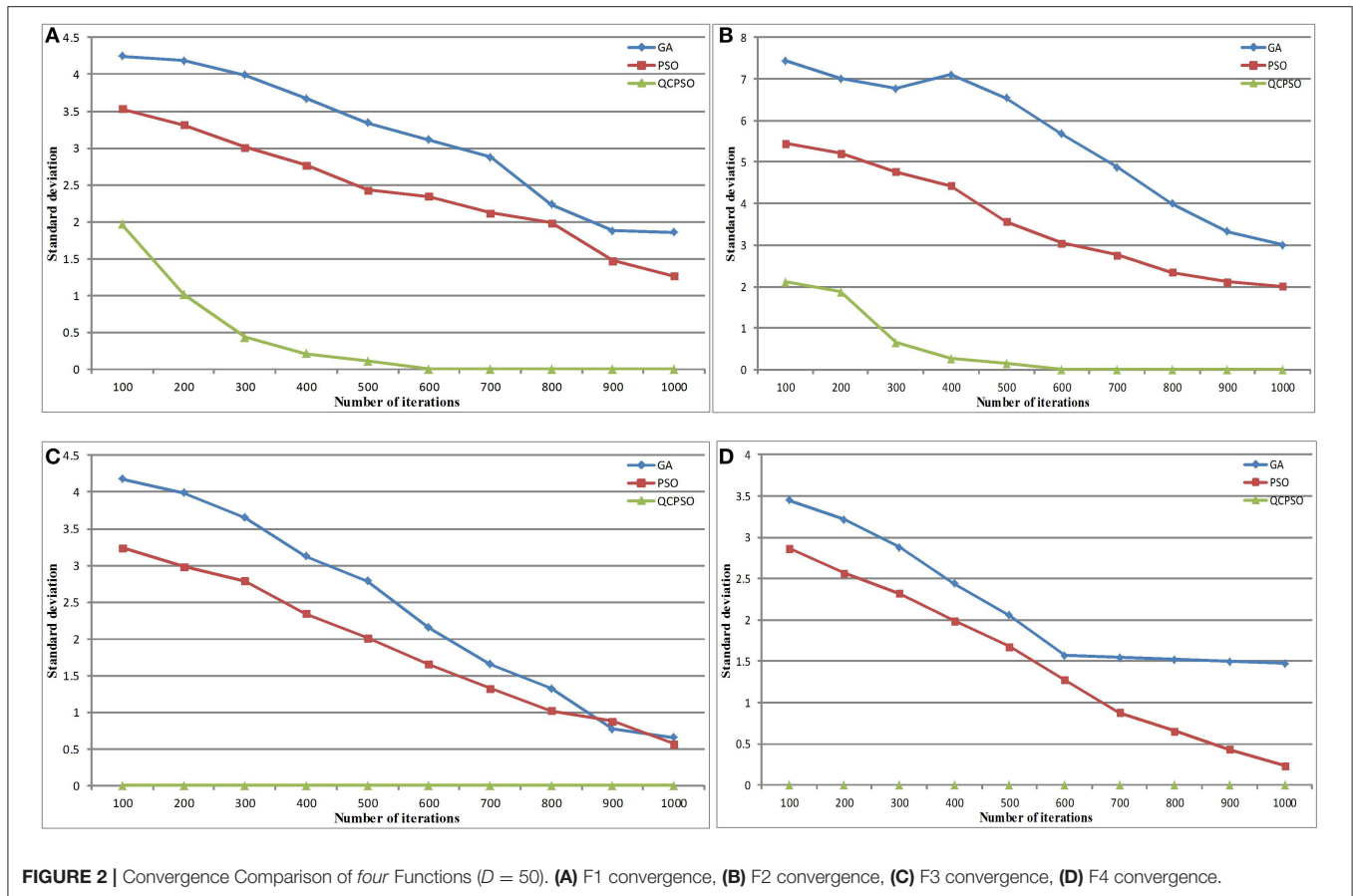
Algorithm	Value	<i>D</i> = 30, Iteration = 100				<i>D</i> = 50, Iteration = 100			
		F1	F2	F3	F4	F1	F2	F3	F4
PSO	Optimal	2.397	5.761	1.986	1.238	1.987	4.679	1.658	0.832
	Average	3.537	6.231	2.421	2.179	2.589	4.998	2.821	1.998
	S.D	4.176	7.876	3.876	3.459	3.765	5.452	3.239	2.987
GA	Optimal	2.568	6.324	2.451	1.591	2.512	5.311	1.981	1.287
	Average	3.679	6.981	3.318	2.993	2.991	5.688	3.003	2.342
	S.D	4.876	9.546	4.584	3.987	4.327	7.444	4.176	3.561
QCPSO	Optimal	0.112	2.346	0	0	0.021	0.114	0	0
	Average	1.543	3.378	1.2×10^{-3}	1.6×10^{-9}	1.221	1.557	1.1×10^{-18}	1.2×10^{-25}
	S.D	2.129	4.127	0.0334	0.0564	1.967	2.227	0	0

The optimization experiments are carried out under the test (F1, F2, F3, and F4) functions from the PSO, GA, and QCPSO algorithm, and the experimental results are analyzed and compared.



algorithms. First of all, the inertia weight is used to improve the learning factor. The relationship between the two can be divided into three types: linear, non-linear and trigonometric functions. In this paper, the non-linear relationship is used to carry out the relationship of learning factors, so that the learning factor changes non-linearly and gradually with the inertia weight, and the equation is expressed as $c = A\omega^2 + B\omega + C$; $c_2 + c_1 = 2$, and the inertia weight adopts the commonly used exponential

function decreasing method, taking $A = 0.45$, $B = 0.9$, $C = 0.45$, $\omega_{max} = 0.9$, $\omega_{min} = 0.4$. Secondly, the optimal solution and suboptimal solution in the current iteration are searched again, and the particles outside the tabu region are optimized by chaos. The improved algorithm is compared with PSO, GA and QCPSO through four typical test functions. The comparison of four typical test functions in four algorithms is shown in **Table 1**.



For a variable dimension of 20, As can be seen from **Table 1**: In 20 independent repetitions, For the test functions F3 and F4, QCP SO algorithm result the best value, but for average value and standard deviation of these two test functions, QCP SO is best again, and QCP SO are obviously better than PSO algorithm and GA algorithm. Accord to four functions, QCP SO is significantly better than PSO and GA, accord to the average value, standard deviation and the optimal value.

For the variable dimension of 50, it can be seen from **Table 1** that for the test functions F1, F2, F3, and F4, the average value and standard deviation of the optimal value of QCP SO are significantly better than those of PSO algorithm and GA algorithm, and when the corresponding value ratio dimension is 20, it is smaller as a whole.

In order to verify the convergence, the following is verified by iterations of the algorithm. The standard deviation value increases with the number of iterations. The specific effect is shown in **Figures 1, 2**.

Figures 1, 2 is a 30-dimensional optimization curve of 4 benchmark test functions in QCP SO, PSO, and GA (semi-logarithmic curve, and the optimization curve is drawn by semilogy function). The 30-and 50-dimensional optimization curves obtained in the experiment are similar to those in **Figure 2** and will not be given here due to limited space.

As can be seen from **Figures 1, 2**, when QCP SO algorithm optimizes F2 and F4 functions, there are many inflection points,

indicating that QCP SO’s ability to jump out of local optimization is enhanced. As can be seen intuitively from **Figure 1**, QCP SO algorithm is more effective than the PSO and GA algorithms for most of the four functions in **Table 1** with dimension 30. Among them, QCP SO algorithm obtains the optimal values in four test functions (F1, F2, F3, F4). The convergence speed of QCP SO is best than other three algorithms. As can be seen from **Figures 1, 2**, QCP SO quickly searches for satisfactory solutions for most optimization problems.

In short, QCP SO algorithm has greatly improved its optimization capability compared with standard PSO. For most optimization problems, QCP SO algorithm is better than GA and standard PSO algorithms.

CONCLUSION

In our research, an improved chaotic PSO algorithm based on complex functions are proposed. By comparing the convergence of four complex functions, the proposed QCP SO algorithm has high convergence and stable performance. The convergence of complexity functions is verified to illustrate the advantages of the algorithm, and more complexity functions are used to verify the advantages of the algorithm in the later period. It is proved in different dimensions that the reliability of the algorithm is verified by

the population of higher dimensions, and the universality of the algorithm is better explained. In the future research work, the improved particle swarm optimization algorithm for complex functions will be applied in other fields to improve the actual effect of the existing work [11–14]. For example, geographic location prediction, GPS trajectory prediction, flow prediction, etc.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author/s.

REFERENCES

- Liu H, Niu P. Enhanced shuffled frog-leaping algorithm for solving numerical function optimization problems. *J Intell Manufact.* (2015) **29**:1133–53. doi: 10.1007/s10845-015-1164-z
- Chen Z, Zhou S, Luo J. A robust ant colony optimization for continuous functions. *Exp Syst App.* (2017) **81**:309–20. doi: 10.1016/j.eswa.2017.03.036
- Li G, Cui L, Fu X, Wen Z, Lu N, Lu J. Artificial bee colony algorithm with gene recombination for numerical function optimization. *Appl Soft Comput.* (2017) **52**:146–59. doi: 10.1016/j.asoc.2016.12.017
- Margreitter C, Oostenbrink C. Correction to optimization of protein backbone dihedral angles by means of hamiltonian reweighting. *J Chem Inform Model.* (2018) **58**:1716–20. doi: 10.1021/acs.jcim.8b00470
- Palma LB, Coito FV, Ferreira BG. PSO based on-line optimization for DC motor speed control. In: *2015 9th International Conference on Compatibility and Power Electronics (CPE)*. Costa da Caparica (2015).
- Sheng-Feng C, Xiao-Hua C, Lu Y. Application of wavelet neural network with improved PSO algorithm in power transformer fault diagnosis. *Pow Syst Protect Control.* (2014) **42**:37–42. Available online at: https://www.researchgate.net/publication/286059512_Application_of_wavelet_neural_network_with_improved_particle_swarm_optimization_algorithm_in_power_transformer_fault_diagnosis
- Prativa A, Sumitra M. Efficient player selection strategy based diversified PSO algorithm for global optimization. *Inform Sci.* (2017) **397**–8:69–90. doi: 10.1016/j.ins.2017.02.027
- Jiang X, Ling H. A new multi-objective particle swarm optimizer with fuzzy learning sub-swarms and self-adaptive parameters. *Adv Sci Lett.* (2012) **7**:696–9. doi: 10.1166/asl.2012.2720
- Clerc M. The swarm and the queen: towards a deterministic and adaptive PSO. *IEEE Congress on Evolutionary Computation-CEC.* (2002) **3**:1951–7. doi: 10.1109/cec.1999.785513
- Tsubouchi M, Momose T. Rovibrational wave-packet manipulation using shaped mid-infrared femtosecond pulses toward quantum computation: optimization of pulse shape by a genetic algorithm. *Phys Rev A.* (2008) **77**:052326. doi: 10.1103/PhysRevA.77.052326
- Jiang W, Carter DR, Fu HL, Jacobson MG, Zipp KY, Jin J, et al. The impact of the biomass crop assistance program on the United States forest products market: an application of the global forest products model. *Forests.* (2019) **10**:1–12. doi: 10.3390/f10030215
- Liu JB, Zhao J, Zhu ZX. On the number of spanning trees and normalized laplacian of linear octagonal-quadrilateral networks. *Int J Quant Chem.* (2019) **119**:1–21. doi: 10.1002/qua.25971
- Liu JB, Zhao J, Cai Z. “On the generalized adjacency, laplacian and signless laplacian spectra of the weighted edge corona networks. *Phys A.* (2020) **540**:1–11. doi: 10.1016/j.physa.2019.123073
- Liu W, Li J, Ren L, Xu J, Li C, Li S. Exploring livelihood resilience and its impact on livelihood strategy in rural China. *Soc Indicat Res.* (2020) **150**:977–98. doi: 10.1007/s11205-020-02347-2

AUTHOR CONTRIBUTIONS

XX and SL participated in the preparation and presentation of the manuscript.

FUNDING

This work was supported by the Science and Technology Research Program of Chongqing Municipal Education Commission (Grant No. KJQN201902103) and the Open Fund of Chongqing Key Laboratory of Spatial Data Mining and Big Data Integration for Ecology and Environment (No. ZKPT0120193007).

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Copyright © 2020 Xia and Li. This is an open-access article distributed under the terms of the Creative Commons Attribution License (CC BY). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.