Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 10, October 2019, pp. 2636-2644
DOI: 10.23940/ijpe.19.10.p9.26362644

An Adaptive Cooperative Dual Particle Swarm Optimization Algorithm with Chaotic Mutation and Quantum Behavior

Tianfei Chen[a,b], Lijun Sun[a,b,*], Xiaodong Song[a,b], and Haixu Niu[c]

[a]*School of Electrial Engineering, Henan University of Technology, Zhengzhou, 450001, China*
[b]*Zhengzhou Key Laboratory of Machine Perception and Intelligence Systems, Zhengzhou, 450001, China*
[c]*Belarusian State Pedagogical University, Minsk, 220030, Belarus*

Abstract

An adaptive cooperative dual particle swarm optimization algorithm with chaotic mutation and quantum behavior is proposed to solve the contradiction between global search and local refinement search for basic particle swarm optimization algorithms. The strategy of adaptive cooperative evolution for two subgroups is used to parallel search, the subgroup with the chaotic mutation operator modifies the historical optimal position of particles and the subgroup optimal position using the principle of chaotic randomly ergodicity, and the chaotic mutation radius is increasing with the iterative evolution to enhance the global search ability. Additionally, in order to improve the local refinement search ability, the subgroup with quantum behavior, which casts off the searching orbital, updates the average optimal position of the subgroup and the subgroup optimal position during evolution. Finally, the numerical simulation results demonstrate that the proposed algorithm not only has fast convergence speed and high convergence accuracy, but also has significant advantages in dimension expansion.

Keywords: particle swarm optimization; chaotic mutation; quantum behavior; adaptive cooperative

(Submitted on June 20, 2019; Revised on July 16, 2019; Accepted on August 10, 2019)

1. Introduction

Basic particle swarm optimization (PSO), which was proposed by Kennedy [1], is an optimization evolution algorithm based on swarm intelligence. It is one of the most popular algorithms in the domain of evolutionary computation, with characteristics of simple optimization mechanism, fast convergence rate, few parameters, and easy implementation. However, the diversity of the swarm population is decreasing in the process of evolution, and it is easy for PSO to fall into local optimal positions, which may lead to the phenomenon of prematurity. Therefore, some improved PSO variants for improving the optimization performance have been put forward to further expand the application areas.

Some scholars have improved the optimization performance of PSO just by adjusting corresponding control parameters. The inertial weight was introduced by Shi [2]. It was adjusted linearly (or nonlinearly) according to the iteration number, so that the search ability of the algorithm could be changed at different evolutionary stages. Similarly, Juang [3] used the fuzzy set theory to adjust the two cognitive coefficients dynamically. Considering the premature of PSO, some auxiliary operators, which help particles jump out of local optimal positions, have been designed to strengthen the global search ability. For example, Alatas [4] proposed a chaos embedded particle swarm optimization algorithm, and the selection operator [5] and the crossover operator were introduced to effectively improve the convergence of PSO. Xin [6] combined PSO with differential evolution (DE). Moreover, Han [7] proposed diversity-guided hybrid particle swarm optimization, which integrates gradient information and attraction-exclusion mechanism. In order to solve the complex multimodal problems, Liang [8] proposed a comprehensive learning method of particles, which can enable the algorithm to have a better diversity preservation mechanism. The ability of the above algorithms to jump out of local optimal positions is enhanced; nevertheless, the local refinement search is inadequate to an extent. In order to avoid prematurity and enhance local refinement optimization, Chen [9] introduced a local search operator to give each particle the ability of self-improvement.

* Corresponding author.
E-mail address: sunlijunzz@163.com

Similarly, Xia [10] used the difference from better particles to help the optimal particles learn and search locally. According to the ideal of cultivating outstanding individuals, Wang [11] perturbed the optimal solution of population at dynamic probability to improve the ability of exploiting solution space.

Aiming at solving the contradiction between the global search and the local refinement search for PSO, this paper proposes an adaptive cooperative dual particle swarm optimization algorithm with chaotic mutation and quantum behavior (ACPSO). The proposed algorithm is composed of two subgroups, and the optimal position of the two subgroups interacts with each other according to the probability changed with the iteration number. One subgroup carries out chaotic perturbation on the historical optimal position of particles and the subgroup optimal position to enhance the global search ability, and the other subgroup, which makes full use of the properties of quantum behavior, sets the average optimal position and randomly corrects the corresponding subgroup optimal solution to strengthen the local search ability. The final simulation experimental results demonstrate that the proposed algorithm not only has fast convergence rate and high convergence accuracy, but also has significant advantages in dimension expansion.

2. Basic PSO

2.1. Algorithm Description

PSO is a swarm intelligence stochastic search algorithm that simulates birds' flying and foraging behavior to find the global optimal solution through individual cooperation. We assume that the population consists of M particles in the D-dimensional space, $\boldsymbol{p}_i$ indicates the historical optimal position of the i^{th} particle, $i \in \{1, 2, \cdots, M\}$, and the optimal position of the population is represented by $\boldsymbol{p}_g$. In every iteration step, the velocity and position of all the particles are dynamically updated by tracking the historical optimal solution and the optimal solution of the population. We use v_i and x_i respectively to show the velocity and position of the i^{th} particle, so the updated formula can be expressed as

$$v_{i,d}(t+1) = \omega \cdot v_{i,d}(t) + c_1 \cdot r_1 \cdot (p_{i,d}(t) - x_{i,d}(t)) + c_2 \cdot r_2 \cdot (p_{g,d}(t) - x_{i,d}(t)) \tag{1}$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \tag{2}$$

Where t is the iteration number; d denotes each dimension, and $d \in \{1,2,\cdots,D\}$; $x_{i,d}(t)$ is the d^{th} dimensional variable of the i^{th} particle in the iteration number t, and $v_{i,d}(t)$, $p_{g,d}(t)$, and $p_{i,d}(t)$ also have similar meanings; ω represents the inertial weight; c_1 and c_2 are respectively indicated as the individual cognitive parameter and the swarm cognitive parameter; and r_1 and r_2 are both random numbers uniformly distributed in the interval [0,1].

2.2. Convergence Analysis

Theoretical analysis for the movement trajectories of particles and the stability of PSO can provide important guidance for performance improvements or parameter selection.

In terms of the evolution Equation (1) of PSO, we define $\varphi_1 = c_1 \cdot r_1$, $\varphi_2 = c_2 \cdot r_2$, and then the formula can be transformed into

$$v_{i,d}(t+1) = \omega \cdot v_{i,d}(t) + \varphi_1 \cdot p_{i,d}(t) + \varphi_2 \cdot p_{g,d}(t) - (\varphi_1 + \varphi_2) \cdot x_{i,d}(t) \tag{3}$$

When $t \to \infty$, $v(t) = 0$, and $v(t+1) = 0$, according to Equation (3), we can conclude that

$$\lim_{t \to \infty} x_d^* = \frac{\varphi_1 \cdot p_{i,d} + \varphi_2 \cdot p_{g,d}}{\varphi_1 + \varphi_2} = (1-\alpha) \cdot p_{i,d} + \alpha \cdot p_{g,d} \tag{4}$$

Where α is a random number uniformly distributed in [0,1].

It can be seen from Equation (4) that the evolution of particles will be continuously gathered to the convergence point

x_d^* for each dimension of the population, and the point x_d^* is determined by $p_{i,d}$ and $p_{g,d}$. Reference [12] has already pointed out that the convergence of particle trajectory is determined by the control parameters of PSO, and the sufficient condition for the mean square convergence is supplied as follows:

$$\begin{cases} 0 < \mu < 2(\omega + 1) \\ |\omega| < 1 \\ \underline{\omega} < \omega < \overline{\omega} \end{cases} \tag{5}$$

Where $\mu = E(\varphi(t))$ and $\varphi = \varphi_1 + \varphi_2$. The absolute value of inertial weight ω is less than 1 and also in the interval $[\underline{\omega}, \overline{\omega}]$, and the formula for calculating the upper and lower bounds of the above interval is expressed as

$$\underline{\omega} = \frac{\mu^2 - \sigma^2 - \sqrt{(\mu^2 - \sigma^2)^2 - 8\mu((\mu - 1)^2 + \sigma^2 - 1)}}{4\mu} \tag{6}$$

$$\overline{\omega} = \frac{\mu^2 - \sigma^2 + \sqrt{(\mu^2 - \sigma^2)^2 - 8\mu((\mu - 1)^2 + \sigma^2 - 1)}}{4\mu} \tag{7}$$

Where $\sigma^2 = D(\varphi(t))$. Generally, we make $c_1 = c_2$, and thus we can have $\mu = (c_1 + c_2)/2 = c_1 = c_2$ and $\sigma^2 = (c_1^2 + c_2^2)/12$.

3. Our Proposal: ACPSO

In order to comprehensively balance the global search and the local refinement performance of PSO, an adaptive cooperative dual particle swarm optimization algorithm with chaotic mutation and quantum behavior is proposed in this paper. The specific algorithm improvement includes three aspects: the global search subgroup with the chaotic mutation operator, the local refinement subgroup with quantum behavior, and the adaptive cooperative strategy.

3.1. The Global Search Subgroup with Chaotic Mutation Operator

It can be seen that the convergence positions of particles are determined by both the optimal position of population $\boldsymbol{p}_g$ and the historical optimal of particles $\boldsymbol{p}_i$. When PSO starts running, the particles in the population will gradually gather as the number of iterations increases, and the diversity of the population will begin to decrease. In order to improve the global exploration ability of PSO and break away from the local optimal value, the chaotic mutation operator is introduced. It can guide particles to explore and move to new search space, and thus the probability of finding the global optimal solution can be improved.

Chaos [4] is a common phenomenon in nonlinear systems. Generated chaotic sequences can traverse all states without repetition in a certain range. Thus, chaos has the advantage of being global and has become a very effective search tool. In this paper, the simplified logistic function describing the evolution of biological population is used.

$$z(t+1) = \lambda \cdot z(t) \cdot (1 - z(t)) \tag{8}$$

Where $t = 1, 2, \cdots$, the initial value of sequence $z(0) \in (0,1)$, and λ is the chaotic control parameter. The logistic map is in a completely chaotic state when $\lambda = 4$.

The parameter *Stagnation* is defined, which is regarded as the number of stagnation for the global optimal value. The chaotic mutation operation will be executed on the optimal position of the subgroup and all the historical optimal values of particles when the *Stagnation* is larger than the pre-set value *limit*. The detailed steps of chaotic mutation operation are as follows:

Step 1 An initial *D*-dimensional vector $Z_0 = [z_{0,1}, z_{0,2}, \cdots, z_{0,d}, \cdots, z_{0,D}]$ is randomly generated, $z_{0,d} \in (0,1)$, and the

vector Z_0 is used as the initial value of the chaotic iteration. The chaotic sequence $\{Z(k)\}$ can be formed according to Equation (8), $k = 1, 2, \cdots, C_{\max}$, and $C_{\max}$ is the maximum number of chaotic sequences.

Step 2 The chaotic sequence is transformed into a range of variables.

$$x_d^k = x_{\min,d} + z_d^k \cdot (x_{\max,d} - x_{\min,d}) \tag{9}$$

Where $x_{\min,d}$ and $x_{\max,d}$ are respectively the lower bound and the upper bound of variables in the d^{th} dimension and z_d^k represents the d^{th} dimension of the k^{th} value in the chaotic sequence.

Step 3 Chaotic mutation for the optimal position of the subgroup and the historical optimal positions of particles. Let ξ be the variable before chaotic mutation and ξ^* represent the variable after chaotic mutation. Then, the chaotic mutation operation is based on the following expression:

$$\xi^* = (1-\beta) \cdot \xi + \beta \cdot x_d^k \tag{10}$$

Where β is the chaotic search radius and $0 < \beta < 1$. The chaotic search radius is made to increase with the iteration number of evolution. In the early stage of evolution, the swarm diversity is high, so the chaotic search radius is relatively small. However, the chaotic search radius gradually increases in the later stage of evolution, which is conducive to the algorithm to jump out of the local optimal value. The chaotic search radius is defined as

$$\beta = \frac{floor\left(\frac{it}{Maxit} \cdot 9 + 1\right)}{10} \tag{11}$$

Where *it* represents the current iteration number of population evolution, *Maxit* is the maximum iteration number of population evolution, and *floor*() is the downward integral function.

Step 4 The chaotic mutation variable is compared with the original value. If the optimal value of the chaotic sequence is better than the original value, the original value should be replaced by the optimal value of the chaotic sequence, and the step number of stagnation for the global optimal value needs to be reset to zero at the same time.

3.2. The Local Refinement Subgroup with Quantum Behavior

According to the convergence analysis of PSO, the particles take the attractor as the center, which is determined by both the optimal position of population $\boldsymbol{p}_g$ and the historical optimal position of particle $\boldsymbol{p}_i$. The particles will gradually approach the corresponding attractors until final convergence, as the velocity of the particles decreases. However, the convergence of particles for the basic PSO is achieved in the form of orbits, and the velocities of particles are always limited. Thus, the evolutionary orbit of particles cannot completely cover the search space centered on the attractors, which limits the local refinement ability of the algorithm and prevents the optimal solution from being found.

In reference [13], quantum-behaved particle swarm optimization (QPSO) was proposed. It allows each particle to behave in accordance with quantum dynamics, and each particle moves in a delta potential well centered at its local attractor. In this way, particles can appear at any point in the search space according to a certain probability, and thus the local refinement ability of PSO can be enhanced. The updating equation of particle position for QPSO is expressed as

$$x_{i,d}(t+1) = q_{i,d}(t) \pm \chi \cdot \left| m_{best,d}(t) - x_{i,d}(t) \right| \cdot \ln(1/u) \tag{12}$$

Where u is a random number uniformly distributed in the interval [0,1], χ is the search expansion factor, $q_{i,d}(t)$ represents the d^{th} dimension value of the attractor corresponding to particle i in the t^{th} iteration, $m_{best,d}(t)$ is the average optimal position in the t^{th} iteration, and $\left| m_{best,d} - x_{i,d} \right|$ determines the search range of particle i in the d^{th} dimension.

The local attractors of particles and the average optimal position of this subgroup are respectively calculated as follows:

$$q_{i,d}(t) = (1-\alpha) \cdot p_{i,d}(t) + \alpha \cdot p_{g,d}(t) \tag{13}$$

$$m_{best,d}(t) = \frac{1}{M} \sum_{i=1}^{M} p_{i,d}(t) \tag{14}$$

Where M indicates the total particle number of the population and α is a random number uniformly distributed in the interval [0,1].

In order to improve the local refinement ability of QPSO, the calculation method of the average optimal position is improved in this paper, and the optimal position of this subgroup is randomly modified at the same time.

3.2.1. Calculation of the Average Optimal Position

To strengthen the local refinement ability of the algorithm, the average optimal position of the population should change with the iteration number of evolution. All the historical optimal position of particles in this subgroup should be sorted according to their fitness value at each evolution step, and the historical optimal values of particles with different numbers are selected in order for different iteration numbers. Then, the average optimal position of population can be calculated accordingly.

The number of the historical optimal position of particles N_p can be regarded as a function of the population iteration *it*. The equation is as follows:

$$N_p = ceil\left(\frac{it}{Maxit} \cdot M \right) \tag{15}$$

Where $1 \le N_p \le M$, *ceil*() represents the top integer function.

In the early stage of evolution, the diversity of the population is relatively high, so the calculation of the average optimal position depends on some historical optimal position of particles with higher fitness. As the iteration number increases, the particles of the population will gradually gather; thus, more historical optimal positions of particles with better fitness will be selected for the calculation of the average optimal position.

3.2.2. Random Modification for the Optimal Position

The dimension increase in the optimization problem can produce more local optimal extremes and simultaneously affect the optimization performance of the algorithm. Thus, the random modification for the optimal position is carried out to further improve the convergence accuracy.

First, the normalization of each dimension should be performed according to the corresponding value range, so that the value range of each dimension is limited to the interval [0,1]. Then, the random modification for the optimal position will be carried out, and the pseudo-code of the specific steps is as follows:

for j = 1, 2, ⋯, D
  $\boldsymbol{t}_1 = \boldsymbol{t}_2 = \boldsymbol{p}_g$;
  $avg(\boldsymbol{p}_g) = (\min(\boldsymbol{p}_g) + \max(\boldsymbol{p}_g)) / 2$;
  $\boldsymbol{t}_1(j) = \min(\boldsymbol{p}_g) + rand() * (avg(\boldsymbol{p}_g) - \min(\boldsymbol{p}_g))$;
  $\boldsymbol{t}_2(j) = avg(\boldsymbol{p}_g) + rand() * (\max(\boldsymbol{p}_g) - avg(\boldsymbol{p}_g))$;
  if $f(\boldsymbol{t}_1) < f(\boldsymbol{p}_g)$ **and** $f(\boldsymbol{t}_1) < f(\boldsymbol{t}_2)$
    $\boldsymbol{p}_g(j) = \boldsymbol{t}_1(j)$;
  elseif $f(\boldsymbol{t}_2) < f(\boldsymbol{p}_g)$ and $f(\boldsymbol{t}_2) \le f(\boldsymbol{t}_1)$

$\boldsymbol{p}_g(j) = \boldsymbol{t}_2(j)$;

end

end

Where $\min(\boldsymbol{p}_g)$ is the minimum value of all dimensions for the optimal position of the population $\boldsymbol{p}_g$, $\max(\boldsymbol{p}_g)$ represents the maximum value of all dimensions, $avg(\boldsymbol{p}_g)$ indicates the average value between $\min(\boldsymbol{p}_g)$ and $\max(\boldsymbol{p}_g)$, and $f()$ means the objective function.

The random modification is carried out at each iteration, so that the optimal position of the population can be updated from the first dimension to the D^{th} dimension. Therefore, it is more likely to search for a better solution and thus improve the convergence accuracy.

3.3. Adaptive Cooperative Strategy

The global search subgroup with the chaotic mutation operator helps jump out of the local optimal extremum and enhance the global search ability, and the local refinement subgroup with quantum behavior can further improve the optimization accuracy. In order to balance the global exploration ability and the local development ability of PSO, the above two subgroups are organized for cooperative evolution.

The adopted cooperative strategy compares the optimal solutions from the two subgroups, and the better one between the two is regarded as the global optimal solution and should replace the relatively poor solution. However, the frequency of cooperation will also influence the optimization performance of the algorithm. An adaptive cooperative control function changed with iteration numbers is designed, and its expression is as follows:

$$C(it) = \frac{it}{Maxit} \tag{16}$$

Where *it* is the current iteration number and *Maxit* is the maximum iteration number. Consequently, $0 < C(it) < 1$.

The cooperative control parameter p_c is designed as a random number uniformly distributed in the interval [0,1]. If $p_c < C(it)$, the cooperative strategy will be carried out. Otherwise, the two subgroups will continue to run independently. In this strategy, the control function value is relatively small in the early stage of evolution, so the operation of cooperative evolution is not easy to occur, and the search advantages of the two subgroups can be fully used. Besides that, the cooperative operation is more likely to happen at a higher probability in the late evolution stage, and the two subgroups have already been running independently for a period of time; thus, the cooperative strategy can take full advantage of the results from the two subgroups so as to find the global optimal solution.

4. Experimental Studies

4.1. Benchmark Functions and Parameter Setting

In this paper, six different benchmark functions are selected. Table 1 shows the detailed expression of all functions, the corresponding search range, and the global optimal value. Sphere and Rosenbrock are both unimodal functions, and the other benchmark functions are multimodal and have many local extrema.

The ACPSO is compared with basic PSO and QPSO, the population size is M = 50, the maximum iteration number is *Maxit* = 3000, and the other parameters for each algorithm are set as follows: for PSO, the inertial weight is ω = 0.7 and both cognitive parameters are $c_1 = c_2 = 2.0$; for QPSO, the search expansion factor is χ = 1.0, the attractors are calculated according to Equation (13), and Equation (14) is used to compute the average optimal position; for ACPSO, the pre-set limit in the global search subgroup is *limit* = 10, the other remaining parameters in the global search subgroup are the same as those of PSO, the search expansion factor in the local refinement subgroup is χ = 0.8, and the calculation of the average optimal position and the random modification of the optimal position in the local refinement subgroup are executed according to Section 3.2.

Table 1. Benchmark functions

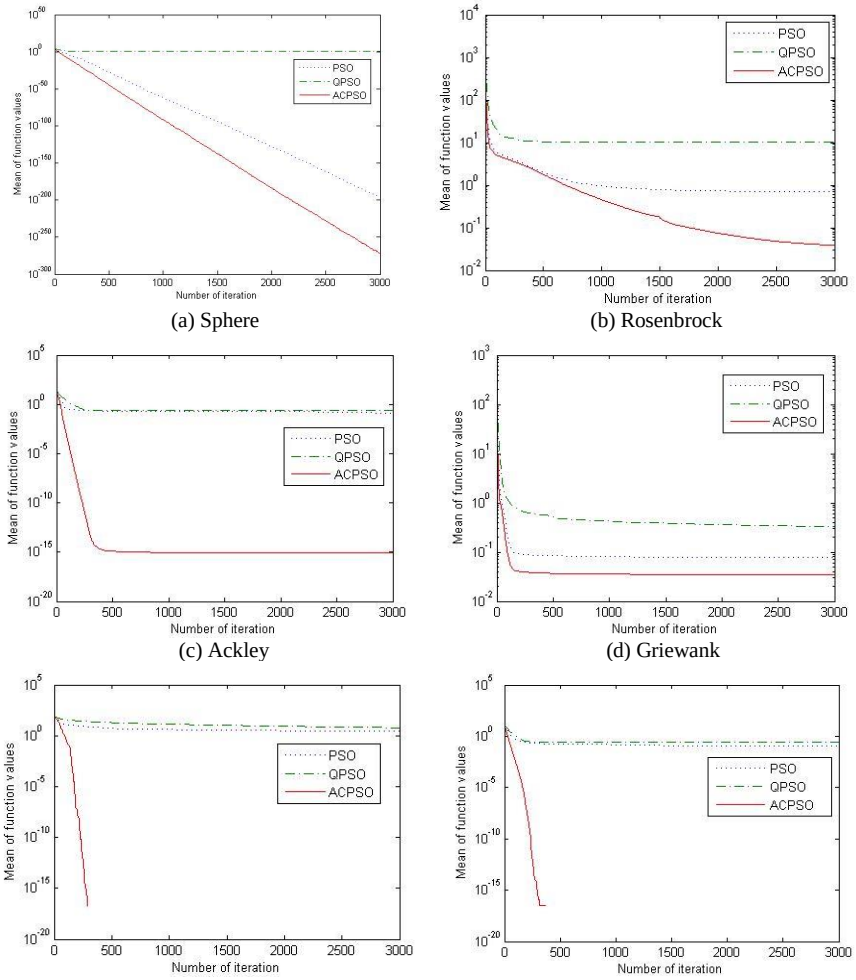
Function name	Function expression	Search range	Global optimal value	Function properties
Sphere	$f_1(\mathbf{X}) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	0	UniModal
Rosenbrock	$f_2(\mathbf{X}) = \sum_{i=1}^{D-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-2.048, 2.048]^D$	0	UniModal
Ackley	$f_3(\mathbf{X}) = \mathrm{e} + 20 - 20\exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - \exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i))$	$[-32.768, 32.768]^D$	0	MultiModal
Griewank	$f_4(\mathbf{X}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$	0	MultiModal
Rastrigin	$f_5(\mathbf{X}) = \sum_{i=1}^{D}(x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	0	MultiModal
Weierstrass	$f_7(\mathbf{X}) = \sum_{i=1}^{D}\{\sum_{k=0}^{k_{\max}}[a^k \cos(2\pi b^k (x_i + 0.5))]\} - D\sum_{k=0}^{k_{\max}}[a^k \cos(2\pi b^k \cdot 0.5)]$, $a = 0.5,\ b = 3,\ k_{\max} = 20$	$[-0.5, 0.5]^D$	0	MultiModal

4.2. Comparison of Performance

The convergence performance of algorithms mainly includes two aspects: the convergence rate and the optimization accuracy. Firstly, the dimensions of all benchmark functions are set to 10, and the three algorithms are run 50 times independently on each benchmark function. The final statistical convergence results of each algorithm are shown in Table 2. In the table, **Mean** is the average optimization results. **Std** is the standard deviation of multiple optimization results, which can be used to measure the optimization stability of algorithms. From the results, it can be seen that the optimization accuracy of ACPSO has significant advantages. Figure 1 shows the convergence curves of algorithms on each benchmark function, where the abscissa axis presents the number of iterations and the ordinate axis indicates the average results of multiple running. From the convergence curves of algorithms for all benchmark functions, we can see that QPSO has the worst convergence curve, and the main reason is that QPSO focuses on local searching. The diversity of population for QPSO will gradually decrease as the iteration increases, which may lead to premature convergence. Because of the balance between local refinement and global searching in ACPSO, the experimental results demonstrate that the optimization accuracy of ACPSO is significantly better than that of PSO and QPSO, and the convergence rate is further improved.

Table 2. Comparison of optimization accuracy for different algorithms

Function	Algorithm	10D		50D	
		Mean	**Std**	**Mean**	**Std**
Sphere	PSO	5.4943E-197	0	2.6999e+04	1.9311e+04
	QPSO	7.4264	50.4849	6.9164e+03	6.6859e+03
	ACPSO	**8.3639E-273**	**0**	**61.1125**	**236.4355**
Rosenbrock	PSO	0.6924	1.5263	3.2328e+03	1.7163e+03
	QPSO	10.3695	15.5435	856.8981	477.8899
	ACPSO	**0.0383**	**0.2792**	**176.5323**	**230.6955**
Ackley	PSO	0.1356	0.6367	17.4200	1.4179
	QPSO	0.2398	0.5036	10.3916	1.9315
	ACPSO	**8.8818e-16**	**0**	**0.4601**	**1.8762**
Griewank	PSO	0.0764	0.0375	249.8660	137.3768
	QPSO	0.3277	1.2655	54.9508	44.3286
	ACPSO	**0.0341**	**0.0220**	**1.0518**	**4.2017**
Rastrigin	PSO	2.9550	2.1654	275.0553	106.0819
	QPSO	6.1974	4.2022	498.4858	34.3707
	ACPSO	**0**	**0**	**110.2393**	57.1151
Weierstrass	PSO	0.1045	0.2876	48.4562	9.7018
	QPSO	0.2752	0.6379	29.3781	4.4829
	ACPSO	**0**	**0**	**0.2986**	**1.4786**



(a) Sphere (b) Rosenbrock

(c) Ackley (d) Griewank

(e) Rastrigin (f) Weierstrass

Figure 1. Convergence curves of all benchmark functions

The increase in the dimensions of the optimization problem will cause a dramatic decline in the optimization performance. In order to further investigate the performance of algorithms, the dimension of each benchmark function is increased to 50, and the three algorithms are run 50 times independently. From Table 2, we can see that the optimization performance of PSO decreases dramatically as the dimensions increase, but the performance of QPSO is better than that of PSO for most benchmark functions. For the optimization problem with high dimensions, ACPSO still has obvious higher optimization accuracy than PSO and QPSO, and the average optimization result of ACPSO is the best on all benchmark functions.

In summary, ACPSO uses the subgroup with chaotic mutation operator to realize global search and maintains local refinement ability by the subgroup with quantum behavior, and then an adaptive strategy is designed to balance the two subgroups. It not only has a fast convergence rate, but also ensures higher optimization accuracy. In addition, ACPSO also maintains a significant advantage in dimension expansion.

5. Conclusions

An adaptive cooperative dual particle swarm optimization algorithm with chaotic mutation and quantum behavior is proposed in this paper, and the proposed algorithm is composed of two subgroups. The two subgroups work together, and an adaptive cooperative strategy is used at the same time. The global search subgroup executes the chaotic mutation operation on the optimal position of the subgroup and the historical optimal positions of particles based on the randomness and ergodicity of chaotic system. This helps jump out of the local extremum and thus enhance the global search ability. In addition, the local refinement subgroup breaks through the limitation of orbit search and make full use of the advantage of quantum behavior, so that particles in this subgroup can appear at any position with a certain probability, and then the local refinement ability is improved. Finally, the numerical simulations on different benchmark functions demonstrate that the

proposed algorithm not only has a fast convergence rate but also ensures higher optimization accuracy, and the contradiction between global search and local refinement can be solved properly. In addition, ACPSO also maintains a significant advantage in dimension expansion.

Acknowledgements

The authors sincerely thank the editors and reviewers for their valuable suggestions and useful comments that have led to the present improved version of the original manuscript. This work was supported in part by the National Natural Science Foundation of China (No. U1604151, 61803146), Outstanding Talent Project of Science and Technology Innovation in Henan Province (No. 174200510008), Science and Technology Project of Henan Province (No. 182102210094), Natural Science Project of Education Department of Henan Province (No. 18A510001), and Fundamental Research Funds of Henan University of Technology (No. 2015QNJH13, 2016XTCX06).

References

1. R. C. Eberhart and J. Kennedy, "A New Optimizer using Particle Swarm Theory," in *Proceedings of International Symposium on Micro Machine and Human Science*, pp. 39-43, 2002
2. Y. Shi and R. C. Eberhart, "Empirical Study of Particle Swarm Optimization," *IEEE Congress on Evolutionary Computation*, Vol. 3, pp. 1945-1950, 1999
3. Y. T. Juang, S. L. Tung, and H. C. Chiu, "Adaptive Fuzzy Particle Swarm Optimization for Global Optimization of Multimodal Functions," *Information Sciences*, Vol. 181, No. 20, pp. 4539-4549, 2011
4. B. Alatas, E. Akin, and A. B. Ozer, "Chaos Embedded Particle Swarm Optimization Algorithms," *Chaos Solitons and Fractals*, Vol. 40, No. 4, pp. 1715-1734, 2009
5. P. J. Angeline, "Using Selection to Improve Particle Swarm Optimization," in *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 84-89, 1998
6. B. Xin, J. Chen, Z. H. Peng, and F. Pan, "An Adaptive Hybrid Optimizer based on Particle Swarm and Differential Evolution for Global Optimization," *Science China* (*Information Sciences*), Vol. 53, No. 5, pp. 980-989, 2010
7. F. Han and Q. Liu, "A Diversity-Guided Hybrid Particle Swarm Optimization," *Neurocomputing*, Vol. 137, No. 4, pp. 234-240, 2014
8. J. Liang, K. Qin, P. N. Suganthan, and B. Subramanian, "Comprehensive Learning Particle Swarm Optimizer for Global Optimization of Multimodal Functions," *IEEE Transactions on Evolutionary Computation*, Vol. 10, No. 3, pp. 281-295, 2006
9. J. Y. Chen, Z. Qin, Y. Liu, and J. Lu, "Particle Swarm Optimization with Local Search," in *Proceedings of International Conference on Neural Networks and Brain*, pp. 481-484, 2005
10. X. W. Xia, J. N. Liu, K. F. Gao, Y. X. Li, and H. Zeng, "Particle Swarm Optimization Algorithm with Reverse-Learning and Local-Learning Behavior," *Journal of Software*, Vol. 38, No. 7, pp. 1397-1407, 2015
11. H. Wang, H. B. Ouyang, and L. Q. Gao, "An Improved Global Particle Swarm Optimization," *Control and Decision*, Vol. 31, No. 7, pp. 1161-1168, 2016
12. X. L. Jin, L. H. Ma, T. J. Wu, and J. X. Qian, "Convergence Analysis of the Particle Swarm Optimization based on Stochastic Processes," *Acta Automatica Sinica*, Vol. 33, No. 12, pp. 1263-1268, 2007
13. W. Fang, J. Sun, Y. R. Ding, X. J. Wu, and W. B. Xu, "A Review of Quantum-Behaved Particle Swarm Optimization," *IETE Technical Review*, Vol. 27, No. 4, pp. 336-348, 2010

Tianfei Chen received his B.S., M.S., and Ph.D. degrees from Dalian Maritime University in 2008, 2010, and 2013, respectively. He is currently an assistant professor at Henan University of Technology. His main research interests include swarm intelligence and sensor networks.

Lijun Sun received her B.S. degree from Xidian University in 1989, her M.S. degree from Hefei University of Technology in 2001, and her Ph.D. from Northwestern Polytechnical University in 2005. She is currently a professor in the School of Electrical Engineering at Henan University of Technology. Her research interests include artificial intelligence, wireless sensor networks, computational intelligence, and imaging processing.