

## Dual sub-swarm interaction QPSO algorithm based on different correlation coefficients

Tao Wu, Lei Xie, Xi Chen & Jia He

To cite this article: Tao Wu, Lei Xie, Xi Chen & Jia He (2017) Dual sub-swarm interaction QPSO algorithm based on different correlation coefficients, *Automatika*, 58:4, 375-383, DOI: 10.1080/00051144.2018.1454732

To link to this article: <https://doi.org/10.1080/00051144.2018.1454732>



© 2018 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 03 May 2018.



Submit your article to this journal [↗](#)



Article views: 278



View related articles [↗](#)



View Crossmark data [↗](#)



## Dual sub-swarm interaction QPSO algorithm based on different correlation coefficients

Tao Wu<sup>a</sup>, Lei Xie<sup>a</sup>, Xi Chen<sup>b</sup> and Jia He<sup>a</sup>

<sup>a</sup>Department of Computer Science, Chengdu University of Information Technology, Chengdu, Sichuan, China; <sup>b</sup>School of Computer Science and Technology, Southwest University for Nationalities, Chengdu, Sichuan, China

### ABSTRACT

A novel quantum-behaved particle swarm optimization (QPSO) algorithm, the dual sub-swarm interaction QPSO algorithm based on different correlation coefficients (DCC-QPSO), is proposed by constructing master-slave sub-swarms with different potential well centres. In the novel algorithm, the master sub-swarm and the slave sub-swarm have different functions during the evolutionary process through separate information processing strategies. The master sub-swarm is conducive to maintaining population diversity and enhancing the global search ability of particles. The slave sub-swarm accelerates the convergence rate and strengthens the particles' local searching ability. With the critical information contained in the search space and results of the basic QPSO algorithm, this new algorithm avoids the rapid disappearance of swarm diversity and enhances searching ability through collaboration between sub-swarms. Experimental results on six test functions show that DCC-QPSO outperforms the traditional QPSO algorithm regarding optimization of multimodal functions, with enhancement in both convergence speed and precision.

### ARTICLE HISTORY

Received 12 October 2016  
Accepted 7 February 2018

### KEYWORDS

Particle swarm optimization;  
quantum-behaved particle  
swarm optimization; sub-  
swarm; interaction

## Introduction

Optimization techniques are a mathematically-based technology used to search for optimal or satisfactory solutions to various engineering problems [1]. The Particle Swarm Optimization algorithm (PSO) proposed by James Kennedy and Russell Eberhart [2] is a recent implementation of these techniques. The essential idea of PSO is to emulate the flocking behaviour of birds [3]. Initially, PSO is given with a population of random solutions, and then it searches for optima by updating generations [4]. Unlike some other meta-heuristic algorithms, the standard PSO sustains the global search strategy, avoids complex evolutionary operations and enhances convergence capability. Therefore, the PSO algorithm has drawn broad attention in various application fields. The random inertia weight Particle Swarm Optimization (RNW-PSO) is introduced for trajectory tracking of wheeled mobile robots [5] and coefficients optimizing of proportion integration differentiation (PID) controller [6]. An improved PSO algorithm is used to deal with the problem of controlling a class of uncertain nonlinear systems in the presence of external disturbances [7]. A new and powerful optimization algorithm, known as chaotic accelerated PSO (CAPSO), is used for determining the coefficients of the proportional–integral controller of Dynamic voltage restorer [8].

Nevertheless, some problems remain to be solved in the PSO algorithm. One that has been raised and proven by Bergh [9] is that PSO is not guaranteed to always converge to the global optimal solution. Although a significant amount of work [10–12] has been done in recent years to modify and improve PSO, the state-of-the-art shows that some limitations remain. Aiming at PSO's convergence bottleneck, through the comparison between the human learning processes and particles' behaviour in quantum spaces, Sun [13] proposed the quantum-behaved PSO (QPSO) algorithm, which leverages the aggregation tendency of collective intelligence in a population. In QPSO model, individuals are represented as particles in quantum space, which continuously iterate according to characteristics seen in human society, such as self-organization, collaboration, etc. Theoretical proofs have shown that QPSO is a globally convergent algorithm. Thus, it has become an active area of research in several fields [13–15].

Although the standard QPSO outperforms original PSO in search ability, its significant drawback is the prematurity. When one particle finds a local optima, the others will quickly close to it due to the single potential well centre. If the particles could not find any better locations during the process to the local optima, the algorithm will run into prematurity. In order to jump out of the local optima, some improved ways have been presented in the practical applications, such as the QPSO

with the extremum disturbed arithmetic operators [16], the chaos QPSO [17], QPSO with mutation operator [18], etc. However, these improved algorithms increase the complexity of the algorithm while avoiding prematurity through the immune mechanism. In order to further improve the performance of the QPSO algorithm without introducing complexity, this paper proposed a novel algorithm building on existing research results, which is the dual sub-swarm interaction DCC-QPSO. First, this new algorithm divides the entire population into two sub-populations equal in size. Then, each sub-group applies different processing methods in order to achieve the separation of potential well centres. The collaboration between main-group and slave-group can take advantage of effective information, avoid rapid loss of population diversity and prevent particles becoming trapped in local optima by transferring information between the two subgroups with different potential well centres. Experimental results on test functions show that DCC-QPSO outperforms the traditional QPSO algorithm in the optimization of multimodal functions, with enhancement in both convergence speed and precision.

## Binary correlation QPSO algorithm

### The standard QPSO algorithm

In a PSO optimization algorithm, the solution space is abstracted as the birds' foraging space, where each bird is abstracted as a massless and size-free particle flying at a certain speed. Each particle has a fitness value determined by the function to be optimized. According to the fitness value, a random search is carried out by each particle. In every round of iteration, each particle updates itself by tracking two optima: The first one is the optimal solution found by each particle itself, commonly referred to as the personal optimum  $p_{\text{best}}$ ; the other is the optimal solution found by the entire population, commonly referred to as the global optimum  $g_{\text{best}}$ . For simplicity, in this article,  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  and  $G = (p_{g1}, p_{g2}, \dots, p_{gD})$  are used to describe the personal optimum and global optimum of particle  $i$  in the  $D$ -dimensional search space, respectively. Particle  $i$ 's personal best position  $p_{\text{best}}$  is determined by Equation (1)

$$P_i(t) = \begin{cases} X_i(t) & \text{if } f(X_i(t)) < f(P_i(t-1)) \\ P_i(t-1) & \text{if } f(X_i(t)) \geq f(P_i(t-1)) \end{cases} \quad (1)$$

The index  $g$  of the global best position  $G = (p_{g1}, p_{g2}, \dots, p_{gD})$  is determined by Equation (2):

$$\begin{aligned} g &= \arg \min_{1 \leq i \leq D} \{f(P_i(t))\} \\ g &\in \{1, 2, \dots, D\} \end{aligned} \quad (2)$$

Reference [19] has demonstrated that standard PSO is not guaranteed to converge on the global optimum solution with probability 1, which is a major

shortcoming of the traditional PSO. In order to achieve global convergence, on the basis of previous studies of the particles' convergence behaviours, the QPSO algorithm was proposed based on the  $\delta$  potential well by representing the PSO system as a quantum space [13].

According to the analysis of particle orbits in the PSO algorithm done by Clerc and Kennedy [20], a  $\delta$  potential well can be established at the local attraction point  $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  to impact particles in the population, whose coordinate is:

$$p_{ij}(t) = \frac{c_1 r_{1,ij}(t) P_{ij}(t) + c_2 r_{2,ij}(t) G_j(t)}{c_1 r_{1,ij}(t) + c_2 r_{2,ij}(t)}, 1 \leq j \leq D \quad (3)$$

In this equation,  $r_1$  and  $r_2$  are random numbers independently distributed within the interval  $[0, 1]$ , called random factors;  $c_1$  is the individual cognitive acceleration coefficient, whereas  $c_2$  is the global cognitive acceleration coefficient.

Equation (3) can be simplified as

$$p_{ij}(t) = \varphi_{ij}(t) P_{ij}(t) + [1 - \varphi_{ij}(t)] G_j(t) \quad (4)$$

where,

$$\varphi_{ij}(t) = c_1 r_{1,ij}(t) / [c_1 r_{1,ij}(t) + c_2 r_{2,ij}(t)] \quad (5)$$

In our representation, particles are moving in a quantum space, thus the particles' states can be described using the wave function  $\Psi(X, t)$ . From the point of view of the theory of dynamics, the convergence process of a particle can be described as follows: a particle is continuously approaching the local attractor  $p_i$  with decreasing speed, and eventually overlaps with  $p_i$ . The steady-state of a particle in the  $\delta$  potential well can be expressed using the Schrödinger equation.

$$\frac{d^2 \psi}{dX^2} + \frac{2m}{\hbar^2} [E + \gamma \delta(X - p_i)] \psi = 0 \quad (6)$$

By solving the above equation, the probability distribution function in every dimension can be obtained for each particle.

$$\psi(X_{ij}) = \frac{1}{\sqrt{L}} e^{-|X_{ij} - p_{ij}|/L} \quad (7)$$

And the position-updating equation of every particle in each generation in the QPSO can be deduced as:

$$X_{ij}(t+1) = p_{ij}(t) \pm \frac{L_{ij}(t)}{2} \ln[1/u_{ij}(t)] \quad (8)$$

where  $u_{ij}(t) \sim U(0,1)$ ,  $L_{ij}(t)$  is the length of the potential well.  $L_{ij}(t)$  can be evaluated using the following equation:

$$L_{ij}(t) = 2\alpha \cdot |C_j(t) - X_{ij}(t)| \quad (9)$$

where  $C(t)$  is the average of all personal best positions, known as the gravity position of the population, and is evaluated as follows:

$$\begin{aligned} C(t) &= (C_1(t), C_2(t), \dots, C_n(t)) = \frac{1}{N} \sum_{i=1}^N P_i(t) \\ &= \left( \frac{1}{N} \sum_{i=1}^N P_{i,1}(t), \frac{1}{N} \sum_{i=1}^N P_{i,2}(t), \dots, \frac{1}{N} \sum_{i=1}^N P_{i,N}(t) \right) \end{aligned} \quad (10)$$

Therefore, a particle's position evolution equation in (6) is finally defined as follows:

$$X_{i,j}(t+1) = p_{i,j}(t) \pm \alpha \cdot |C_j(t) - X_{i,j}(t)| \ln[1/u_{i,j}(t)] \quad (11)$$

In the above equation,  $\alpha$  is referred to as the contraction-expansion coefficient, which is the only parameter in this algorithm except for the population size and iteration count. In the iteration process, convergence performance is controlled by fine-tuning  $\alpha$ . The value of  $\alpha$  can be fixed or decreased in a linear manner.

### Binary correlation QPSO algorithm

The coordinate formula of the potential well centre, which is denoted as  $p_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  can be divided into two portions

- (1) Individual cognitive component:  $\frac{c_1 r_{1ij}(t)}{c_1 r_{1ij}(t) + c_2 r_{2ij}(t)} P_{ij}(t)$ , which indicates the experiences of the particles themselves;
- (2) Social cognitive component:  $\frac{c_2 r_{2ij}(t)}{c_1 r_{1ij}(t) + c_2 r_{2ij}(t)} G_j(t)$ , which represents shared information among the particle population.

Under the combined effect of the two components above, the QPSO algorithm aims to find the optimal solution, adjusting the position of  $p_i$  constantly in the solution space according to the sharing of information and the experiences of each particle.

The coefficients  $c_1, c_2$  represent the statistic weights of the particle's acceleration, reflecting the information exchanged in the particle swarm. Setting a large  $c_1$  will cause the particles to wander in the local area because of an undue reliance on their own experience, while a large  $c_2$  will cause the particles to converge to a local optimum prematurely [19].

As they are important parameters in the standard PSO algorithm, there are a lot of related studies about how best to set the values of the acceleration factors  $c_1$  and  $c_2$  [1,12,20–24]. These policies have obtained some improvement in the PSO algorithm, however, they do not take into account the impacts of the random factors  $r_1$  and  $r_2$  on algorithm performance.

The independence assumption between  $r_1$  and  $r_2$  in the  $p_i$  formula means the algorithm cannot distinguish the utilization of  $p_{best}$  and  $g_{best}$ . At present there are few studies on the effects of parameters  $r_1$  and  $r_2$  on the algorithm. However, it is necessary to analyse the random factors in order to further study the impacts of the utilization of the particles' own experiences and community sharing information on the performance of the QPSO algorithm, respectively.

To analyse the connection between  $r_1$  and  $r_2$  in QPSO, reference [20] suggested the concept of binary correlation factors and proposed the Binary correlation QPSO algorithm, referred to as the BC-QPSO algorithm. The BC-QPSO algorithm constructed the relations between  $r_1$  and  $r_2$  using the bivariate normal Copula function:  $\Phi_\rho(\Phi^{-1}(r_1), \Phi^{-1}(r_2))$ , the Fréchet-Hoeffding lower bound:  $W(u, v) = \max(u + v - 1, 0)$ , the Fréchet-Hoeffding upper bound:  $M(u, v) = \min(u, v)$  and the product Copula:  $\Pi(u, v) = uv$ . The particle's position evolution equation of BC-QPSO is finally defined as follows:

$$\begin{cases} X_{i,j}(t+1) = p_{i,j}(t) \pm \alpha \cdot |C_j(t) - X_{i,j}(t)| \cdot \ln[1/u_{i,j}(t)], 1 \leq j \leq D \\ p_{i,j}(t) = \frac{c_1 r_{1ij}(t) P_{ij}(t) + c_2 r_{2ij}(t) G_j(t)}{c_1 r_{1ij}(t) + c_2 r_{2ij}(t)} \\ H(r_1, r_2) = C_\rho(r_1, r_2) \end{cases} \quad (12)$$

$$C_\rho(r_1, r_2) = \begin{cases} W(r_1, r_2), \rho = -1 \\ \Pi(r_1, r_2), \rho = 0 \\ M(r_1, r_2), \rho = 1 \\ \Phi_\rho(\Phi^{-1}(r_1), \Phi^{-1}(r_2)), -1 < \rho < 1, \rho \neq 0 \end{cases} \quad (13)$$

where  $H$  is the joint distribution function of the binary correlation factors  $r_1, r_2$ ;  $C$  is the binary normal Copula function;  $\rho$  is the specified correlation coefficient, which is an indicator of the relevant strength and could reflect linear correlation properties between variables  $r_1$  and  $r_2$ ;  $\Phi_\rho$  is the two-dimensional standard normal distribution function of the correlation coefficient  $\rho$ ; and  $\Phi^{-1}$  is an inverse function of the one-dimensional standard normal distribution function.

### The dual sub-swarm interaction QPSO algorithm based on different correlation coefficients

The BC-QPSO proposed by reference [20] enhances the optimization performance to some extent. However, this algorithm builds the potential well on a single position, thus the global searching ability is compromised by the optimum's attraction for particles. The evaluation pattern offered by the multi-group interaction has revealed an effective method to enhance the overall performance of the BC-QPSO. In order to

further enhance the BC-QPSO's searching ability and convergence precision, on the basis of our previous work, a novel interactive QPSO is proposed in this paper – the dual sub-swarm interaction DCC-QPSO. This new algorithm separates the single potential well into a double one, and then the entire population is divided into two sub-swarms in which different information processing strategies are adopted, respectively, to expand the search range. Meanwhile, the wait effect [15] among the particles avoids excessive accumulation of the population. Therefore, superior convergence performance is achieved for the new algorithm.

### Learning strategies

The analysis of BC-QPSO done by reference [20] shows that the diversity of the population decreases with increasing correlation coefficient  $\rho$  ( $-1 \leq \rho \leq 1$ ) of the binary correlation factors (i.e.  $r1$  and  $r2$ , seen in Equation (13)). In the BC-QPSO model, given that  $\rho = -1$ , the particle's unbalanced use of  $p_{best}$  and  $g_{best}$  in determining the position of the potential well centre deviates the potential well from  $g_{best}$ , resulting in a trend of expansion of the population, which helps extending the search space as well as population diversity. In the BC-QPSO model given that  $\rho = 1$ , particles make the most of  $p_{best}$  and  $g_{best}$  in a highly balanced manner, making the potential well rapidly move towards extremum, resulting in a quick contraction of the population, which helps to speed up the convergence. Based on the above analysis, a conclusion can be drawn that the higher the degree of correlation between the binary correlation factors  $r1$  and  $r2$ , the faster the convergence of the algorithm; conversely, the lower the degree of correlation, the greater the diversity of the population and the higher the precision of convergence. In order to guarantee the desired convergence precision and efficiency, by the combination of the fully positively correlated BC-QPSO (i.e.  $\rho = 1$ ) and the fully negatively correlated BC-QPSO (i.e.  $\rho = -1$ ), we derive a new QPSO algorithm – DCC-QPSO.

Suppose that the entire population is  $S$  and the size of the population is  $N$ ; the master group is denoted as  $S_1$ , its population size is  $N_{S_1}$ , and its global best is  $g_{best1}$ ; the slave group is  $S_2$ , its population size is  $N_{S_2}$ , and its global best is  $g_{best2}$ ; thus we have  $S_1 \cup S_2 = S$  and  $N_{S_1} + N_{S_2} = N$ .

The basic ideas of DCC-QPSO are: in a randomized and quantum-behaved particle swarm, we divide the whole population into two fully independent sub-groups. One group adopts the fully negative correlation strategy against  $p_{best}$  and  $g_{best}$  when determining its potential well, that is, we carry out an iterated search of the solution space given  $\rho = -1$  (i.e. the correlation coefficient between  $r1$  and

$r2$ ). This group is referred to as the master group  $S_1$ . The other group adopts the fully positive correlation strategy against  $p_{best}$  and  $g_{best}$  when determining its potential well, that is, we carry out an iterated search of the solution space given  $\rho = 1$ . This group is referred to as the slave group  $S_2$ .

The evolution equation of each particle in the DCC-QPSO based on the  $\delta$  potential well is as follows:

$$\begin{cases} X_{i,j}(t+1) = p_{i,j}(t) \pm \alpha_1 |C_j(t) - X_{i,j}(t)| \ln[1/u_{i,j}(t)], u_{i,j}(t) \sim U(0,1) \\ P_{i,j}(t) = \frac{c_1 r1_{i,j}(t)}{c_1 r1_{i,j}(t) + c_2 r2_{i,j}(t)} P_{i,j}(t) + \frac{c_2 r2_{i,j}(t)}{c_1 r1_{i,j}(t) + c_2 r2_{i,j}(t)} G_j(t), 1 \leq j \leq D \\ H(r1, r2) = W(r1, r2), i \in N_{S_1} \\ H(r1, r2) = M(r1, r2), i \in N_{S_2} \end{cases} \quad (14)$$

Due to the different learning strategies, the two sub-groups play different roles in the evolution of the swarm. The master group's learning strategy against existing information helps maintaining the diversity and enhancing the global searching ability of each particle. The slave group's learning strategy against existing information helps in accelerating the convergence and enhancing the local searching ability of each particle. By enabling information exchange between these two sub-groups, DCC-QPSO takes advantage of the two learning strategies and compensates for the shortcomings of each one.

### Interaction between sub-swarms

Information exchange between the two sub-groups is achieved through their respective global best fitness values. At the end of each iteration, a comparison between the fitness values corresponding to  $S_1$  and  $S_2$ 's current best position is carried out. If  $g_{best2}$  is more optimal than  $g_{best1}$ , then  $g_{best2}$  is assigned to  $g_{best1}$ ; otherwise,  $g_{best1}$  is assigned to  $g_{best2}$ . The essence of the above operation is to update the whole swarm's best position, which helps sub-groups escape from local optima. Thus, the whole swarm does not become trapped in a local optimum even if a "super individual" appears.

By the mutually complementary and collaborative evolution of the master group  $S_1$  and slave group  $S_2$ , advantageous information is utilized to prevent the swarm from having its evolution halted by settling into local optimal values. This is done without increasing the swarm size, adding parameters, complicating the algorithm or compromising convergence. In addition, the mutual collaborative iteration during the search process maintains higher diversity and global search ability, while avoiding excessive aggregation of swarm particles.



### Algorithm execution process

Based on the designs and definitions discussed above, the execution process of the DCC-QPSO algorithm is as follows:

Step 1: Parameters setting. The parameters to be set up include the individual cognitive acceleration coefficient  $c_1$ , the global cognitive acceleration coefficient  $c_2$ , the contraction-expansion factor  $\alpha$ , the swarm population size  $N$  and the maximum number of iterations  $iterMax$  or the error precision of fitness. Meanwhile, the whole population is divided into two fully independent sub-groups. The sub-group adopting the fully negative correlation (i.e. the correlation coefficient  $\rho = -1$ ) is referred to as the master group  $S_1$ , and its population size is  $N_{s1}$ ; the other sub-group adopting the fully positive correlation is referred to as the slave group  $S_2$ , and its population size is  $N_{s2}$ .

Step 2: Initialization of the population. For  $S_1$ , initialize the position of every particle in the solution space by randomly generating  $X_{i,j}(0)$  for every particle and let be the personal best position  $P_{i,j}(0) = X_{i,j}(0)$ , where  $i \in \{1, \dots, N_{s1}\}$ ,  $j \in \{1, \dots, D\}$ . For  $S_2$ , initialize the position of every particle in the solution space by randomly generating  $X_{i,j}(0)$  for every particle and let it be the personal best position  $P_{i,j}(0) = X_{i,j}(0)$ , where  $i \in \{N_{s1}+1, \dots, N\}$ ,  $j \in \{1, \dots, D\}$ ;

Step 3: Calculate the fitness values for all particles in  $S_1$  and  $S_2$ . Suppose that the optimization problem to be solved is a minimization one, and then assign the position corresponding to the smallest fitness value to the global best position of sub-groups respectively, i.e.  $g_{best1} = \{X_i | \min(f(X_i)), i \in \{1, \dots, N_{s1}\}\}$ ,  $g_{best2} = \{X_i | \min(f(X_i)), i \in \{1, \dots, N_{s2}\}\}$ .

Step 4: Calculate the average best positions of the entire population  $C(t)$  according to Equation (10) and evaluate the parameters  $L_{i,j}(t)$  of the master and slave sub-groups, respectively.

Step 5: Update the position for particle  $i$  ( $1 \leq i \leq N$ ), that is calculate new positions for all particles according to the objective function. (for  $t$  iterations).

Step 6: Update the personal best positions using Equation (1) for the master and slave sub-groups, respectively. If  $f(X_i(t)) < f(P_i(t-1))$ , then let be  $P_i(t) = X_i(t)$ ; otherwise,  $P_i(t) = P_i(t-1)$ .

Step 7: If  $i \in S_1$  and the fitness value of  $P_i(t)$  is better than the fitness value of the global best position for the whole swarm  $P_g(t-1)$ , i.e.  $f(P_i(t)) < f(P_g(t-1))$ , then  $P_i(t)$  is saved as the global best position of master group  $S_1$ , which is denoted as

$P_{gs1}(t)$ ; otherwise,  $P_{gs1}(t) = P_g(t-1)$ . If  $i \in S_2$  and the fitness value of  $P_i(t)$  is better than the fitness value of  $P_g(t-1)$ , i.e.  $f(P_i(t)) < f(P_g(t-1))$ , then  $P_i(t)$  is saved as the global best position of the slave group  $S_2$ , which is denoted as  $P_{gs2}(t)$ ; otherwise,  $P_{gs2}(t) = P_g(t-1)$ .

Step 8: Compare the fitness value of  $P_{gs1}(t)$  and  $P_{gs2}(t)$ , if  $f(P_{gs2}(t)) < f(P_{gs1}(t))$  then assign  $P_{gs2}(t)$  to be the global best position  $P_g(t)$ , i.e.  $P_g(t) = P_{gs2}(t)$ ; Otherwise,  $P_{gs1}(t)$  is assigned to  $P_g(t)$ , i.e.  $P_g(t) = P_{gs1}(t)$ .

Step 9: Termination determination. If the maximum number of iterations  $iterMax$  has been executed or the error precision of fitness value is achieved, then stop the searching process and output the results. Otherwise, let  $t = t + 1$  and repeat steps 3–9.

The execution flow chart of the DCC-QPSO algorithm is as shown in Figure 1.

## Experimental design and results analysis

### Experimental design

The performance and efficiency of the intelligent algorithms tend to be affected by the experimental parameter settings [16]. How to determine the parameters that achieve optimal performance is in itself a very complex optimization problem. In order to obtain reasonable experimental results, a set of benchmark functions, including the Sphere function, Rosenbrock function, Rastrigin function, Griewank function, Ackley function and Schaffer function, were adopted to test QPSO, BC-QPSO, DIR-QPSO and DCC-QPSO in a performance comparison.

### Benchmark functions

Benchmark functions with various characteristics are a major tool for performance evaluation in evolutionary algorithms. Unimodal and multimodal problems are commonly seen in engineering projects, thus, unimodal and multimodal functions are used as testing functions in this paper. Expressions, search range of variables, initialization range, optimal solution and optimal values are given in Tables 1 and 2.

### Experimental parameter configuration

The parameter  $\alpha$  decreases linearly from 1 to 0.5 in every algorithm to be tested, and the correlation parameter  $\rho$  is set to be  $-1$  in the BC-QPSO algorithm. The dimensionality of the benchmark functions is set to be 20 and the maximum number of iterations is set to be 1000. The size of the swarm population is set as 50, the size of the master group is 25 and the slave group is 25 as well; every benchmark function is tested times independently,

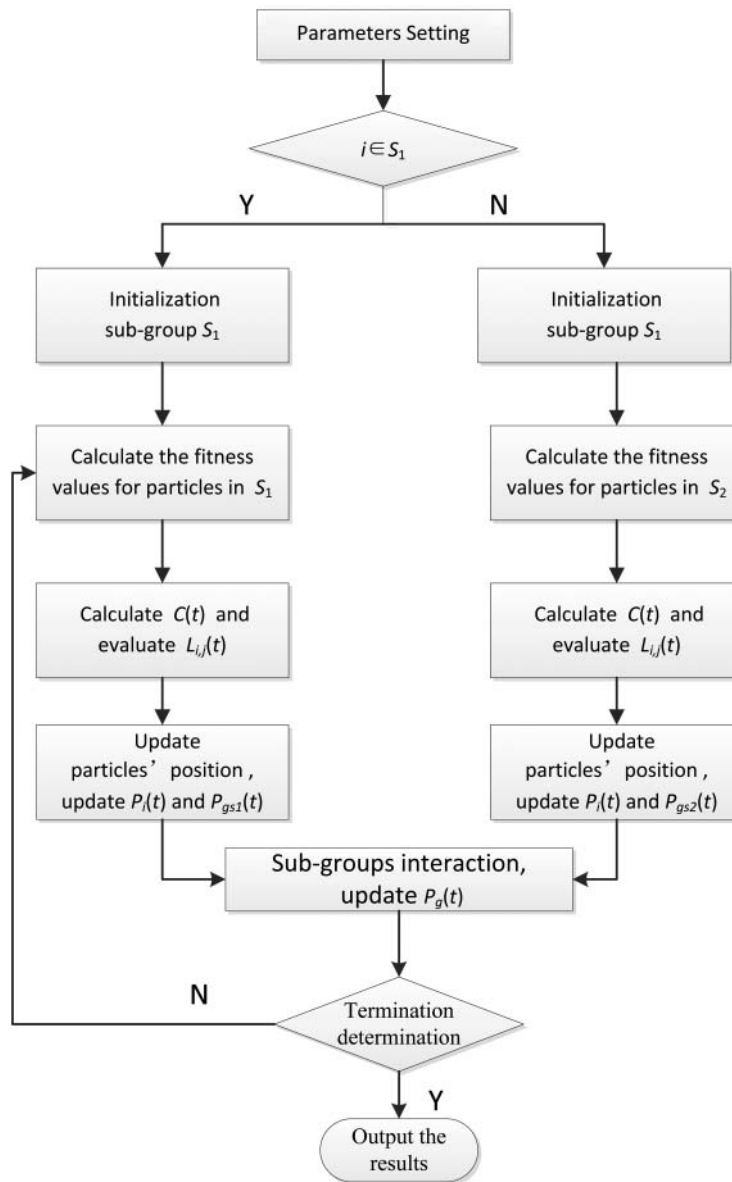


Figure 1. The execution flow chart of the DCC-QPSO algorithm.

Table 1. Unimodal functions.

Benchmark function	Equation	Searching range	Initial range	Optimal and fitness value
Sphere	$f_1(X) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	$[-100, 50]^D$	$f_1(0, 0, \dots, 0) = 0$
Rosenbrock	$f_2(X) = [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	$[-10, 10]^D$	$f_2(1, 1, \dots, 1) = 0$

and the average value of each test function is evaluated at the end of 30 iterations.

### Experimental results

The mean fitness value and the central processing unit (CPU) time of each iteration using QPSO, BC-QPSO, DIR-QPSO [23], DCC-QPSO solving benchmark functions is as shown in Table 3.

We see from Table 3. that compared with QPSO, DIR-QPSO and BC-QPSO, DCC-QPSO finds the optimal value for the Rastrigin function through

interaction between sub-groups. In addition, it leads to the best optimization for the Sphere function, multimodal Ackley function and Expanded Shaffer function. DIR-QPSO finds the optimal value for the Rastrigin function. The standard QPSO offers the best optimization precision while processing the Rosenbrock function and non-linear multimodal Griewank functions, whereas it provides the optimal value while processing the Ackley function.

Figure 2 depicts the convergence curves of the unimodal and multimodal test functions using QPSO, DIR-QPSO, DCC-QPSO and BC-QPSO

**Table 2.** Multimodal function.

Benchmark function	Equation	Searching range	Initial range	Optimal and fitness value
Rastrigin	$f_3(X) = \sum_{i=1}^D (x_i^2 - 10\cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	$[-5.12, 2]^D$	$f_3(0, 0, \dots, 0) = 0$
Griewank	$f_4(X) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^D$	$[-600, 200]^D$	$f_4(0, 0, \dots, 0) = 0$
Ackley	$f_5(X) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	$[-32.786, 32.786]^D$	$[-32.786, 16]^D$	$f_5(0, 0, \dots, 0) = 0$
Expanded shaffer	$f_6(X) = g(x_1, x_2) + g(x_2, x_3) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1)$ $g(x, y) = 0.5 + \frac{(\sin\sqrt{x^2 + y^2})^2 - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$	$[-100, 100]^D$	$[-100, 50]^D$	$f_6(0, 0, \dots, 0) = 0$

**Table 3.** The results of optimization algorithms ( $D = 20, N = 50$ ).

Algorithm		$f_1$	$f_2$	$f_3$	$f_4$	$f_5$	$f_6$
QPSO	FV	2.0941E-289	1.4351E001	9.6164	0.3631	2.0428E-014	2.8404E-007
	T(s)	1.2823	0.3123	0.2946	0.3410	0.3542	0.2607
BC-QPSO	FV	2.34E-231	1.5205 E001	16.9143	0.756309	2.94E-005	9.77E-009
	T(s)	1.3921	0.3122	0.2876	0.3479	0.3333	0.5797
DIR-QPSO	FV	2.02E-278	1.49123 E001	0	0.672136	1.02E-013	3.49E-009
	T(s)	1.2906	0.2062	0.2062	0.2625	0.2641	0.2219
DCC-QPSO	FV	1.53E-289	1.7494 E001	0	0.858269	2.66E-015	1.12E-009
	T(s)	1.2686	0.1987	0.2133	0.2532	0.3111	0.1984

algorithms when the population size is 50 and the problem dimensionality is 20. The comparison shows that the  $\delta$  potential well based DCC-QPSO algorithm not only attains a better optimization precision, but also provides faster convergence when processing unimodal functions, including Sphere function and Rosenbrock function, which are illustrated in Figure 2(a,b), respectively.

The convergence curve of Rastrigin is showed in Figure 2(c). This function is a non-linear multimodal

function with many local optimal solutions. Therefore, it is hard to find the global optimum and easy to become trapped in a local optimum. In the solving of the 20 dimensional Rastrigin function, the two single-population QPSO algorithms are trapped in the local optima, whereas the two interactive dual-group QPSO algorithms exhibit better performance in solving the problem due to the separation of potential well centres and the interaction between the master group and the slave group.

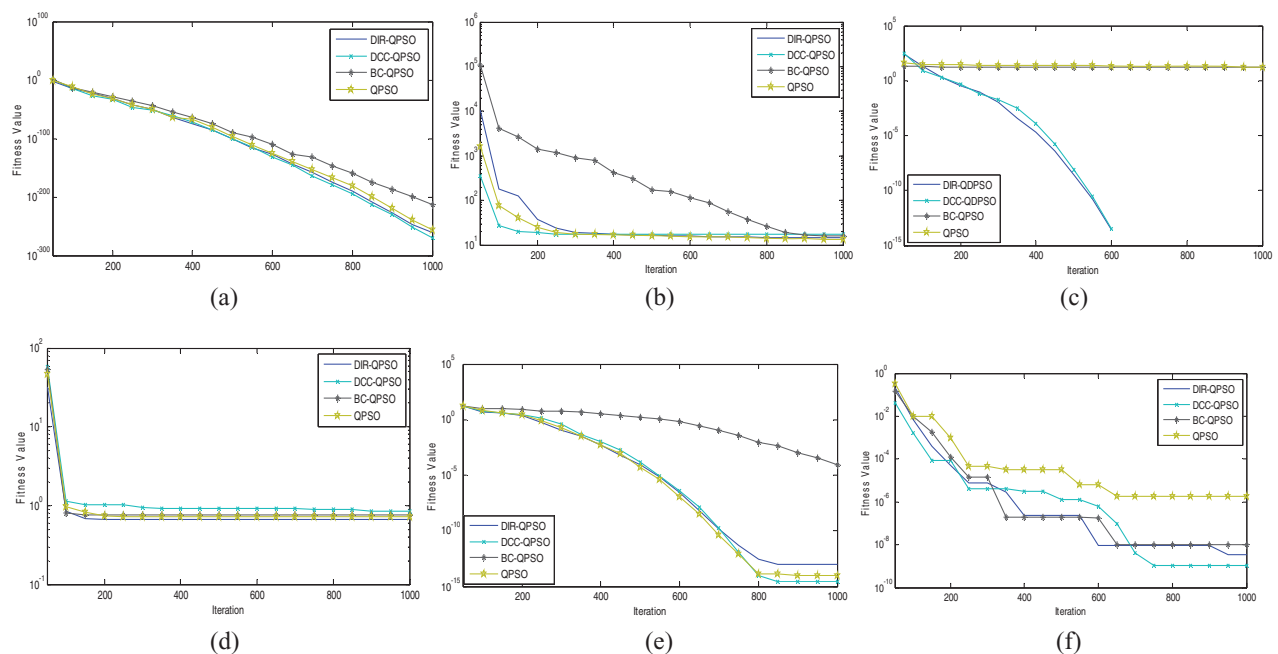
**Figure 2.** Convergence curves of test functions ( $D = 20, N = 50$ ). (a)  $f_1$ , (b)  $f_2$ , (c)  $f_3$ , (d)  $f_4$ , (e)  $f_5$ , (f)  $f_6$ .



Figure 2(d) shows the convergence curve of the Griewank function. The Griewank function is a strongly nonlinear multimodal function. This function has some features such as the strong interference among the product terms of the function, local optima distribute in a predictable manner, and the number of the local optima reduces as the dimensionality of the problem increases. The experimental results show that DIR-QPSO, DCC-QPSO and BC-QPSO exhibit similar performance when solving the 20-dimensional Griewank function, while DIR-QPSO slightly outperforms the others.

The convergence curve of the Ackley function is shown in Figure 2(e). The optimal solution of Ackley lies deep down in a “valley”, surrounded by regularly distributed local optima. The DCC-QPSO algorithm proposed by this paper gives the highest convergence precision, whereas, the QPSO shows its advantage in convergence speed when dealing with the Ackley function.

Figure 2(f) illustrates the convergence curve of the Expanded Shaffer function. The optimal value of this function is surrounded by a number of concentrically distributed local minima. In addition, the strong fluctuation of the function surface makes it even harder to find the real optimal solution. The four algorithms mentioned in this section consistently fall into the local optima in the process of iteration. The standard QPSO converges after 650 iterations and the DCC-QPSO does so after 750 iterations.

In summary, the single-population QPSO algorithms (the standard QPSO and the BC-QPSO) are capable of solving the simpler unimodal functions with one optimal solution in an efficient manner because they maintain the diversity of the population while elevating the average best position  $C$  to its optimal level. Meanwhile, unnecessary complexity is added to the optimization process when multi-population interactive algorithms are adopted, offering no advantages in solving unimodal functions. The dual-group QPSO algorithms exhibit better convergence performance when dealing with different multimodal functions. Compared with the other QPSO algorithms, the DCC-QPSO constructs two potential wells for a master and a slave group through each using a strategy based on different correlation coefficients. This policy offers the novel algorithm outstanding global searching ability due to the sharing of best positions between  $S1$  and  $S2$  through mutual complementation and evolution. The search pattern that two sub-groups carry out involving simultaneous searching and mutual learning enhances the probability of finding the optimal solution. It also offers higher convergence precision and speed within limited iterations.

Although superior performance can be gained from the multi-group interactive algorithms when solving

multimodal functions, the size of the population needs to be expanded. For single-population QPSO algorithms, the size of the population can be chosen between 20 and 60 while for multi-population interactive algorithms, the size of the population should not be too small, otherwise optimization may be compromised [24]. For instance, in this experiment, the size of population in the DIR-QPSO and that in DCC-QPSO are both set to be 50, divided equally into 25 for the main and slave groups, respectively. The energy-consuming problem brought by multi-groups can be solved by parallel computing techniques, thus the multi-group algorithm is a rational and effective route towards performance enhancement.

## Conclusion

This paper proposed the DCC-QPSO, a new Quantum-behaved PSO algorithm. This new algorithm divides the entire population into two sub-populations equal in size and applies different processing methods to these sub-groups in order to achieve the separation of potential well centres. The collaboration between main and slave sub-groups makes possible a more thorough mutual learning between particles, which leads to an enhancement in global searching ability, avoidance of falling into local optima prematurely and improvement of QPSO convergence performance. By the comparison of the solution processes of unimodal and multimodal functions, the DCC-QPSO proves to be a promising global optimization algorithm. This algorithm exhibits higher convergence precision and speed when dealing with multimodal functions, compared with traditional QPSO algorithms, and it also reflects its broad application potential in related areas where high solution precision is required.

## Disclosure statement

No potential conflict of interest was reported by the authors.

## Funding

This research is jointly supported by the Science and Technology Support Program of Sichuan Province [grant number 2016RZ0065], [grant number 2016RZ0053]; the Projects of the Education Department of Sichuan Province [grant numbers 15ZA0396 and 16ZB0212].

## References

- [1] Wang L, Liu B. Particle swarm optimization and scheduling algorithms. Beijing: Tsinghua University Press; 2008.
- [2] Kennedy J, Eberhart R. Particle swarm optimization. Proceedings of the IEEE International Conference on Neural Networks; Perth, Australia; 1995.

- [3] Wu T, Yan Y, Chen X. Adaptive partly informed PSO algorithm and its characteristics analysis. *Comput Eng Appl.* 2013;06:1–7+28.
- [4] Masdari M, Salehi F, Jalali M, et al. A survey of PSO-based scheduling algorithms in cloud computing. *J Netw Syst Manage.* 2017;25(1):122–158.
- [5] Abadi DNM, Khooban MH. Design of optimal Mamdani-type fuzzy controller for nonholonomic wheeled mobile robots. *J King Saud Univ Eng Sci.* 2015;27(1):92–100.
- [6] Khooban MH, Abadi DNM, Alfi A. Optimal type-2 fuzzy controller for HVAC systems. *Automatika.* 2014;55(1):69–78.
- [7] Niknam T, Khooban MH, Kavousifard A, et al. An optimal type II fuzzy sliding mode control design for a class of nonlinear systems. *Nonlinear Dyn.* 2014;75(1–2):73–83.
- [8] Khalghani MR, Shamsi-Nejad MA, Khooban MH. Dynamic voltage restorer control using bi-objective optimization to improve power quality's indices. *IET Sci Meas Technol.* 2014;8(4):203–213.
- [9] Bergh FVD. An analysis of particle swarm optimizers. South Africa: University of Pretoria; 2002. p. 1.
- [10] Gao Y, Duan Y. An adaptive particle swarm optimization algorithm with new random inertia weight, advanced intelligent computing theories and applications. Berlin: Springer; 2007. p. 342–350.
- [11] Zhang L, Tang Y, Hua C, et al. A new particle swarm optimization algorithm with adaptive inertia weight based on Bayesian techniques. *Appl Soft Comput.* 2015;28:138–149.
- [12] Zhan Z, Zhang J, Li Y, et al. Adaptive particle swarm optimization. *IEEE Trans Syst Man Cybern B Cybern.* 2009;39(6):1362–1381.
- [13] Sun J. Quantum-behaved particle swarm optimization algorithm. Jiangsu: Jiangsu University; 2009.
- [14] Mikki SM, Kishk AA. Investigation of the quantum particle swarm optimization technique for electromagnetic applications. *IEEE Antennas Propag Soc Int Symp.* 2005;2:45.
- [15] Mikki SM, Kishk AA. Quantum particle swarm optimization for electromagnetics. *IEEE Trans Antennas Propag.* 2006;54(10):2764–2775.
- [16] Lin X, Feng B, Sun J. Chaos quantum-behaved particle swarm optimization algorithm. *Comput Eng Design.* 2008;29(10):2610–2612.
- [17] Yang CH, Du W. Optimal pricing with group-buying auction based on modified QPSO. *Comput Eng Appl.* 2008;44(8):210–211.
- [18] Ge HW, Jin WH. Application of quantum-behaved PSO algorithm with mutation operator in system parameters identification. *Comput Eng Appl.* 2007;43(29):222–224.
- [19] Li P, Wang H, Song K, et al. Research on the improvement of quantum potential well-based particle swarm optimization algorithm. *Acta Phys Sin.* 2012;61(6): 19–28.
- [20] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans Evol Comput.* 2002;6(1):58–73.
- [21] Wu T, Chen X, Yan Y. Study of the binary correlation quantum-behaved PSO algorithm. *J Sichuan Univ (Eng Sci Ed).* 2014;46(4):103–110.
- [22] Sun J, Fang W, Wu X. Quantum-behaved particle swarm optimization: theory and application. Beijing: Tsinghua University Press; 2011.
- [23] Wang L. Intelligent optimization algorithm and its application. Beijing: Tsinghua University Press; 2001.
- [24] Wu T, Yan Y, Chen X. Improved dual-group interaction QPSO algorithm based on random evaluation. *Control Decis.* 2015;3:526–530.