Computer Science and Information Systems 18(1):189–212 https://doi.org/10.2298/CSIS200223033Y

A QPSO Algorithm Based on Hierarchical Weight and Its Application in Cloud Computing Task Scheduling

Guolong Yu[1], Yong Zhao[2], Zhongwei Cui[1], and Yu Zuo[1]

[1] School of Mathematics and Big Data, Guizhou Education University,
Guiyang, 550000, China
heihuzhiguang@163.com
[2] School of Information Engineering, Shenzhen Graduate School of Peking
University, Shenzhen, 518000, China
516636425@qq.com

Abstract. The computing method of the average optimal position is one of the most important factors that affect the optimization performance of the QPSO algorithm. Therefore, a particle position weight computing method based on particle fitness value grading is proposed, which is called HWQPSO (hierarchical weight QPSO). In this method, the higher the fitness value of a particle, the higher the level of the particle, and the greater the weight. Particles at different levels have different weights, while particles at the same level have the same weight. Through this method, the excellent particles have higher average optimal position weight, and at the same time, the absolute weight of a few particles is avoided, so that the algorithm can quickly and stably converge to the optimal solution, and improve the optimization ability and efficiency of the algorithm. In order to verify the effectiveness of the method, five standard test functions are selected to test the performance of HWQPSO, QPSO, DWC-QPSO and LTQPSO algorithm, and the algorithms are applied to the task scheduling of the cloud computing platform. Through the test experiment and application comparison, the results show that the HWQPSO algorithm can converge to the optimal solution of the test function faster than the other three algorithms. It can also find the task scheduling scheme with the shortest time consumption and the most balanced computing resource load in the cloud platform. In the experiment, compared with QPSO, DWC-QPSO and LTQPSO algorithm, HWQPSO execution time of the maximum task scheduling was reduced by 35%, 23% and 21% respectively.

Keywords: QPSO algorithm, hierarchical weight, cloud computing, task scheduling, average optimal location.

1. Introduction

Particle swarm optimization (PSO) is one of the most widely used swarm intelligence algorithms. The algorithm is relatively easy to implement, needs to determine fewer parameters and has the advantages of efficient parallel search, which can effectively solve complex optimization problems. Its performance is a hot research issue in the field of Intelligent Computing in recent years, and it has been widely used in resource scheduling, pattern recognition, complex optimization and other issues. However, the PSO algorithm is easy to fall into the local optimal solution when searching, later the particle convergence efficiency is lower, and it can not converge to the optimal solution with probability

1[1-3]. To improve the global search ability of particles, SUN et al. based on the aggregation of particle swarm, established the delta potential well model in quantum state, and then set the control parameters according to the coordination and self-organization of particle swarm, proposed the quantum behaved particle swarm optimization algorithm, i.e. QPSO algorithm[4,5]. In the QPSO algorithm, particles in the quantum space can appear at any point in the search space with a certain probability, and the motion state of the particle is represented by wave function instead of Newton space motion of the particle, and the probability density function of wave function is used to determine the position of the particle in solution space. This position is random, as long as the particle iterations continuously, it will pass through any position in solution space with a certain probability [6-8]. In this way, particles update their positions according to the quantum behavior, and gradually iterate to the global optimal solution. Compared with PSO algorithm, QPSO algorithm increases the randomness of particles, makes particle updating equation simple, has few control parameters and fast convergence speed.

Although the QPSO algorithm has more advantages than the PSO algorithm, there are still many shortcomings in the QPSO algorithm. Many researchers have made a lot of optimization improvements in its contraction and expansion coefficient, population diversity, convergence efficiency, decision-making strategy of average optimal location, etc. For example, Zhen-Lun Y proposed an improved quantum-behaved particle swarm optimization with elitist breeding (EB-QPSO) for unconstrained optimization in reference [9]. During the iterative optimization process of EB-QPSO, when criteria met, the personal best of each particle and the global best of the swarm are used to generate new diverse individuals through the transposon operators. The new generated individuals with better fitness are selected to be the newpersonal best particles and global best particle to guide the swarm for further solution exploration. In addition to the above optimization of individual and global optimal particles, the dual-group search is also an important QPSO optimization method. Such as a dual-group QPSO with different well centers (DWC-QPSO) algorithm, is proposed by constructing the master-slave subswarms in reference [10]. This algorithm avoids the rapid disappearance of swarm diversity and enhances the global searching ability through collaboration between subswarms. Xue T considers the method of mixed optimization of QPSO under complex conditions, so he proposed a hybrid improved quantum behaved particle swarm optimization (LTQPSO) in reference [11]. The algorithm combines the individual particle evolutionary rate and the swarm dispersion with the natural selection method in the particle evolution process. The algorithm has good robustness and convergence. In order to improve the evolution of quantum individuals and the ability to converge to the optimal solution of the QPSO algorithm, Chen W proposed a mixed quantum algorithm based on local optimization strategy and improved optimization rotation angle in reference [12].

Not limited to the above introduction, many researchers have done a lot of work to optimize the convergence efficiency of the QPSO algorithm. In this paper, from the point of view that the weight of each particle's position should be different in the calculation of the average optimal position of different particles, it is considered that the excellent particles should have a larger decision weight, while the inferior particles should have a relatively smaller decision weight. A weight calculation method is proposed, which classifies the weights based on the fitness value of particles, to improve the global search ability and search efficiency of the QPSO algorithm. After the standard test function is tested in the HWQPSO algorithm and the original QPSO algorithm, the DWC-QPSO algorithm in reference [10] and the LTQPSO algorithm in reference [11]. The optimized QPSO algorithm in this paper has more advantages than other algorithms not only in local accuracy and global search ability, but also in convergence speed and stability.

With the wide application of cloud computing technology, due to the large amount of data calculation, the computing efficiency of the platform is paid more and more attention. In addition to improving the hardware performance of the platform, the computing efficiency of the software system also greatly restricts the overall performance of the cloud computing platform. One of the most concerned methods is how to achieve efficient resource scheduling of the cloud computing platform. As one of the most excellent swarm intelligence algorithms, QPSO algorithm has strong optimization ability. It has obvious advantages to apply QPSO algorithm to resource scheduling strategy optimization of cloud computing platform. In this paper, the optimization performance of QPSO algorithm is optimized. By grading the weight coefficients in the average optimal position calculation of particles, the fairness of the average optimal position calculation is improved, and the average optimal position can guide particles to converge to the optimal solution more accurately and quickly. Finally, the optimized QPSO algorithm is applied to the cloud computing task scheduling, and HWQPSO algorithm is used to allocate the tasks of cloud computing to different cloud computing resources reasonably, so as to the overall computing efficiency of the task set is more efficient. The simulation experiment on the CloudSim cloud platform shows that the algorithm in this paper can provide efficient task scheduling strategy for the cloud computing platform, make the resource load of the cloud computing platform more balanced, and improve the computing efficiency of the cloud platform.

2. QPSO Algorithm Model

QPSO algorithm is a kind of PSO algorithm with quantum behavior. Unlike PSO algorithm, the particle in QPSO algorithm is in quantum space, and the particle appears at any point in space according to probability. It is assumed that there are *N* particles representing the solution in the solution space, The position of the *i*-th particle in the *D* dimensional search space is $X_i = (x_{i1}, x_{i2}, \ldots, x_{iD})$. The local optimal position of particle *i* is $pb_i = (pb_{i1}, pb_{i2}, \ldots, pb_{iD})$. The global optimal position of the whole particle swarm is $gb_i = (gb_{i1}, gb_{i2}, \ldots, gb_{iD})$. Using wave function ψ to determine the state of particles in quantum space, the probability of a particle appearing at a certain position in space can be expressed by $|\psi|^2$. If the potential well in *D* dimension is $pb_{id}(t)$ in the *t*-th iteration of particle *i* [13-15].

The wave function ψ(x, t) is used to describe the particle's position and search speed in space, X=(x, y, z), which is a vector, is the position of particles in three-dimensional space, then $|\psi|^2$ is the probability density of particles appearing in three-dimensional space (x, y, z) at time t, as shown in formula (1).

$$|\psi|^2 d_x d_y d_z = Q d_x d_y d_z \qquad (1)$$

In the formula, Q is the probability density function. Q should meet the normalization requirements, such as formula (2).

$$\int_{-\infty}^{+\infty} |\psi|^2 d_x d_y d_z = \int_{-\infty}^{+\infty} Q d_x d_y d_z = 1 \qquad (2)$$

In QPSO algorithm, the state change of each particle in the system follows the Schrodinger equation. At the same time, the δ potential well is introduced into the system, and the potential well is established at p_{id} point. The potential energy function is as formula (3). The steady state Schrodinger equation of the particle in the potential well can be obtained, such as formula (4).

$$V(x) = -\gamma\delta(X - p_{id}) \qquad (3)$$

$$\frac{d^2\psi}{d(X - p_{id})^2} + \frac{2m}{h^2}[E + \gamma\delta(X - p_{id})]\psi = 0 \qquad (4)$$

E is the energy of the particle, h is the Planck constant, and m is the mass of the particle.

The wave function can be obtained by solving the Schrodinger equation, such as formula (5).

$$\psi(X - p_{id}) = \frac{1}{\sqrt{L}} e^{-|X - p_{id}|/L}, L = 1/\beta = h^2 m\gamma \qquad (5)$$

Monte Carlo method is used to sample the particle position randomly, and the position component of the *i*-th particle in the *d* dimension is obtained in the *(t + 1)*-th iteration, as shown in formula(6).

$$x_{id}(t+1) = pb_{id}(t) \pm \frac{L_{id}(t)}{2} ln[\frac{1}{u_{id}(t)}] \qquad (6)$$

In the formula(6), $u_{id}(t) \sim U(0,1)$. The characteristic length of potential well $L_{id}(t)$ is calculated by formula(7).

$$L_{id}(t) = 2\alpha(t)|mb_d(t) - x_{id}(t)| \qquad (7)$$

The *mb* is called the average optimal position, it is the center of the optimal position of all particles. In *D* dimensional space, mb(t) can be calculated by formula(8).

$$mb(t) = (mb_1(t), mb_2(t), ..., mb_D(t)) = \frac{1}{N}\sum_{i=1}^{N} pb_i(t)$$

$$= (\frac{1}{N}\sum_{i=1}^{N} pb_{i1}(t), \frac{1}{N}\sum_{i=1}^{N} pb_{i2}(t), ..., \frac{1}{N}\sum_{i=1}^{N} pb_{iD}(t)) \qquad (8)$$

In the formula(7), α is the contraction expansion coefficient, whose value will directly affect the convergence performance of the algorithm. The value of $\alpha(t)$ in this paper is shown in formula(9).

$$\alpha(t) = 0.5 + \frac{(1-0.5)(t_{max}-t)}{t_{max}} \qquad (9)$$

In the formula(9), *t* is the current number of iterations and t_{max} is the maximum number of iterations. The necessary and sufficient condition for QPSO algorithm to converge to the center of potential well is that the coefficient α <1.78 [16].

The updated formulas of particle's current optimal position pb_i and global optimal position *gb* are shown in formula (10) and formula (11) respectively.

$$pb_i(t+1) = \begin{cases} x_i(t+1) & f[x_i(t+1)] < f[pb_i(t)] \\ pb_i(t) & f[x_i(t+1)] \geq f[pb_i(t)] \end{cases} \qquad (10)$$

$$gb_g(t+1) = argmin\{f[pb_i(t)]\} \qquad (11)$$

In the formula, *f* is the objective function.

In the QPSO algorithm, the particle has only displacement value but no velocity vector in quantum space. The determination of particle position is mainly to obtain the wave function by solving the Schrodinger equation, such as formula (6), to calculate the probability density function and the probability of particle appearing at a certain point in quantum space, and then use Monte Carlo method to randomly sample the particle position to obtain the particle position component, such as formula(6). In the potential well characteristic length $L_{id}(t)$, as shown in formula(7), the average best position *mb* of particles is introduced, as shown in formula (8), to measure the creativity of particles. To improve the ability of interaction between particle swarm and enhance the global search ability of the algorithm. Therefore, the average optimal position *mb*, which is the center of the optimal position of all particles, is one of the core parameters of the whole algorithm.

3. Average Optimal Position of Particles Based on Hierarchical Weight

The biggest difference between the QPSO algorithm and the PSO algorithm is that the particle position update method is different. When updating the particle position, it not only considers the local and global optimal position of the current particle, but also introduces the average optimal position *mb*, which increases the interaction between particles and strengthens the global search ability of particle swarm.

The average optimal position *mb* of the original QPSO algorithm is shown in formula(8). It can be seen that it is the average of the local optimal value of each particle position, which determines the update of particle position. In the calculation process of *mb*, the weight of the local optimal value $pb_i(t)$ of each particle is the same, as shown in formula(12), the proportion of each particle's position in the calculation of *mb* is 1, that is, each particle has the same influence on the final average optimal position *mb* decision. This is not in line with the group intelligent decision-making strategy. In reality, the decision weight of excellent particles is higher than that of inferior particles.

$$mb(t) = (mb_1(t), mb_2(t), ..., mb_D(t)) = \frac{1}{N}\sum_{i=1}^{N} pb_i(t) = \frac{1}{N}\sum_{i=1}^{N}[1 \times pb_i(t)] \qquad (12)$$

Aiming at the problem of unbalanced influence of particles in the calculation of the average optimal position in QPSO algorithm, this paper also introduces a weight factor δ, $\delta_i(t)$ represents the weight of the local optimal value $pb_i(t)$ of the *i*-th particle in the calculation of the average optimal position *mb(t)* of the particle in the *t*-th iteration. After introducing the weight factor δ, the calculation formula(8) of the average optimal position *mb(t)* of the particle can be expressed as the formula(13) [17-20].

$$mb(t) = \frac{1}{N}\sum_{i=1}^{N}[\delta_i(t) \times pb_i(t)] \qquad (13)$$

First, the fitness value $f_i(1 \leq i \leq N, i \in Z)$ of particles is sorted from large to small, and the fitness value after sorting is $f'_j(1 \leq j \leq N, j \in Z)$, $f'_1 \geq f'_2 \geq f'_3 \geq ... \geq f'_N$. According to the fitness value *f*, the particles are divided into $r(1 \leq r \leq N, r \in Z)$ levels, $F_1, F_2, F_3...F_r$. Particles with the same level have the same weight value δ, and the weight values of particles with different levels are $\delta_1, \delta_2, \delta_3, ..., \delta_r, \delta_1 \geq \delta_2 \geq \delta_3 \geq ... \geq \delta_r$. Let δ_r obey the uniform distribution of some subinterval on $[\theta_1, \theta_2]$, and assume that $\theta_1 \leq a_r \leq b_r \leq ... \leq a_3 \leq b_3 \leq a_2 \leq b_2 \leq a_1 \leq b_1 \leq \theta_2$, then $\delta_1 \sim U_1(a_1, b_1), \delta_2 \sim U_2(a_2, b_2), \delta_3 \sim U_3(a_3, b_3), ..., \delta_r \sim U_r(a_r, b_r)$. Then the weight value $\delta_i(t)$ in formula(9) can be calculated by formula(14).

$$\delta_i(t) = \begin{cases} \delta_1(t) \sim U_1(a_1, b_1), & f'_1 \geq f[pb_i(t)] \geq f'_{\lceil (b_1 - a_1)\cdot N \rceil} \\ \delta_2(t) \sim U_2(a_2, b_2), & f'_{\lfloor (b_1 - b_2)\cdot N \rfloor} \geq f[pb_i(t)] \geq f'_{\lceil (b_1 - a_2)\cdot N \rceil} \\ \delta_3(t) \sim U_3(a_3, b_3), & f'_{\lfloor (b_1 - b_3)\cdot N \rfloor} \geq f[pb_i(t)] \geq f'_{\lceil (b_1 - a_3)\cdot N \rceil} \\ \vdots & \\ \delta_r(t) \sim U_r(a_r, b_r), & f'_{\lfloor (b_1 - b_r)\cdot N \rfloor} \geq f[pb_i(t)] \geq f'_N \end{cases} \qquad (14)$$

Through the calculation method of formula(10), the particles with the same fitness level will have the uniform distribution value that obeys the corresponding level interval of $[\theta_1, \theta_2]$. This can make the particles with higher fitness occupy a higher weight in the calculation of individual average optimal position *mb(t)*, otherwise, the smaller the weight is, so that the decision-making influence of particles with different fitness can be well balanced. It is beneficial to increase the interaction between particles and enhance the search ability of particle swarm. The comparison between the weight value of the improved algorithm and the original QPSO algorithm is shown in figure 1 and figure 2. In figure 1, the traditional particle swarm optimization method, no matter what the ability of particle optimization is, each particle gets the same weight 1. In figure 2, the improved method in this paper, the stronger the particle optimization ability is, the higher the weight is in the calculation of the average optimal position.

4. HWQPSO for Function Optimization

In order to evaluate the performance of HWQPSO algorithm, it has been applied to some well-known benchmark functions, these functions are used in both reference [10] and [11]. These standard test functions have been adopted in many literatures and are very

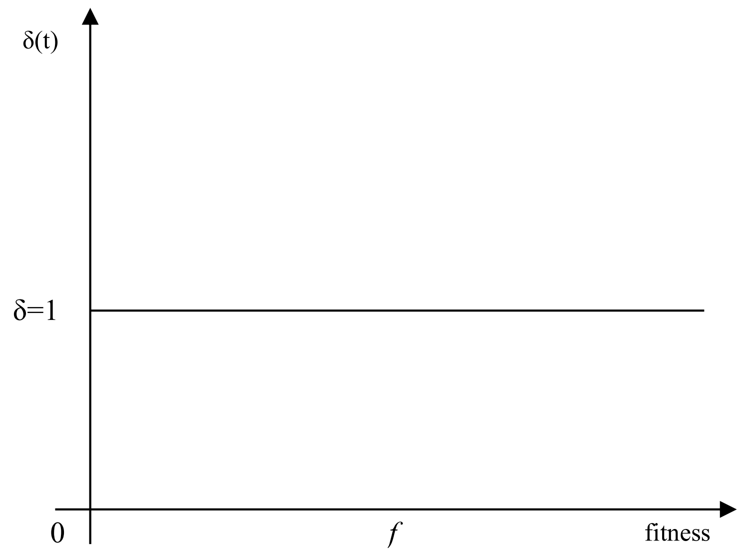


Fig. 1. Weight value graph of QPSO algorithm

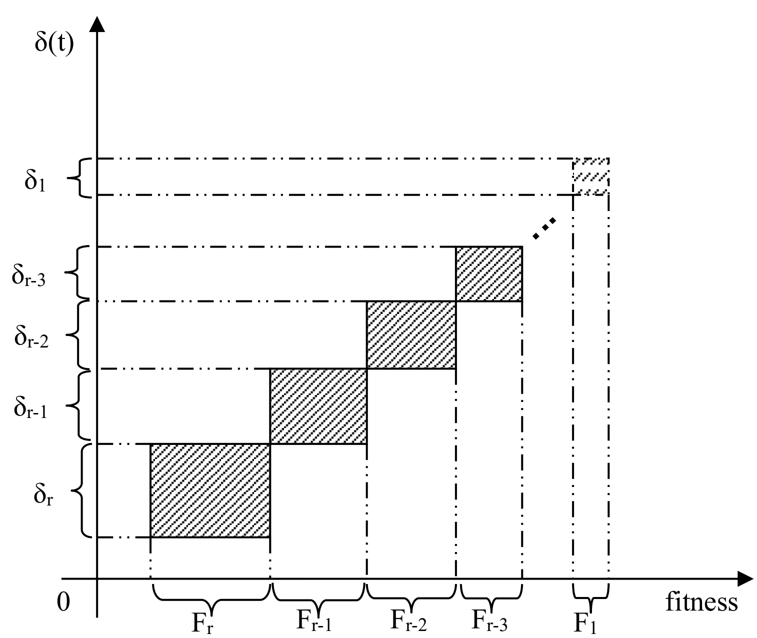


Fig. 2. Weight value graph of HWQPSO algorithm

representative. They can well evaluate the performance of the optimization algorithm. The details of the benchmark functions are given in Table 1, including function name, specific formula, range min value and search ability. These benchmark functions are minimization problems and the global best value for all these functions is zero. The experimental results of HWQPSO are compared with those of QPSO [4], DWC-QPSO [10] and LTQPSO [11]. The parameters of QPSO, DWC-QPSO and LTQPSO select original paper parameters. The weight parameter of HWQPSO algorithm is shown in Table 2. r=4, θ_1=0.5, θ_2=1.5. we compare the convergence rate of the four algorithms in the process of 8 standard function tests, in which the average optimal value of the objective function changes with the number of iterations. Figures 3 $\sim$ 10 show average of convergence curves for 50 runs of QPSO, DWC-QPSO, LTQPSO and HWQPSO under the condition of N = 40, D = 30, M = 2000, but the Schaffer function under the condition of N = 40, D = 2, M = 2000. In order to show the results more intuitively, when drawing the contrast curves, the log value of the fitness value is calculated on the vertical axis.

Table 1. Mathematical benchmark functions

Function	Formulation	Range	Min value	Search ability
Sphere	$f_1(x) = \sum_{i=1}^{D} x_i^2$	(-100,100)	0	Local
DeJong's	$f_2(x) = \sum_{i=1}^{D} i \cdot x_i^4$	(-100,100)	0	Local
Rosenbrock	$f_3(x) = \sum_{i=1}^{D-1}(100 \cdot (x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	(-5.12,5.12)	0	Global/ Local
Griewank	$f_4(x) = \sum_{i=1}^{D} x_i^2/4000 - \prod_{i=1}^{D} \cos(x_i/\sqrt{i}) + 1$	(-600,600)	0	Global
Rastrigin	$f_5(x) = \sum_{i=1}^{D}(x_i^2 - 10 \cdot \cos(2\pi x_i) + 10)$	(-5.12,5.12)	0	Global
Ackley	$f_6(x) = -20exp(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}) - exp(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)) + 20 + e$	(-32,32)	0	Global
Schaffer	$f_7(x) = 0.5 + ((\sin\sqrt{x_1^2 + x_2^2})^2 - 0.5)/((1 + 0.001(x_1^2 + x_2^2))^2)$	(-2.048,2.048)	0	Global
Schwefel	$f_8(x) = 418.9829D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	(-500,500)	0	Global

From the experimental results curves in figure 3 to figure 10, it can be seen that the HWQPSO algorithm has the best optimization ability and search stability compared with the other three algorithms, Among them, HWQPSO algorithm performs best in the Sphere and DeJong's function tests, and worst in Rosenbrock function tests. Among the eight

Table 2. Particle weight distribution table

The value of r	Distribution interval	Distribution ratio
r=1	$\delta_1 \sim U_1(1.4, 1.5)$	10%
r=2	$\delta_2 \sim U_2(1.2, 1.4)$	20%
r=3	$\delta_3 \sim U_3(0.9, 1.2)$	30%
r=4	$\delta_4 \sim U_4(0.5, 0.9)$	40%

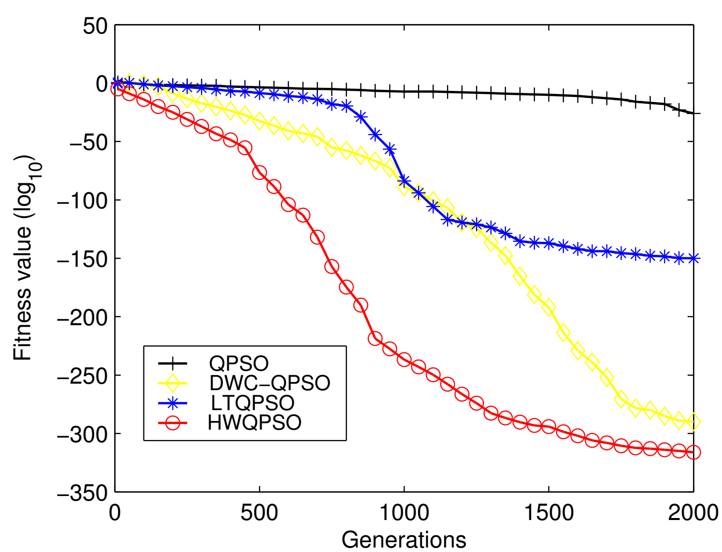


Fig. 3. Convergence curve of the Sphere function

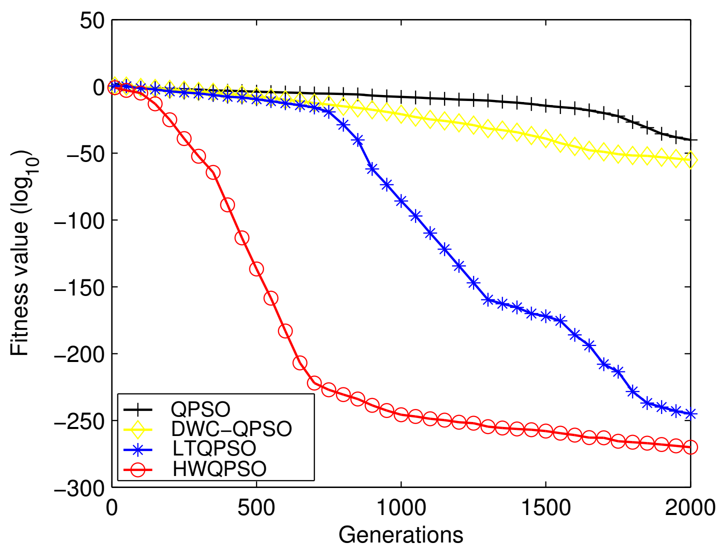


Fig. 4. Convergence curve of the DeJong's function

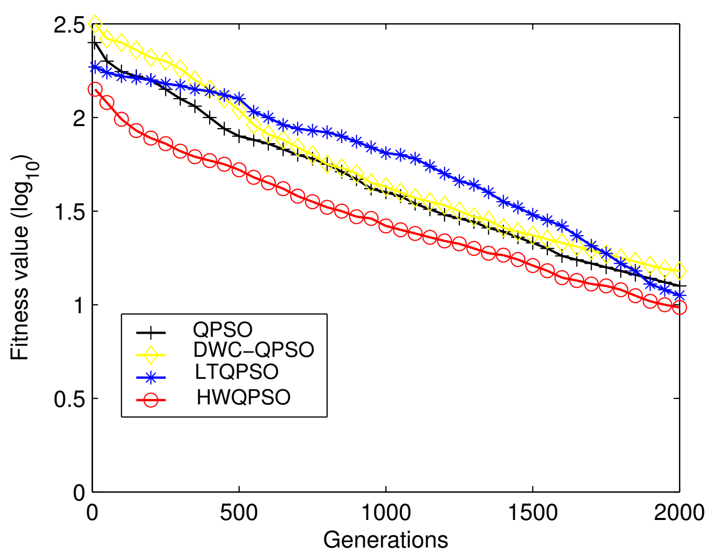


Fig. 5. Convergence curve of the Rosenbrock function

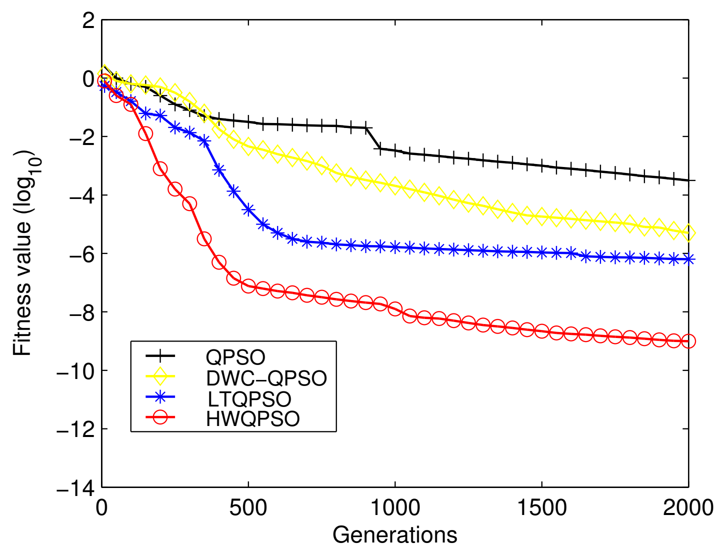


Fig. 6. Convergence curve of the Griewank function

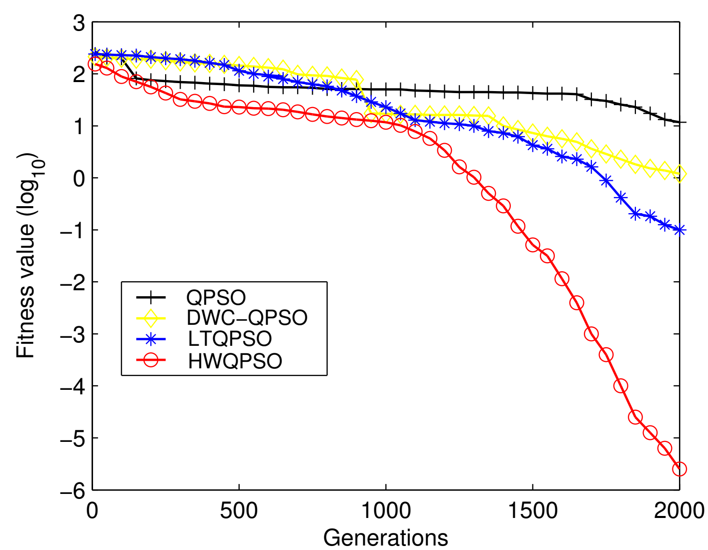


Fig. 7. Convergence curve of the Rastrigin function

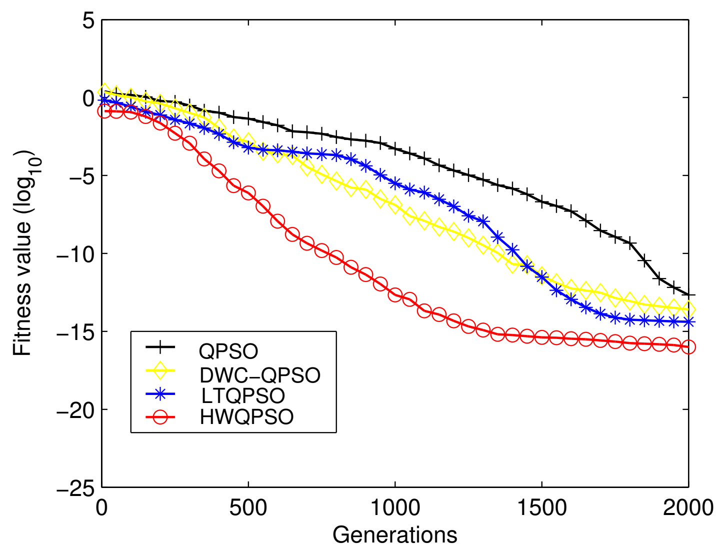


Fig. 8. Convergence curve of the Ackley function

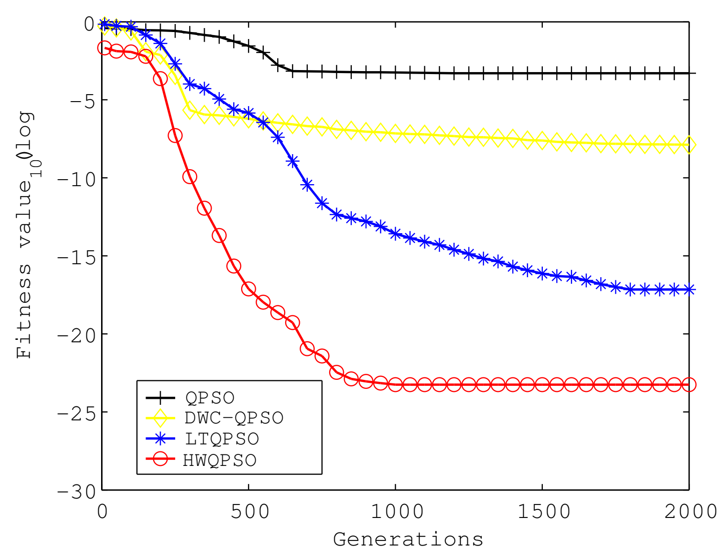


Fig. 9. Convergence curve of the Schaffer function

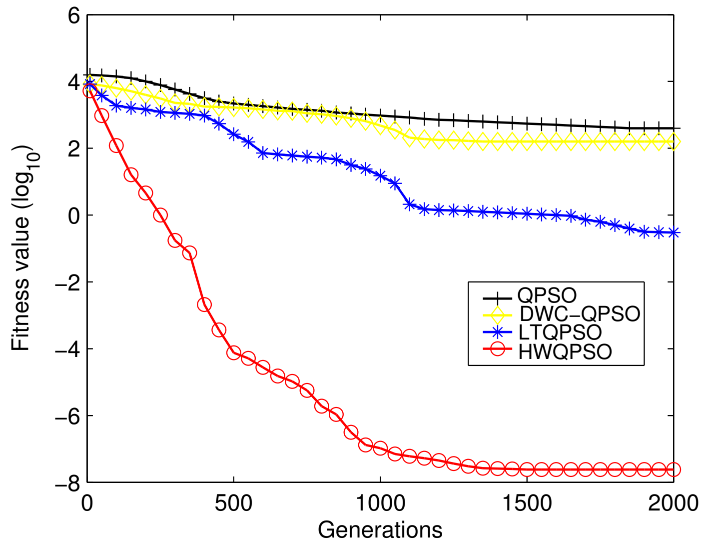


Fig. 10. Convergence curve of the Schwefel function

standard test functions, Sphere and DeJong's are unimodal functions, which are generally used to test the local search ability of the algorithm. From the comparison of the experimental results on these two functions, HWQPSO algorithm has better local search ability and search stability than the other three algorithms, but from the average optimal value data, DeJong's function has higher search accuracy and stability than Sphere function, and DWC-QPSO algorithm also shows the performance close to HWQPSO algorithm on Sphere function. Rosenbrock function is usually used to test the local and global search ability of optimization algorithm. Each contour line of Rosenbrock function is approximately parabola shaped, and its global minimum value is located in the parabola shaped Valley, which is easy to find, but because the value in the valley changes little, it is very difficult to find the global minimum value. So in eight test functions, the results of four algorithms on Rosenbrock function are the worst, but the results of hwqpso algorithm in this paper are still better than the other three algorithms. Griewank, Rastrigin, Ackley, Schaffer and Schwefel functions are nonlinear multi peak functions, which are usually used to test the global search ability of optimization algorithms. The experimental results show that the optimization effect of the four algorithms on these five functions is not as good as that of sphere and DeJong's unimodal functions, but better than Rosenbrock functions. The global optimization results of HWQPSO are better than those of other three algorithms. Among them, the search performance of Rastrigin and Schwefel functions is better than that of the other three functions.

On the other hand, for HWQPSO algorithm, it can be found that the fitness values of the HWQPSO algorithm are lower than those of the QPSO, DWC-QPSO and LTQPSO in 2000 iterations of 8 standard functions. That is to say, the red curve representing the iterative fitness values of the HWQPSO algorithm in 8 figures is always lower than the other three curves. At the same number of iterations, the local solution found by the HWQPSO algorithm is better than the other three algorithms. Especially in Sphere, De-Jong's, Griewank, Schaffer, and Schwefel functions, it shows that the local optimization accuracy of the HWQPSO algorithm is better, and each iteration can find a better solution than the other three algorithms.

At the same time, the lower red curve also means that the convergence speed of the HWQPSO algorithm is faster than other three algorithms. The optimal solution of the HWQPSO algorithm is closer to the optimal value, it will quickly approach the optimal solution. For example, when 500 iterations in figure 10, the fitness value of the HWQPSO algorithm is 10^{-4}, which converges to the optimal solution 0 faster than other three algorithms.

It can also be found from the experimental results that the HWQPSO algorithm has the strongest global optimization ability, which is also higher than the other three algorithms. That is to say, at the abscissa 2000 point in the experimental result graph, the HWQPSO algorithm has the least fitness value compared with the other three algorithms in 8 standard functions, that is, the solution is optimal.

All these are mainly due to the particles with higher fitness value of the HWQPSO algorithm, and the larger the calculated weight of the average optimal position, which makes the average optimal position tend to be excellent particles and find better solutions under the leadership of excellent particles.

To sum up, through the experiments of the QPSO, DWC-QPSO, LTQPSO and HW-QPSO algorithm on 8 standard test functions, the results show that the HWQPSO algorithm proposed in this paper has more advantages than the QPSO, DWC-QPSO and LTQPSO in local accuracy, convergence speed and global search ability.

5. HWQPSO for Task Scheduling in Cloud Computing

Assuming that there are *H* computing resources available in the cloud computing platform, the computing resources are the position *X* of particles in the space search, that is, the resource set $X = \{x_1, x_2, x_3, \ldots, x_H\}$; There are *D* computing task requests in the cloud computing system at a certain time, Task set $S = \{s_1, s_2, s_3, \ldots, s_D\}$; The matrix *T* is used to represent the time when different tasks calculate data on different resources, such as formula (15), that is t_{ij} represents the time required for the *i*-th task to complete data processing on the *j*-th calculation resource, $1 \leq i \leq D, 1 \leq j \leq H$. In the process of searching space, the position of the *i*-th particle is $X_i = \{x_{i1}, x_{i2}, \ldots, x_{iD}\}$, $1 \leq i \leq D$, and the value of dimension *D* is the number of tasks in the cloud computing platform. x_{id} means that task *d* is scheduled to execute on resource x_i, and its matrix representation is as shown in formula (16). *X* is a $0 - 1$ matrix, when $x_{ij} = 1$, it means that the data to be processed by the *i*-th task is processed by the *j*-th computing resource; when x_{ij}=0, it means that the data to be processed by the *i*-th task is not processed by the *j*-th computing resource[21-24].

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & ... & t_{1H} \\ t_{21} & t_{22} & t_{23} & ... & t_{2H} \\ t_{31} & t_{32} & t_{33} & ... & t_{3H} \\ ... & ... & ... & ... & ... \\ t_{D1} & t_{D2} & t_{D3} & ... & t_{DH} \end{bmatrix} \quad (15) \qquad X = \begin{bmatrix} x_{11} & x_{12} & x_{13} & ... & x_{1D} \\ x_{21} & x_{22} & x_{23} & ... & x_{2D} \\ x_{31} & x_{32} & x_{33} & ... & x_{3D} \\ ... & ... & ... & ... & ... \\ x_{N1} & x_{N2} & x_{N3} & ... & x_{ND} \end{bmatrix} \quad (16)$$

In the task scheduling method in this paper, it is expected that all computing tasks in the task set will be completed, and the less the total time T_{total} is, the better. From the above analysis, it can be concluded that the total time T_{total} taken by all calculation tasks is shown in formula (17).

$$T_{total} = max_{j=1}^{H} \sum_{i=1}^{D} x_{ij} \times t_{ij} \qquad (17)$$

Through the above optimization of QPSO algorithm, it can be seen that the higher the fitness of particles, the higher the proportion of particles should be when calculating the average optimal position mb of particles. Therefore, the fitness function f of particles is defined as shown in formula (18).

$$f = \frac{1}{T_{total}} = \frac{1}{max_{j=1}^{H} \sum_{i=1}^{D} x_{ij} \times t_{ij}} \qquad (18)$$

It can be seen from formula (14) that the longer the total execution time of the calculation task set is, the smaller the value of fitness f will be, and the lower the calculation efficiency of the cloud computing platform will be; on the contrary, the shorter the total execution time of the calculation task set is, the larger the value of fitness f will be, and the higher the processing efficiency of the cloud computing platform will be, which meets the processing efficiency expectation of the cloud computing platform.

6. Experiments and Analysis

In this paper, Cloudsim4.0, a cloud computing simulation tool, is used as the experimental platform, and its parameters are shown in Table 3. The main classes and methods of the simulation process and its implementation based on Java development environment are shown in Figure 11. According to the modeling requirements of this paper, the task model and virtual machine model are adjusted on the platform, and the DatacenterBroker and Cloudlet classes are rewritten. The HWQPSO algorithm is implemented in DatacenterBroker, and the standard QPSO algorithm, DWC-QPSO algorithm and LTQPSO algorithm are reproduced. In order to test the search performance of this algorithm in cloud computing task scheduling, this section will compare the application of QPSO, DWC-QPSO, LTQPSO and HWQPSO algorithms in cloud platform task scheduling, and verify the application performance of this algorithm in cloud computing task scheduling from the following two perspectives [25-28].

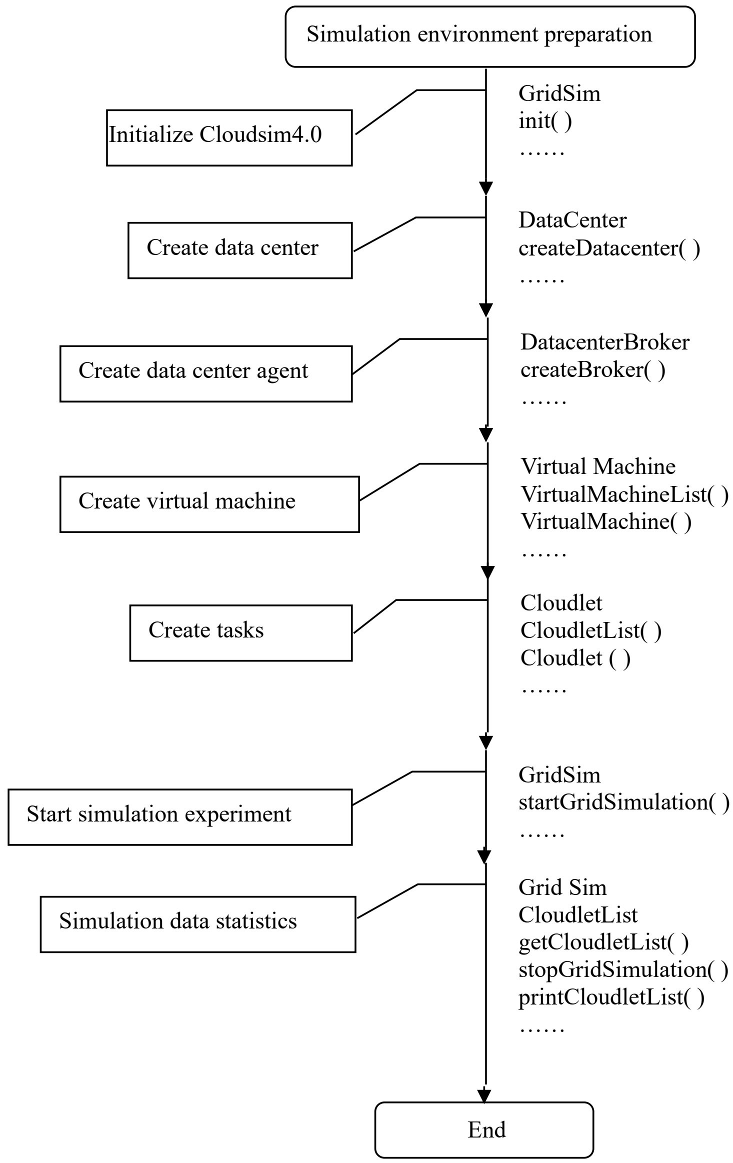


Fig. 11. Cloudsim4.0 simulation experiment process

Table 3. Cloud Sim simulator parameter list

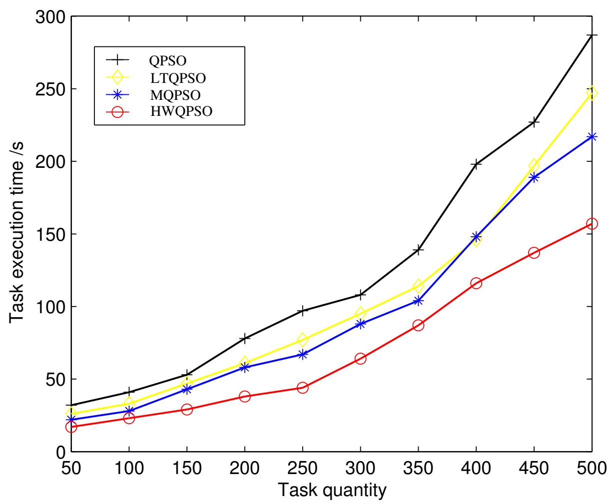
Type	Parameters	Value
Datacenter	number of Datacenter	6
	number of Host per Datacenter	2-5
	type of Manager	Space_shared/ Time_shared
Virtual Machine(VM)	total number of VMs	30
	number of PE per VM	4-12
	MIPS of PE (processing element)	300-1500 (MIPS)
	VM memory	512-2048(MB)
	Bandwidth	500-1000 bit
	Type of Manager	Time_shared
Task	Total number of task	50-500
	Length of task	5000-15000MI(Million Instruction)
	Number of PEs requirement	1-6

6.1. Task Execution Time Comparison Experiment

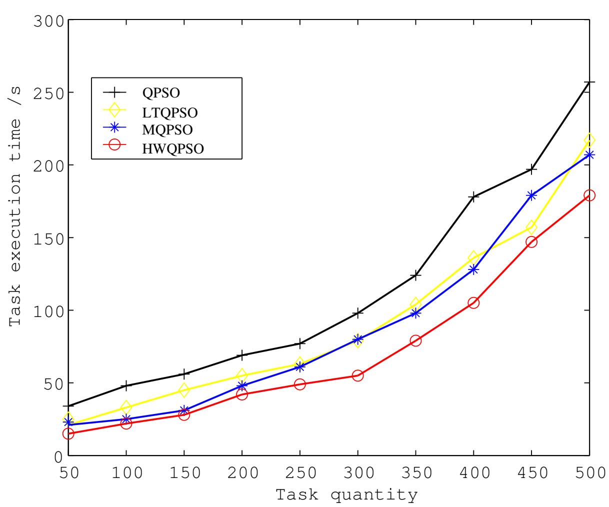
This experiment is based on the task scheduling schemes of QPSO, DWC-QPSO, LTQPSO, HWQPSO. The experiment is carried out under different scale tasks. The number of tasks is between [50,500], and the increment is 50. By randomly selecting the cloud computing resource configuration parameters, the execution time of four different task scheduling strategies was compared for four times, and the results are shown in figure 12.

According to the experimental results curves in figure 12, the red curve is the task scheduling time curve of the HWQPSO algorithm. The horizontal axis is the number of tasks, and the vertical axis is the time spent scheduling tasks. From the overall trend, the original QPSO algorithm takes the most time when scheduling the same number of tasks, while the HWQPSO algorithm in this paper takes the least time, and the DWC-QPSO and LTQPSO algorithm have their own advantages and disadvantages. As shown in figure 12(a) (d), when scheduling the same number of tasks, the red curve representing the HWQPSO algorithm is much lower than the black curve representing QPSO algorithm. The blue curve representing LTQPSO algorithm and the Yellow curve representing DWC-QPSO algorithm are interwoven, with high and low among them. For example, in figure 12(d), when the number of tasks is 350, the scheduling time of DWC-QPSO algorithm is more than that of LTQPSO algorithm, but when the number of tasks is 450, the scheduling time of DWC-QPSO algorithm is less than that of LTQPSO algorithm.

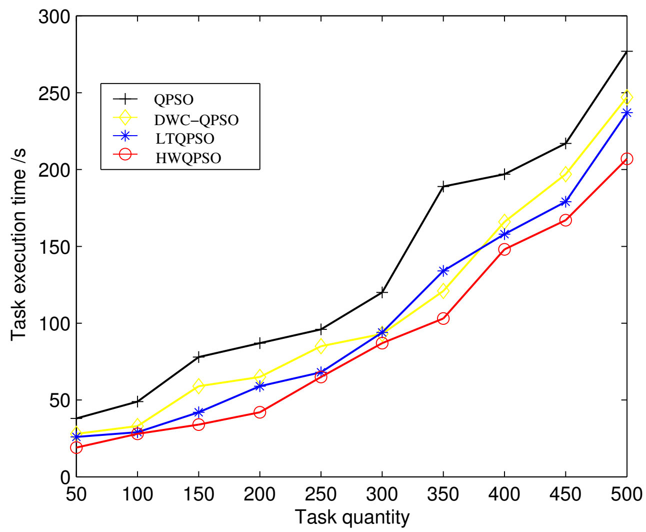
When the number of tasks is small, the difference of task scheduling time between the four algorithms is small, because the QPSO algorithm has strong optimization ability.



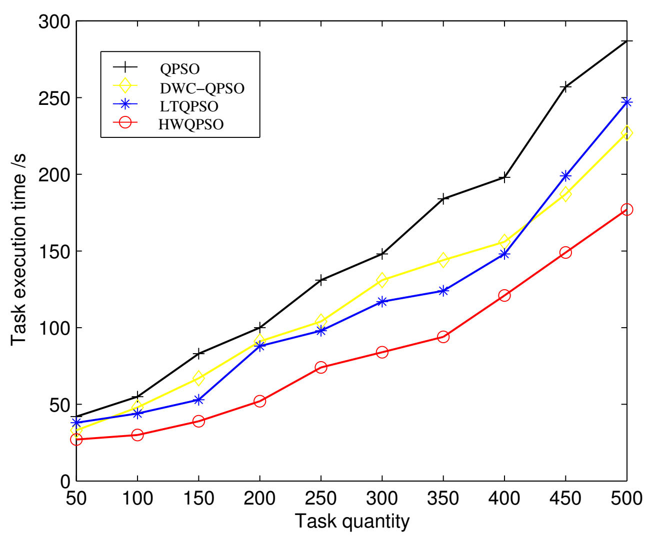
(a) The first experiment



(b) The second experiment



(c) The third experiment



(d) The fourth experiment

Fig. 12. Execution time comparison

When the small-scale task set scheduling is optimized, the optimal scheduling scheme can be found quickly. However, with the increase of task scale, the task scheduling efficiency of the HWQPSO algorithm is significantly higher than the other three. For example, when the number of tasks is 100, the task scheduling time gap of the four algorithms is much smaller than that of 500 tasks. This is because the algorithm in this paper improves the decision weight of excellent particles, so that all particles quickly approach to the optimal particles, and finally quickly converge to the optimal solution, that is to find the optimal scheduling scheme.

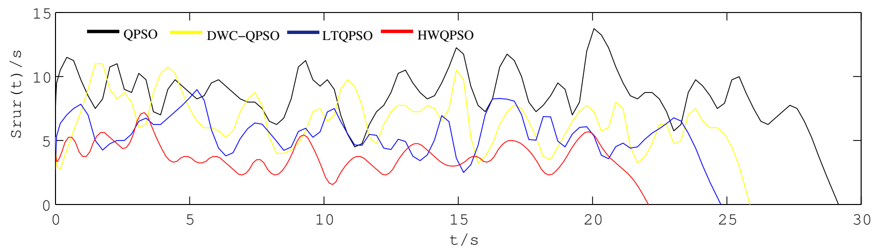
6.2. Computing Resource Load Comparison Experiment

This experiment is based on the task scheduling schemes of QPSO, DWC-QPSO, LTQPSO, HWQPSO. When the number of tasks is 100, 200, 300 and 400 respectively, the resource load of the four scheduling algorithms are compared. In the experiment, for each virtual machine, the time needed to complete all tasks scheduled to the virtual machine is used as the load measurement of the virtual machine. For the convenience of comparison, we calculate the standard deviation of computing resource load of all virtual machines in the cloud platform to describe the balance of computing resource load of the cloud platform at this moment. Assuming that there are N_{vm} virtual machines in the cloud platform in the current experiment, the computing resource load of the *i*-th virtual machine at time *t* is $u_i(t)$, then at time *t*, the calculation of the standard deviation $S_{rur}(t)$ of the computing resource load of all virtual machines in the cloud platform is shown in formula (19) and formula (20). The comparison experiment results of the cloud platform computing resource load of the four scheduling algorithms are shown in figure 13.

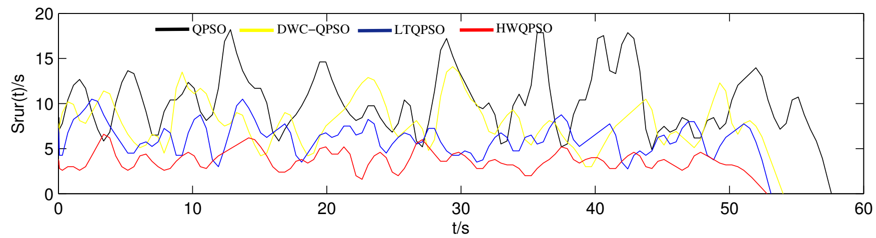
$$S_{rur}(t) = \sqrt{\frac{1}{N_{vm}} \sum_{i=1}^{N_{vm}} (u_i(t) - \overline{u(t)})^2} \qquad (19)$$

$$\overline{u(t)} = \frac{1}{N_{vm}} \sum_{i=1}^{N_{vm}} u_i(t) \qquad (20)$$

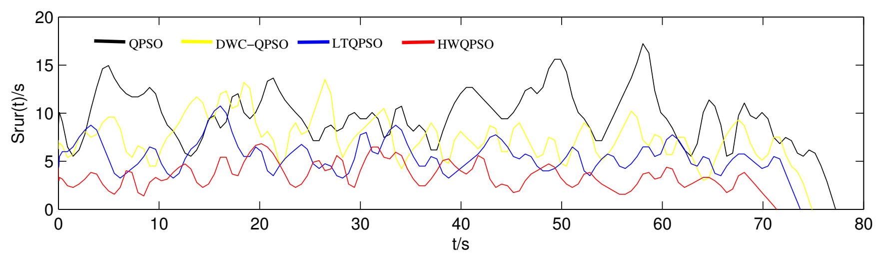
When the $S_{rur}(t)$ of computing resource load of all virtual machines is smaller, it means that the load of each virtual machine is relatively more balanced, and vice versa. According to the experimental results in Figure 13, the red curve is the load balance curve of the HWQPSO algorithm in resource scheduling, the standard deviation of computing resource load of all virtual machines in the cloud platform fluctuates greatly under different tasks of the four algorithms. As shown in Figure13(a)∼(d), the curves corresponding to the four algorithms are interleaved in varying degrees. However, from the overall trend analysis, the standard deviation of the HWQPSO algorithm in this paper is smaller than that of the other three algorithms, which shows that the load of each virtual machine is more balanced than that of the other three algorithms. And the standard deviation of computing resource load using the original QPSO algorithm is the largest, and the computing resource load is the most unbalanced, followed by the LTQPSO and DWC-QPSO algorithm. For example, in figure 13(a)∼(d), the overall trend of the black curve representing the load standard deviation of all virtual machine resources scheduled by QPSO algorithm is at the top, the red curve representing HWQPSO algorithm is at the bottom, and the blue



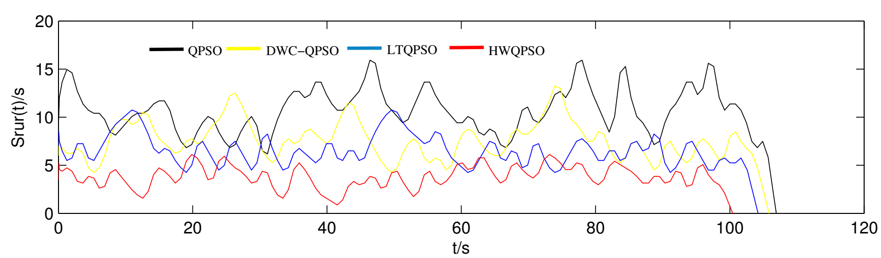
(a) When the number of tasks is 100



(b) When the number of tasks is 200



(c) When the number of tasks is 300



(d) When the number of tasks is 400

Fig. 13. Comparison of computing resource load

and yellow curves representing the LTQPSO and the DWC-QPSO algorithm tend to be in the middle. At the same time, in figure 13(a)$\sim$(d), it can be seen from the horizontal axis that HWQPSO scheduling time is the shortest, this also shows that HWQPSO algorithm has better optimization ability. And the red curve fluctuates the least, it shows that HWQPSO is more stable. All these benefit from the HWQPSO algorithm average optimal location classification weight strategy, improve the optimization accuracy of the algorithm and make the computing resources of each virtual machine can be better used, thus play a role of balancing the computing resources load.

7. Conclusion

In this paper, an average optimal position calculation method of the QPSO algorithm is proposed, which is based on the classification of particle fitness value, and it is used in cloud computing task scheduling. The selection of the average optimal position in the QPSO algorithm determines the global search ability and the final convergence speed of the algorithm. By setting high-level particles with high weight, we can improve the discourse power of excellent particles in the process of optimization, so that particles can quickly approach the optimal solution, to improve the search ability and efficiency of the algorithm. In this paper, five standard test functions are selected to test QPSO, DWC-QPSO, LTQPSO and HWQPSO. The experimental results show that the convergence accuracy and speed of the HWQPSO algorithm proposed in this paper are higher than those of the other three algorithms. At the same time, the HWQPSO algorithm proposed in this paper is applied to the task scheduling of the cloud computing platform. The performance of the HWQPSO algorithm proposed in this paper is tested by comparing the efficiency of the four algorithms QPSO, DWC-QPSO, LTQPSO and HWQPSO in the CloudSim4.0 simulation experiment platform. In the application, when scheduling the same number of tasks, the algorithm in this paper takes shorter time than the other three algorithms, and the load of computing resources is more balanced, so the efficiency of cloud platform is significantly improved. Experiments and application results show that the average optimal position calculation method based on particle fitness value classification improves the local search accuracy and global search ability of the QPSO algorithm, and the search stability is also improved.

Acknowledgment. This work was the achievements of the project" earth thesis project "of Guizhou Education University; And it was supported by the Youth Science and Technology Talents Development Project of Guizhou Education Department in 2018(No. Qian Jiao He KY[2018]257); Reform Project of Teaching Content and Course System in Guizhou Province in 2017(2017520065); 2016 Science and Technology Foundation Project of Guizhou Province(No. Qian Ke He Ji Chu [2016]1114); Key Disciplines of Guizhou Province-Computer Science and Technology (ZDXK [2018]007); Key Supported Disciplines of Guizhou Province-Computer Application Technology (No.QianXueWeiHeZi ZDXK[2016]20).

References

1. Van Den Bergh Frans. *An analysis of particle swarm optimizers*. University of Pretoria, 2002.

2. Yangyang Li, Xiaoyu Bai, Licheng Jiao, and Yu Xue. Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. *Applied Soft Computing*, 56:345–356, 2017.
3. Weiping Ding, Chin Teng Lin, Senbo Chen, Xiaofeng Zhang, and Bin Hu. Multiagent-consensus-mapreduce-based attribute reduction using co-evolutionary quantum pso for big data applications. *Neurocomputing*, page S0925231217311876, 2017.
4. Jun Sun, Bin Feng, and Wenbo Xu. Particle swarm optimization with particles having quantum behavior. In *Congress on Evolutionary Computation*, 2004.
5. Jun Sun, Wenbo Xu, and Bin Feng. A global search strategy of quantum-behaved particle swarm optimization. In *Proc IEEE Conference on Cybernetics & Intelligent Systems*, 2004.
6. Guoqiang Liu, Weiyi Chen, Huadong Chen, and Jiahui Xie. A quantum particle swarm optimization algorithm with teamwork evolutionary strategy. *Mathematical Problems in Engineering*, 2019(8):1–12, 2019.
7. Anupam Trivedi, Dipti Srinivasan, Subhodip Biswas, and Thomas Reindl. Hybridizing genetic algorithm with differential evolution for solving the unit commitment scheduling problem. *Swarm & Evolutionary Computation*, 23:50–64, 2015.
8. Qianqian Zhang, Shifeng Liu, Daqing Gong, Hankun Zhang, and Qun Tu. An improved multi-objective quantum-behaved particle swarm optimization for railway freight transportation routing design. *IEEE Access*, 7:157353–157362, 2019.
9. Zhen Lun Yang, Angus Wu, and Hua Qing Min. An improved quantum-behaved particle swarm optimization algorithm with elitist breeding for unconstrained optimization. *Comput Intell Neurosci*, 2015:1–12, 2015.
10. Tao Wu, Xi Chen, and Yusong Yan. Study of on-ramp pi controller based on dural group qpso with different well centers algorithm. *Mathematical Problems in Engineering*, 2015(PT.5):814871.1–814871.10, 2015.
11. Tao Xue, Renfu Li, Myongchol Tokgo, Junchol Ri, and Gyanghyok Han. Trajectory planning for autonomous mobile robot using a hybrid improved qpso algorithm. *Soft Computing*, 2017:2421–2437, 2017.
12. Wei Chen, Hong Yang, and Yifei Hao. Scheduling of dynamic multi-objective flexible enterprise job-shop problem based on hybrid qpso. *IEEE Access*, PP(99):1–1, 2019.
13. Lin Lin, Feng Guo, Xiaolong Xie, and Bin Luo. Novel adaptive hybrid rule network based on ts fuzzy rules using an improved quantum-behaved particle swarm optimization. *Neurocomputing*, 149(pt.b):1003–1013, 2015.
14. W. J. Chen, C. X. Lin, Y. T. Chen, and J. R. Lin. Optimization design of a gating system for sand casting aluminium a356 using a taguchi method and multi-objective culture-based qpso algorithm. *Mrs Bulletin*, 40(1):46–52, 2011.
15. Wen Jong Chen, Chuan Kuei Huang, Qi Zheng Yang, and Yin Liang Yang. Optimal prediction and design of surface roughness for cnc turning of al7075-t6 by using the taguchi hybrid qpso algorithm. *Transactions of the Canadian Society for Mechanical Engineering*, 40(5):883–895, 2016.
16. Jun Sun, Xiaojun Wu, Vasile Palade, Wei Fang, Choi Hong Lai, and Wenbo Xu. Convergence analysis and improvements of quantum-behaved particle swarm optimization. *Information Sciences*, 193(none):81–103, 2012.
17. Chunming Zhang, Yongchun Xie, Da Liu, and Li Wang. Fast threshold image segmentation based on 2d fuzzy fisher and random local optimized qpso. *IEEE Transactions on Image Processing*, 26(3):1355–1362, 2017.
18. Tianyu Liu, Licheng Jiao, Wenping Ma, Jingjing Ma, and Ronghua Shang. Cultural quantum-behaved particle swarm optimization for environmental/economic dispatch. *Applied Soft Computing*, 48:597–611, 2016.
19. Rajashree Nayak and Dipti Patra. An edge preserving ibp based super resolution image reconstruction using p-spline and mucso-qpso algorithm. *Microsystem Technologies*, 23(3):1–17, 2016.

20. Wei Zhang, Weifeng Shi, and Jinbao Zhuo. Bdi-agent-based quantum-behaved pso for shipboard power system reconfiguration. *International Journal of Computer Applications in Technology*, 55(1), 2017.
21. P. Sivakumar, S. Kalaiyarasi, and S. Shilpa. Qpso based load balancing mechanism in cloud environment. *ICSCAN*, 2019.
22. Zhong Kai Feng, Wen Jing Niu, and Chun Tian Cheng. Multi-objective quantum-behaved particle swarm optimization for economic environmental hydrothermal energy system scheduling. *Energy*, 131(Jul.15):165–178, 2017.
23. Obaid Ur Rehman, Shiyou Yang, Shafiullah Khan, and Sadaqat Ur Rehman. A quantum particle swarm optimizer with enhanced strategy for global optimization of electromagnetic devices. *IEEE Transactions on Magnetics*, PP(99):1–4, 2019.
24. R. Chakraborty, R. Sushil, and M. L. Garg. Icqpso-based multilevel thresholding scheme applied on colour image segmentation. *IET Signal Processing*, 3(13):387–395, 2019.
25. Xingquan Zuo, Guoxiang Zhang, and Tan Wei. Self-adaptive learning pso-based deadline constrained task scheduling for hybrid iaas cloud. *IEEE Transactions on Automation Science & Engineering*, 11(2):564–573, 2014.
26. S. Zhan and H. Huo. Improved pso-based task scheduling algorithm in cloud computing. *Journal of Information & Computational Science*, 9(13):3821–3829, 2012.
27. Jena and K. R. Multi objective task scheduling in cloud environment using nested pso framework. *Procedia Computer Science*, 57:1219–1227, 2015.
28. Mohammad Masdari, Farbod Salehi, Marzie Jalali, and Moazam Bidaki. A survey of pso-based scheduling algorithms in cloud computing. *Journal of Network & Systems Management*, 25(1):122–158, 2017.

Guolong Yu from Guiyang Guizhou Province, received Master degree in 2010 from Lanzhou University of Technology, China. Now, he is an associate professor in college of mathematics and big data, Guizhou Education University. His research interests include computational intelligence and big data.

Yong Zhao (Corresponding Author) from Shenzhen Guangdong Province, received Ph.D. degree in 1991 from Southeast University, China, and Postdoctoral in 2000 from Concordia University, Dept. of ECE. Montreal, Canada. Now, he is a professor in college of information engineering, Peking University. His research interests include deep learning and machine vision.

Zhongwei Cui from Guiyang Guizhou Province, received Ph.D. degree in 2019 from Guizhou University, China. Now, he is an associate professor in college of mathematics and big data, Guizhou Education University. His research interests include internet of things and machine vision.

Yu Zuo from Guiyang Guizhou Province, received Master degree in 2010 from Guizhou University, China. Now, he is a professor in college of mathematics and big data, Guizhou Education University. His research interests include machine vision and big data.

Received: February 23, 2020; Accepted: October 15, 2020.