



An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position

Maolong Xi^{a,b}, Jun Sun^a, Wenbo Xu^{a,*}

^aSchool of Information Technology, Jiangnan University, No. 1800, Lihu Avenue, WuXi, 214122 Jiangsu, China

^bWuXi Institute of Technology, No. 1600, Gaolang West Road, WuXi, 214121 Jiangsu, China

ARTICLE INFO

Keywords:

PSO
QPSO
Mean best position
Weight parameter
WQPSO

ABSTRACT

Quantum-behaved particle swarm optimization (QPSO) algorithm is a global convergence guaranteed algorithms, which outperforms original PSO in search ability but has fewer parameters to control. In this paper, we propose an improved quantum-behaved particle swarm optimization with weighted mean best position according to fitness values of the particles. It is shown that the improved QPSO has faster local convergence speed, resulting in better balance between the global and local searching of the algorithm, and thus generating good performance. The proposed improved QPSO, called weighted QPSO (WQPSO) algorithm, is tested on several benchmark functions and compared with QPSO and standard PSO. The experiment results show the superiority of WQPSO.

© 2008 Published by Elsevier Inc.

1. Introduction

Over the past several decades, population-based random optimization techniques, such as evolutionary algorithm and swarm intelligence optimization, have been widely employed to solve global optimization (GO) problems. Four well-known paradigms for evolutionary algorithms are genetic algorithms (GA) [1], evolutionary programming (EP) [2], evolution strategies (ES) [3] and genetic programming (GP) [4]. These methods are motivated by natural evolution. The particle swarm optimisation (PSO) method is a member of a wider class of swarm intelligence methods used for solving GO problems. The method was originally proposed by Kennedy as a simulation of social behaviour of bird flock and was first introduced as an optimisation method in 1995 [5]. Instead of using evolutionary operators to manipulate the individuals as in other evolutionary algorithms, PSO relies on the exchange of information between individuals. Each particle in PSO flies in search space with a velocity, which is dynamically adjusted according to its own former information. Since 1995, many attempts have been made to improve the performance of the PSO [6,7]. As far as the PSO itself concerned, however, it is not a global optimization algorithm, as has been demonstrated by Van den Bergh [8]. In [9,10], Sun et al. introduce quantum theory into PSO and propose a quantum-behaved PSO (QPSO) algorithm, which can be guaranteed theoretically to find optimal solution in search space. The experiment results on some widely used benchmark functions show that the QPSO works better than standard PSO and should be a promising algorithm.

In this paper, in order to balance the global and local searching abilities, we introduce a weight parameter in calculating the mean best position in QPSO to render the importance of particles in population when they are evolving, and thus proposed an improved quantum-behaved particle swarm optimization algorithm, weighted QPSO (WQPSO). The rest part of the paper is organized as follows. In Section 2, a brief introduction of PSO is given. The QPSO is introduced in Section 3. In Section 4, we propose the improved QPSO and show how to balance the searching abilities to guarantee the better convergence

* Corresponding author.

E-mail addresses: wx_xml@hotmail.com (M. Xi), sunjun_wx@hotmail.com (J. Sun), xwb@sytu.edu.cn (W. Xu).

speed of particles. Some experiments result on benchmark functions and discussions are presented in Section 5. Finally, the paper is concluded in Section 6.

2. PSO algorithms

Particle swarm optimization (PSO), first introduced by Kennedy and Eberhart [5], is a population-based optimization technique, where a population is called a swarm. A simple explanation of the PSO's operation is as follows. Each particle represents a possible solution to the optimization task at hand. During each iteration, the accelerating direction of one particle determined by its own best solution found so far and the global best position discovered so far by any of the particles in the swarm. This means that if a particle discovers a promising new solution, all the other particles will move closer to it, exploring the region more thoroughly in the process.

Let M denote the swarm size and n the dimensionality of the search space. Each individual i ($1 \leq i \leq M$) has the following attributes: A current position in the search space $X_i = (X_{i,1}, X_{i,2}, \dots, X_{i,n})$, a current velocity $V_i = (V_{i,1}, V_{i,2}, \dots, V_{i,n})$, and a personal best ($pbest$) position (the position giving the best fitness value experience by the particle) $P_i = (P_{i,1}, P_{i,2}, \dots, P_{i,n})$. At each iteration, each particle in the swarm updates its velocity according to (1), assuming that the function f is to be minimized, and that $r_1 \sim U(0, 1)$, $r_2 \sim U(0, 1)$ are two random numbers uniformly distributed in the interval $(0, 1)$

$$V_{ij}(t+1) = w \cdot V_{ij}(t) + c_1 \cdot r_{1,i}(t) \cdot [P_{ij}(t) - X_{ij}(t)] + c_2 \cdot r_{2,i}(t) \cdot [P_{gj}(t) - X_{ij}(t)] \quad (1)$$

for all $j \in 1, 2, \dots, n$, where V_{ij} is the velocity of the j th dimension of the i th particle, and c_1 and c_2 called the acceleration coefficients. The new position of a particle can be calculated using (2)

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1). \quad (2)$$

The personal best ($pbest$) position of each particle is updated using the following formula:

$$P_i(t+1) = \begin{cases} P_i(t), & \text{if } f(X_i(t+1)) \geq f(P_i(t)), \\ X_i(t+1), & \text{if } f(X_i(t+1)) \leq f(P_i(t)) \end{cases} \quad (3)$$

and the global best ($gbest$) position found by any particle during all previous iterations, P_g , is defined as

$$P_g(t+1) = \arg \min_{P_i} f(P_i(t+1)), \quad 1 \leq i \leq M. \quad (4)$$

The value of each component in every V_i vector should be clamped to the range $[-V_{\max}, V_{\max}]$ to reduce the likelihood of particles' leaving the search space. The value of V_{\max} is usually chosen to be $k \times X_{\max}$, with $0.1 \leq k \leq 1.0$, where X_{\max} is the up-limit of the search scope on each dimension [14,15].

The parameter w in (1) is called the inertia weight that is typically set up to vary linearly from 0.9 to 0.4 during the course of search process. The inclusion of inertia weight leads to faster convergence of the PSO algorithm.

The acceleration coefficients c_1 and c_2 can be used to control how far a particle will move in a single iteration and thus may exert an great influence on the convergence speed of PSO. Typically, these are both set to a value of 2.0, although assigning different values to c_1 and c_2 sometimes leads to better performance.

An other version of PSO with introduction of constriction factor in the algorithm owes to Clerc [16]. The update equation of the velocity in this version is as follows:

$$V_{ij}(t+1) = \chi[V_{ij}(t) + c_1 \cdot r_{1,i} \cdot (P_{ij}(t) - X_{ij}(t)) + c_2 \cdot r_{2,i} \cdot (P_{gj}(t) - X_{ij}(t))], \quad (5)$$

where

$$\chi = \frac{2}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|} \quad (6)$$

and $\varphi = c_1 + c_2$, $\varphi > 4$. It has been suggested that a constriction factor may help to ensure convergence. The constriction factor, as shown in (5) and (6), could replace the V_{\max} clamping.

3. Quantum-behaved particle swarm optimization

Trajectory analyses in [16] demonstrated that, to guarantee convergence of PSO algorithm, each particle must converge to its local attractor $p_i = (p_{i,1}, p_{i,2}, \dots, p_{i,n})$, of which the coordinates are defined as

$$p_{ij}(t) = (c_1 P_{ij}(t) + c_2 P_{gj}(t)) / (c_1 + c_2), \quad j = 1, 2, \dots, n \quad (7)$$

or

$$p_{ij}(t) = \varphi \cdot P_{ij}(t) + (1 - \varphi) \cdot P_{gj}(t), \quad \varphi \sim U(0, 1), \quad j = 1, 2, \dots, n, \quad (8)$$

where $\varphi = c_1 r_1 / (c_1 r_1 + c_2 r_2)$. It can be seen that the local attractor is a stochastic attractor of particle i that lies in a hyper-rectangle with P_i and P_g being two ends of its diagonal. We introduce the concepts of QPSO as follows.

Assume that each individual particle move in the search space with a δ potential on each dimension, of which the center is the point p_{ij} . For simplicity, we consider a particle in one-dimensional space, with point p the center of potential. Solving Schrödinger equation of one-dimensional δ potential well, we can get the probability density function Q and distribution function F

$$Q(X_{ij}(t+1)) = \frac{1}{L_{ij}(t)} e^{-2|p_{ij}(t)-X_{ij}(t+1)|/L_{ij}(t)}, \quad (9)$$

$$F(X_{ij}(t+1)) = e^{-2|p_{ij}(t)-X_{ij}(t+1)|/L_{ij}(t)}, \quad (10)$$

where $L_{ij}(t)$ is standard deviation of the distribution, which determines search scope of each particle. Employing Monte Carlo method for (11), we can obtain the position of the particle using following equation:

$$X_{ij}(t+1) = p_{ij}(t) \pm \frac{L_{ij}(t)}{2} \ln(1/u) \quad u = \text{rand}(0, 1), \quad (11)$$

where u is a random number uniformly distributed in $(0, 1)$.

To evaluate $L_{ij}(t)$, in [10], a global point called Mainstream Thought or mean best position of the population is introduced into PSO. The global point, denoted as m , is defined as the mean of the $pbest$ positions of all particles. That is

$$m(t) = (m_1(t), m_2(t), \dots, m_3(t)) = \left(\frac{1}{M} \sum_{i=1}^M P_{i,1}(t), \frac{1}{M} \sum_{i=1}^M P_{i,2}(t), \dots, \frac{1}{M} \sum_{i=1}^M P_{i,n}(t) \right), \quad (12)$$

where M is the population size and P_i is the $pbest$ position of particle i . The values of $L_{ij}(t)$ is determined by

$$L_{ij}(t) = 2\beta \cdot |m_j(t) - X_{ij}(t)| \quad (13)$$

and thus the position can be calculated by

$$X_{ij}(t+1) = p_{ij}(t) \pm \beta \cdot |m_j(t) - X_{ij}(t)| \cdot \ln(1/u), \quad (14)$$

where parameter β is called contraction–expansion coefficient, which can be tuned to control the convergence speed of the algorithms. The PSO with Eq. (14) is called quantum-behaved particle swarm optimization (QPSO). The QPSO algorithm is described as follows.

```

Initialize population: random X[i] and set P[i]=X[i];
do
  find out mbest using Eq. (12);
  for i=1 to population size M
    if f(Xi)<f(Pi) then P[i]=X[i];
    g=arg min(f(P[i]));
    for j=1 to dimensionality n
       $\eta = \text{rand}(0,1)$ ;
       $p[i][j]=\eta * P[i][j] + (1-\eta) * P[g][j]$ ;
       $u = \text{rand}(0,1)$ 
      if  $\text{rand}(0,1) > 0.5$ 
         $X[i][j]=p[i][j]-\beta * \text{abs}(m[j]-X[i][j]) * \ln(1/u)$ ;
      else
         $X[i][j]=p[i][j]+\beta * \text{abs}(m[j]-X[i][j]) * \ln(1/u)$ ;
      endif
    endfor
  endfor
Until termination criterion is met

```

4. Weighted quantum-behaved particle swarm optimization

As mentioned above, in the QPSO, the mean best position m is introduced to evaluate the value of L , making the algorithm more efficient than that proposed in [9]. From Eq. (12), we can see that the mean best position is simply the average on the personal best position of all particles, which means that each particle is considered equal and exert the same influence on the value of m . The philosophy of this method is that the Mainstream Thought, that is, mean best position m , determines the search scope or creativity of the particle [10]. The definition of the Mainstream Thought as mean of the personal best positions is somewhat reasonable. The equally weighted mean position, however, is something of paradox, compared with the evolution of social culture in real world. For one thing, although the whole social organism determines the Mainstream Thought, it is not properly to consider each member equal. In fact, the elitists play more important role in culture

development. With this in mind when we design a new control method for the QPSO in this paper, m in Eq. (12) is replaced for a weighted mean best position.

The most important problem is to determine whether a particle is an elitist or not, or say it exactly, how to evaluate its importance in calculate the value of m . It is natural, as in other evolutionary algorithm, that we associate elitism with the particles' fitness value. The greater the fitness, the more important the particle is. Describing it formally, we can rank the particle in descendent order according to their fitness value first. Then assign each particle a weight coefficient α_i linearly decreasing with the particle's rank, that is, the nearer the best solution, the larger its weight coefficient is. The mean best position m , therefore, is calculated as

$$m(t) = (m_1(t), m_2(t), \dots, m_3(t)) = \left(\frac{1}{M} \sum_{i=1}^M \alpha_{i,1} P_{i,1}(t), \frac{1}{M} \sum_{i=1}^M \alpha_{i,2} P_{i,2}(t), \dots, \frac{1}{M} \sum_{i=1}^M \alpha_{i,n} P_{i,n}(t) \right) \quad (15)$$

where α_i is the weight coefficient and α_{id} is the dimension coefficient of every particle, M is the population size. In this paper, the weight coefficient for each particle decreases linearly from 1.5 to 0.5.

The improved algorithm is called Weighted QPSO that is outlined as follows.

```

Initialize population: random X[i] and set P[i]=X[i];
do
  find out mbest using Eq. (15);
  for i=1 to population size M
    if f(Xi)<f(Pi) then P[i]=X[i];
    g=arg min(f(P[i]));
    for j=1 to dimensionality n
      η=rand(0,1);
      p[i][j]=η*P[i][j]+(1-η)*P[g][j];
      u=rand(0,1)
      if rand(0,1)>0.5
        X[i][j]=p[i][j]-β*abs(m[j]-X[i][j])*ln(1/u);
      else
        X[i][j]=p[i][j]+β*abs(m[j]-X[i][j])*ln(1/u);
      endif
    endfor
  endfor
Until termination criterion is met

```

The QPSO or WQPSO is much different from the PSO in that the update equation (14) of QPSO ensures the particle's appearing in the whole n -dimensional search space at each iteration, while the particle in the PSO can only fly in a bounded space at each iteration. Employing the global convergence criterion in [16], we can conclude that the QPSO or WQPSO is a global convergent algorithm and the PSO is not.

5. Experiment results

To test the performance of the improved QPSO (WQPSO), five benchmark functions listed in Table 1 are used here for comparison with QPSO algorithm in [10].

These functions are all minimization problems with minimum value zeros. In our experiments, the initial range of the population in Table 1 is asymmetry as used in [9–11].

The fitness value is set as function value and the neighborhood of a particle is the whole population. We had 100 trial runs for every instance and recorded mean best fitness and standard deviation. In order to investigate the scalability of the algorithm, different population sizes M are used for each function with different dimensions. The population sizes are 20, 40 and 80 and the maximum generation is set as 1000, 1500 and 2000 corresponding to the dimensions 10, 20 and 30 for five func-

Table 1
Benchmark functions

| Functions | Formulations | Initialization | Max. range |
|---------------------------|---|----------------|------------|
| Sphere function f_1 | $f_1(x) = \sum_{i=1}^n x_i^2$ | (50, 100) | 100 |
| Rosenbrock function f_2 | $f_2(x) = \sum_{i=1}^n (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ | (15, 30) | 100 |
| Rastrigrin function f_3 | $f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$ | (2.56, 5.12) | 10 |
| Griewank function f_4 | $f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$ | (300, 600) | 600 |
| De Jong's function f_5 | $f_5(x) = \sum_{i=1}^n i x_i^4$ | (30, 100) | 100 |

tions, respectively. We make three groups of experiments, with each to test the standard PSO (SPSO), the QPSO and the weight QPSO (WQPSO). In the first group of experiments, as in most exiting literature on PSO, the acceleration coefficient is set as $c_1 = c_2 = 2$, the inertia weight w decreases linearly from 0.9 to 0.4 and value of V_{max} is set as the scope of the search space for each cases. In the second set of experiments, the QPSO is tested, and the coefficient β decreases from 1.0 to 0.5 linearly when the algorithm is running as in [10]. The third is to test the performance of WQPSO with the coefficient β also decreases from 1.0 to 0.5 linearly while parameter α decreases from 1.5 to 0.5 linearly. The best fitness values for 100 runs of each function in Tables 2–5.

The numerical results in Table 2 show that the WQPSO could hit the optimal solution with high precision. Because the Sphere function has only a single optimal solution on origin, it usually is employed to test the local search ability of the algorithm. Thus from the result, we can see that WQPSO has stronger local search ability than PSO and QPSO. The Rosenbrock function is a mono-modal function, but its optimal solution lies in a narrow area that the particles are always apt to escape. Therefore, it is always used to test the local and global search ability of the algorithm. The experiment results on Rosenbrock function show that the WQPSO works better than QPSO algorithm in most cases. When the dimension is 80, the proposed technique is not able to generate better results than original QPSO. The advantage of WQPSO on QPSO and PSO may be owed

Table 2
The mean fitness value for Sphere function

| M | D | Ger. | PSO | | QPSO | | WQPSO | |
|----|----|------|-----------|----------|-------------|-------------|-------------|-------------|
| | | | Mean best | St. Var. | Mean best | St. Var. | Mean best | St. Var. |
| 20 | 10 | 1000 | 3.16e–20 | 6.23e–20 | 1.3909e–041 | 1.4049e–043 | 2.2922e–056 | 1.5365e–058 |
| | 20 | 1500 | 5.29e–11 | 1.56e–10 | 3.5103e–022 | 3.5452e–024 | 2.9451e–040 | 2.8717e–042 |
| | 30 | 2000 | 2.45e–06 | 7.72e–06 | 5.3183e–014 | 5.3623e–016 | 3.9664e–033 | 3.8435e–035 |
| 40 | 10 | 1000 | 3.12e–23 | 8.01e–23 | 2.5875e–071 | 2.6137e–073 | 5.5806e–080 | 5.6370e–082 |
| | 20 | 1500 | 4.16e–14 | 9.73e–14 | 3.7125e–042 | 3.7500e–044 | 8.8186e–055 | 7.1785e–057 |
| | 30 | 2000 | 2.26e–10 | 5.10e–10 | 4.2369e–030 | 1.7009e–033 | 5.4389e–044 | 2.4132e–045 |
| 80 | 10 | 1000 | 6.15e–28 | 2.63e–27 | 8.5047e–102 | 7.5974e–104 | 4.7144e–106 | 4.7620e–108 |
| | 20 | 1500 | 2.68e–17 | 5.24e–17 | 1.1542e–068 | 1.1585e–070 | 2.5982e–074 | 2.6243e–076 |
| | 30 | 2000 | 2.47e–12 | 7.16e–12 | 2.2866e–049 | 2.3070e–051 | 2.1422e–060 | 1.9125e–062 |

Table 3
The mean fitness value for Rosenbrock function

| M | D | Ger. | PSO | | QPSO | | WQPSO | |
|----|----|------|-----------|----------|-----------|----------|-----------|----------|
| | | | Mean best | St. Var. | Mean best | St. Var. | Mean best | St. Var. |
| 20 | 10 | 1000 | 94.1276 | 194.3648 | 51.9761 | 0.4737 | 35.8436 | 0.2843 |
| | 20 | 1500 | 204.337 | 293.4544 | 136.8782 | 0.6417 | 62.7696 | 0.4860 |
| | 30 | 2000 | 313.734 | 547.2635 | 157.4707 | 0.8287 | 70.9525 | 0.4283 |
| 40 | 10 | 1000 | 71.0239 | 174.1108 | 17.3177 | 0.1515 | 16.9583 | 0.1336 |
| | 20 | 1500 | 179.291 | 377.4305 | 54.0411 | 0.4210 | 54.2439 | 0.3752 |
| | 30 | 2000 | 289.593 | 478.6273 | 81.1382 | 0.0319 | 57.0883 | 0.3437 |
| 80 | 10 | 1000 | 37.3747 | 57.4734 | 7.5755 | 0.2708 | 10.1650 | 0.2345 |
| | 20 | 1500 | 83.6931 | 137.2637 | 32.9970 | 0.2068 | 47.0275 | 0.3507 |
| | 30 | 2000 | 202.672 | 289.9728 | 53.6422 | 0.2616 | 51.8299 | 0.3103 |

Table 4
The mean fitness value for Rastrigrin function

| M | D | Ger. | PSO | | QPSO | | WQPSO | |
|----|----|------|-----------|----------|-----------|-------------|-----------|----------|
| | | | Mean best | St. Var. | Mean best | St. Var. | Mean best | St. Var. |
| 20 | 10 | 1000 | 5.5382 | 3.0477 | 4.8274 | 0.0015 | 4.0567 | 0.0094 |
| | 20 | 1500 | 23.1544 | 10.4739 | 16.0519 | 0.0414 | 12.1102 | 0.0287 |
| | 30 | 2000 | 47.4168 | 17.1595 | 33.7218 | 0.0114 | 23.5593 | 0.0713 |
| 40 | 10 | 1000 | 3.5778 | 2.1384 | 3.1794 | 6.0338e–004 | 2.8163 | 0.0083 |
| | 20 | 1500 | 16.4337 | 5.4811 | 10.8824 | 0.0496 | 9.7992 | 0.0628 |
| | 30 | 2000 | 37.2796 | 14.2838 | 21.4530 | 0.0949 | 17.4436 | 0.0034 |
| 80 | 10 | 1000 | 2.5646 | 1.5728 | 2.2962 | 0.0130 | 1.8857 | 0.0118 |
| | 20 | 1500 | 13.3826 | 8.5137 | 7.8544 | 0.0011 | 7.2855 | 0.0032 |
| | 30 | 2000 | 28.6293 | 10.3431 | 15.9474 | 0.0198 | 15.0255 | 0.0294 |

Table 5

The mean fitness value for Griewank function

| M | D | Ger. | PSO | | QPSO | | WQPSO | |
|----|----|------|-----------|----------|-------------|----------|-------------|-------------|
| | | | Mean best | St. Var. | Mean best | St. Var. | Mean best | St. Var. |
| 20 | 10 | 1000 | 0.09217 | 0.0833 | 5.5093e-004 | 0.0657 | 5.6353e-004 | 5.5093e-004 |
| | 20 | 1500 | 0.03002 | 0.03255 | 1.0402e-004 | 0.0211 | 2.1318e-004 | 1.0402e-004 |
| | 30 | 2000 | 0.01811 | 0.02477 | 1.2425e-004 | 0.0110 | 2.1286e-004 | 1.2425e-004 |
| 40 | 10 | 1000 | 0.08496 | 0.0726 | 1.6026e-004 | 0.0496 | 0.0020 | 1.6026e-004 |
| | 20 | 1500 | 0.02719 | 0.02517 | 1.7127e-004 | 0.0167 | 1.6861e-004 | 1.7127e-004 |
| | 30 | 2000 | 0.01267 | 0.01479 | 3.9088e-005 | 0.0085 | 3.6762e-005 | 3.9088e-005 |
| 80 | 10 | 1000 | 0.07484 | 0.07107 | 3.3744e-004 | 0.0327 | 1.5281e-004 | 3.3744e-004 |
| | 20 | 1500 | 0.02854 | 0.0268 | 4.1701e-004 | 0.0168 | 3.2549e-004 | 4.1701e-004 |
| | 30 | 2000 | 0.01258 | 0.01396 | 1.3793e-005 | 0.0106 | 4.2231e-005 | 1.3793e-005 |

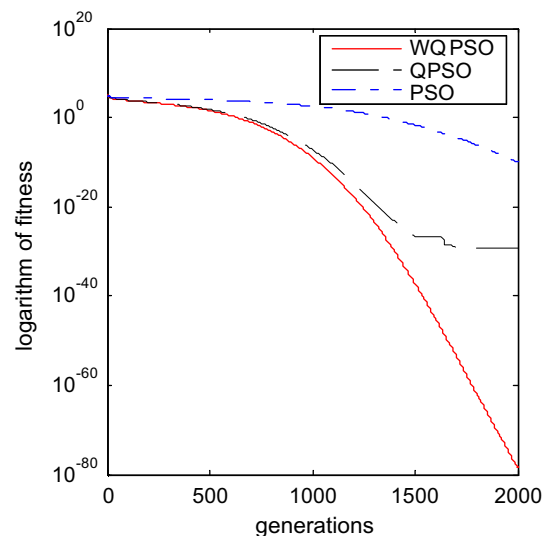
Table 6

The mean fitness value for De Jong's function

| M | D | Ger. | PSO | | QPSO | | WQPSO | |
|----|----|------|-------------|-------------|-------------|-------------|-------------|-------------|
| | | | Mean best | St. Var. | Mean best | St. Var. | Mean best | St. Var. |
| 20 | 10 | 1000 | 1.0403e-032 | 4.9825e-033 | 4.0921e-066 | 4.1335e-068 | 1.2729e-082 | 1.2858e-084 |
| | 20 | 1500 | 2.9108e-015 | 6.9309e-016 | 3.1303e-030 | 3.1619e-032 | 3.8146e-053 | 3.8172e-055 |
| | 30 | 2000 | 3.6326e-008 | 1.5575e-008 | 8.3442e-018 | 8.3338e-020 | 4.1471e-043 | 4.1885e-045 |
| 40 | 10 | 1000 | 1.5754e-040 | 1.104e-040 | 4.3593e-113 | 4.4034e-115 | 5.7373e-122 | 5.7953e-124 |
| | 20 | 1500 | 4.3514e-019 | 1.8339e-019 | 1.1619e-057 | 1.1736e-059 | 1.3193e-073 | 1.1409e-075 |
| | 30 | 2000 | 1.6261e-011 | 8.1139e-012 | 4.0999e-038 | 4.1413e-040 | 2.6766e-054 | 2.2684e-056 |
| 80 | 10 | 1000 | 4.0818e-047 | 2.0426e-047 | 1.8732e-163 | 0 | 3.3279e-171 | 0 |
| | 20 | 1500 | 5.8958e-024 | 2.0068e-024 | 7.2359e-095 | 7.3081e-097 | 3.7806e-103 | 5.0388e-106 |
| | 30 | 2000 | 3.8331e-015 | 1.0922e-015 | 2.3849e-063 | 2.4090e-065 | 7.2855e-074 | 7.3571e-076 |

to its local search ability as well as global search ability. Rastrigrin function and Griewank function are both multi-modal and usually tested for comparing the global search ability of the algorithm. On Rastrigrin function and Griewank function the WQPSO has better performance than QPSO algorithm; the advantage of the former to the latter is not remarkable. On De Jong's function, the experiment results show that WQPSO is superior to QPSO significantly for the same reason as in the experiments on Sphere function (see Table 6).

Figs. 1–5 give the comparison of convergence processes of WQPSO and QPSO in the above five benchmark functions averaged on 100 trial runs, when the population size is 20 and the maximum generation is 2000 according to the dimension 30

**Fig. 1.** Comparison of convergence process with two algorithms in Sphere function.

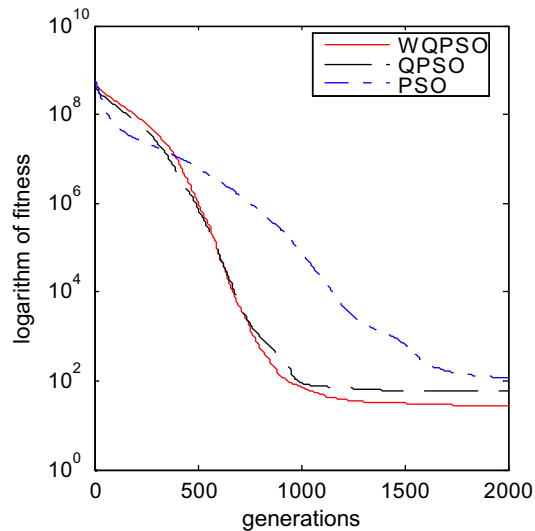


Fig. 2. Comparison of convergence process with two algorithms in Rosenbrock function.

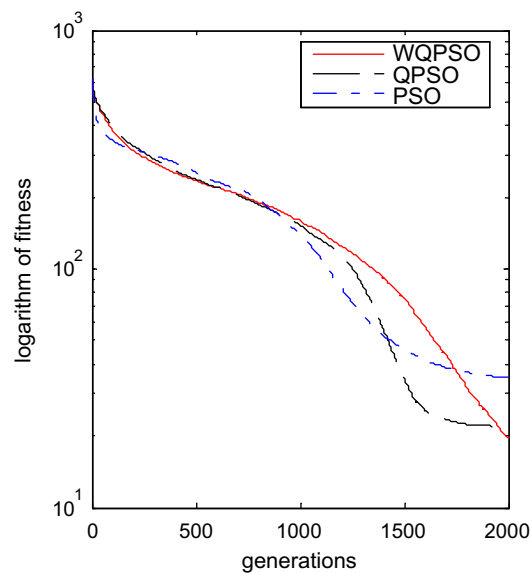


Fig. 3. Comparison of convergence process with two algorithm in Rastrigrin function.

for five benchmarks. The coefficient β also decreases from 1.0 to 0.5 linearly while parameter α decreases from 1.5 to 0.5 linearly. It can be found out that WQPSO has fastest convergence speed than QPSO and PSO. Since the parameter setting in WQPSO is the same as in QPSO, the fast convergence of WQPSO may be due to the weighted mean best position, which makes the particle converge to global best position more quickly. The global search ability of WQPSO, however, is not enhanced by the proposed method.

From the results above in the tables, we can conclude that the calculation method of mean best position with weight parameter introduced can make the convergence speed of QPSO faster, which may lead to good performance of the algorithm on convex function such as Sphere, Rosenbrock and De Jong's. On complex multi-modal function optimization problems, the tradeoff between global search ability and local search ability is vital to the performance of the algorithm. The slow convergence speed corresponds to good global search ability, while fast speed results in good local search ability. It can be conclude by the results on Sphere function and De Jong's function that faster convergence speed leads to stronger local ability of WQPSO. The results on Rastrigrin and Griewank function show the better performance of WQPSO, although the weighted mean best position may not generate better global search ability, since for the optimization problem, tradeoff between

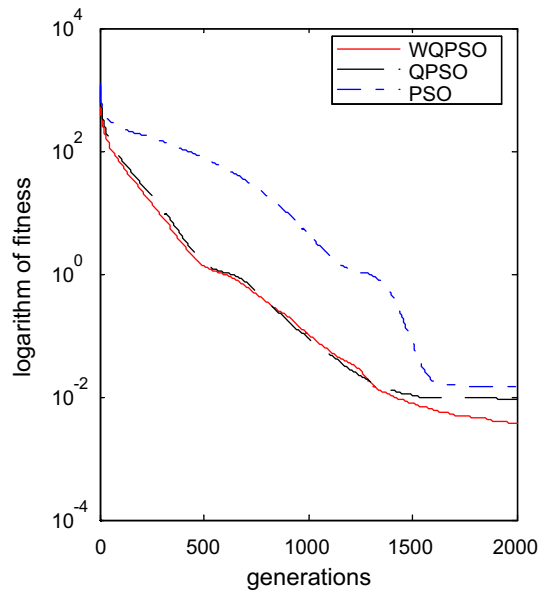


Fig. 4. Comparison of convergence process with two algorithms in Griewank function.

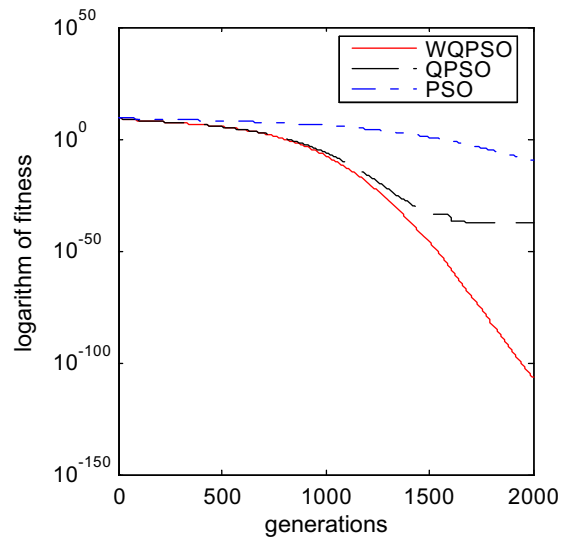


Fig. 5. Comparison of convergence process with two algorithms in De Jong's function.

exploration and exploitation during the search is critical to the performance at the end of the search. It can be concluded that WQPSO is excellent in doing so.

6. Conclusions

In this paper, we have described the quantum-behaved particle swarm optimization and how to using the mean best position of the population to guide the search of the swarm. Based on analysis the mean best position, we introduced a linearly decreasing weight parameter to render the importance of particles in population when they are evolving, and proposed revised QPSO. This method is more approximate to the learning process of social organism with high-level swarm intelligence and can make the population evolve rapidly with good global searching ability. In our future work, we will be devoted to find out a more efficient parameter control method, and thus to enhance the performance of QPSO further.

References

- [1] D.E. Goldberg, Genetic Algorithms in search Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.
- [2] L.J. Fogel, Evolutionary programming in perspective: the top-down view, in: J.M. Zurada, R.J. Marks II, C.J. Robinson (Eds.), Computational Intelligence: Imitating Life, IEEE Press, Piscataway, NJ, 1994.
- [3] I. Rechenberg, Evolution strategy, in: J.M. Zurada, R.J. Marks II, C.J. Robinson (Eds.), Computational Intelligence: Imitating Life, IEEE Press, Piscataway, NJ, 1994.
- [4] J.R. Koza, Genetic Programming: On The Programming of Computers by Means of Natural Selection, MIT Press, Cambridge, MA, 1992.
- [5] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference On Neural Network, 1995, pp. 1942–1948.
- [6] Y.L. Zheng, L.H. Ma, L.Y. Zhang, Empirical study of particle swarm optimizer with an increasing inertia weight, in: Proceedings of IEEE Congress on Evolutionary Computation 2003 (CEC 2003), Canbella, Australia, 2003, pp. 221–226.
- [7] M. Clerc, Discrete Particle Swarm Optimization, New Optimization Techniques in Engineering, Springer Verlag, 2004.
- [8] F. Van den Bergh, An Analysis of Particle Swarm Optimizers, Ph.D. Thesis, University of Pretoria, November 2001.
- [9] J. Sun, B. Feng, W.B. Xu, Particle swarm optimization with particles having quantum behavior, in: IEEE Proceedings of Congress on Evolutionary Computation, 2004, pp. 325–331.
- [10] J. Sun, W.B. Xu, B. Feng, a global search strategy of quantum-behaved particle swarm optimization, in: Cybernetics and Intelligent Systems Proceedings of the 2004 IEEE Conference, 2004, pp. 111–116.
- [11] J. Sun, W.B. Xu, B. Feng, Adaptive parameter control for quantum-behaved particle swarm optimization on individual level, in: IEEE International Conference on Systems, Man and Cybernetics, 2005, pp. 3049–3054.
- [14] Y. Shi, R. Eberhart, Empirical study of particle swarm optimization, in: Proceedings of Congress on Evolutionary Computation, Piscataway, NJ, 1999, pp. 1945–1950.
- [15] Y. Shi, R.C. Eberhart, A modified particle swarm. in: Proceedings, IEEE International Conference on Evolutionary Computation, Piscataway, NJ, 1998, pp. 69–73.
- [16] M. Clerc, J. Kennedy, The particle swarm: explosion, stability, and convergence in a multi-dimensional complex space, in: IEEE Transactions on Evolutionary Computation 6(1), Piscataway, NJ, 2002, pp. 58–73.

Further reading

- [12] J. Sun, W.B. Xu, W. Fang, Enhancing Global Search Ability of Quantum-Behaved Particle Swarm Optimization by Maintaining Diversity of the Swarm, in: RSTC 2006, 2006, pp. 736–745.
- [13] J. Sun, W.B. Xu, W. Fang, quantum-behaved particle swarm optimization algorithm with controlled diversity, in: International Conference on Computational Science (3) 2006, 2006, pp. 847–854.