Improved Quantum-behaved Particle Swarm Optimization Algorithm with Memory and Signal Step Searching Strategy for Continuous Optimization Problems

Deyu TANG\textsuperscript{1,*}, Yongming CAI\textsuperscript{1}, Xianfa CAI\textsuperscript{2}

\textsuperscript{1}College of Medical Information and Engineering, GuangDong Pharmaceutical University, Guangdong 510006, China
\textsuperscript{2}College of Computer Science and Engineering, South China University of Technology, Guangdong 510006, China

Abstract

Quantum-behaved particle swarm optimization (QPSO) algorithm is a global convergence guaranteed algorithms, which has been applied widely for continuous optimization problems. In this paper, we propose an improved quantum-behaved particle swarm optimization with memory according to the means of best position of particles and using signal step searching strategy for solve the multidimensional problem. At the same time, Gaussian distribution was used for the stochastic coefficients and uniform distribution was used for the weight of all the best particles. The proposed improved QPSO is tested on several benchmark functions and compared with standard PSO, standard SFLA, RQPSO and WQPSO. The experiment results show the superiority of our algorithm (called MSQPSO).

Keywords: Memory; Signal Step Searching Strategy; Weighted Best Position; QPSO; Swarm Intelligence

1 Introduction

Many difficulties such as multimodality, dimensionality are associated with the continuous optimization problems. Over the past several decades, nature-inspired meta-heuristic optimization technique, such as evolutionary algorithm and swarm intelligence optimization, have been widely employed to solve global optimization problems. Some of the well known meta-heuristic algorithm are: genetic algorithms\cite{1}, Differential Evolution (DE) \cite{2, 3} which is similar to GA with specialized crossover and selection method. Ant Colony Optimization (ACO) \cite{4} which works on the foraging behavior of the ant for the food; Particle Swarm Optimization (PSO) \cite{5, 6, 7} which works on the foraging behavior of the swarm of birds; SFLA which works on the foraging behavior of the frog for the food \cite{8, 9}. Recently, inspired by quantum mechanics and trajectory analysis of PSO, Sun et al. used a strategy based on a quantum potential well model to sample around...
the previous best points[10], and later introduced the mean best position into the algorithm and proposed a new version of PSO, quantum-behaved particle swarm optimization (QPSO)[11, 12]. Since QPSO was proposed, a set of different mutation operations on the personal best positions of the particles in QPSO were studied in [13, 14, 15]. But these methods do not fully consider the particle search history, the algorithm is easy to fall into local optimal solution.

In this paper, we introduce an improved QPSO algorithm with the strong search capabilities. In order to strengthen the diversity of the particle attractor, we set different weights for all the average particle attractors using uniformed distribution. At the same time, each average attractor has a memory function, which guide all the particles searching to the global direction. The difference compared with the previous algorithms is that we also used a single search technology to improve the exploring ability abilityf of the QPSO algorithm. 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 In Section 5, Experiment show the result of our algorithm.

2 PSO Algorithm

PSO is a population-based optimization tool. It is firstly proposed by American social psychologist J.Kennedy and electrical engineer Russell Eberhart in the IEEE international neural network conference in 1995. The basic idea is inspired by their early studying result of bird group behavior and use the biological population model by the biologist Frank Heppner to solve optimization problems. At present, the study on PSO algorithm with inertia weight is mostly based on the PSO algorithm as a basis for expansion and modification, and the PSO with inertia weight algorithm was called the standard PSO algorithm. The algorithm consider each individual as a particle without weight and volum in n dimensional search space, The flight speed of individual was dynamic adjusted by flight experience of individual and flying experience of population in the search space. The standard PSO algorithm accord to the following formula to update the position and velocity of particles:

\[
V_{i,t+1} = w \cdot V_{i,t} + c_1 \cdot r_1(t) \cdot (P_{i,t} - X_{i,t}) + c_2 \cdot r_2(t) \cdot (P_{g,t} - X_{i,t}) 
\]

\[
X_{i,t+1} = X_{i,t} + V_{i,t+1}
\]

In which, \(X\) and \(V\) present the position and velocity according to the particle \(i\); \(r1\) and \(r2\) are two independent random variables in \([0,1]\); \(P_i\) is the best position of particle \(i\); \(P_g\) is the best position among all the particles; \(c1, c2\) are the acceleration constant, normally set value in \([0, 2]\). Particles in search space continuously determines the speed and direction of movement through its own information \(P_{i,t}\) and population information \(P_{g,t}\).

3 Quantum-behaved Particle Swarm Optimization(QPSO) Algorithm

Based on the basic convergence characteristic of PSO algorithm, inspired by the quantum of physics basic theory, In 2004, Sun etc studied the research results of the particle convergence
behavior by Clerc etc. puts forward a new PSO model from quantum mechanics. This model is based on the DELTA potential as the foundation[7], and consider the particles with quantum behavior, and according to the model proposed a quantum particle swarm optimization algorithm (QPSO). QPSO algorithm changed the search strategy of the PSO algorithm, and evolutionary equation does not need velocity vector, with fewer parameters, and easy to control. The equation is described as follows:

\[ p_{i,j}(t) = (c_1 \cdot P_{i,j}(t) + c_2 \cdot P_{g,j}(t)) / (c_1 + c_2), j = 1, 2, \ldots n \]
\[ c1, c2 \sim U(0, 1) \]

(3)

\[ p_{i,j}(t) = \alpha \cdot P_{i,j}(t) + (1 - \alpha) \cdot P_{g,j}(t), \alpha \sim U(0, 1) \]

(4)

where \( \alpha = c_1 r_1 / (c_1 r_1 + c_2 r_2) \). It can be seen that the local attractor \( P_{i,j} \) is a stochastic attractor of particle \( i \) 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 potential on each dimension, of which the center is the point \( P_{i,j} \). For simplicity, we consider a particle in one-dimensional space, with point \( p \) the center of potential \( S \). Solving the time-dependent Schrödinger equation of one-dimensional potential well, we can get the probability density function \( Q \) and distribution function \( F \):

\[ Q(X_{i,j}(t+1)) = \frac{1}{L_{i,j}(t)} e^{-(2p_{i,j}(t)-X_{i,j}(t+1))^2/L_{i,j}(t)} \]

(5)

\[ F(X_{i,j}(t+1)) = 1 - e^{-(2p_{i,j}(t)-X_{i,j}(t+1))^2/L_{i,j}(t)} \]

(6)

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

\[ X_{i,j}(t+1) = p_{i,j}(t) + \beta \cdot \frac{L_{i,j}(t)}{2} \ln(1/u), u = \text{rand}(0, 1) \]

(7)

where \( u \) is a random number uniformly distributed in \((0, 1)\). To evaluate \( L_{i,j}(t) \), a global point called Mainstream Thought or mean best position of the population is introduced into QPSO. The global point \( m \), denoted as, is defined as the mean of the pbest positions of all particles. That is

\[ m(t) = (m_1(t), m_2(t), \ldots, m_n(t)) = \left( \frac{1}{n} \sum_{i=1}^{n} P_{i,1}(t), \frac{1}{n} \sum_{i=1}^{n} P_{i,2}(t), \ldots, \frac{1}{n} \sum_{i=1}^{n} P_{i,d}(t) \right) \]

(8)

where \( n \) is the population size and \( P_i \) is the pbest position of particle \( i \). The values of \( L_{i,j}(t) \) is determined by

\[ L_{i,j}(t) = 2 \cdot \beta \cdot (m_{j}(t) - X_{i,j}(t)) \cdot \ln(1/u) \]

(9)

and thus the position can be calculated by

\[ X_{i,j}(t+1) = p_{i,j}(t) + \beta \cdot m_{\text{best},j} - X_{i,j}(t) \cdot \ln(1/u), u = \text{rand}(0, 1) \]

(10)
where parameter $\beta$ is called contraction-expansion coefficient, which can be tuned to control the convergence speed of the algorithms. The QPSO algorithm is described as follows.

Initialize population: random $X[i]$ and set $P[i]=X[i]$;

do  
  find out mbest using Eq. (8);
  for i=1 to population size n
    if $f(Xi) < f(Pi)$ then $P[i]=X[i]$;
    r= min($f(P[i])$);
    for j=1 to dimensionality D
      r = rand(0,1);
      $p[i][j]=r*P[i][j]+(1-r)*P[g][j]$;
      $u= rand(0,1)$
      if rand(0,1) > 0.5 $X[i][j]=p[i][j]-b*abs(m[j]-X[i][j])*ln(1/u)$;
      else $X[i][j]=p[i][j]+b*abs(m[j]-X[i][j])*ln(1/u)$;
    endif
  endfor
  endfor

Until termination criterion is met

4 Improved QPSO Algorithm

4.1 Weighted mean of the personal best positions and the stochastic coefficients of QPSO by gaussian distribution

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. So, we use uniform distribution for the weighted mean of the personal best positions as follow:

$$m(t) = (m_1(t), m_2(t),..., m_n(t)) = \frac{1}{n} \sum_{i=1}^{n} r \cdot (P_{i,1}(t), P_{i,2}(t),..., P_{i,d}(t))$$

$$r \sim U(r_{min}, r_{max})$$

Generating random numbers using Gaussian distribution [15]sequences with zero mean and unit variance for the stochastic coefficients of QPSO [12]may provide a good compromise between the probability of having a large number of small amplitudes around the current points (fine tuning) and a small probability of having higher amplitudes, which may allow particles to move away from the current point and escape from local minima. In this work, firstly, random numbers are generated using the absolute value of the Gaussian probability distribution with zero mean and unit variance, i.e., $N(0,2)$. 


4.2 Search strategy with memory

In nature, in the flight of birds, the leading role of birds (it is a ‘bird king’) is more important. In QPSO algorithm, \( m \) can be seen as the ‘bird king’ in the population, which integrated all the thoughts in the population, thus leads other birds flying to the best position with him.

But in the previous QPSO algorithms, the ‘bird king’ has no memory, each of its search integrated the new ideas currently without regard to previous good ideas. Discarding the old ideas which can generally find the best position faster, but at the same time, it also possibly trap into the local optimal solution. To this, we let the ‘bird king’ has memory, it not only summed up the past ideas but also comprehensive the current thought. The ideas presented here can be regarded as the position of the birds flying. The mathematical model is described as follows:

\[
M_{m,n}(0) = \frac{1}{n} \left( \sum_{i=1}^{n} p_{i,1}(0), \sum_{i=1}^{n} p_{i,2}(0), \ldots, \sum_{i=1}^{n} p_{i,d}(0) \right)
\]

\[
M_{m,n}(t) = \frac{1}{n} \left( M_{m,n}(t-1), \sum_{i=1}^{n} p_{i,1}(t), \sum_{i=1}^{n} p_{i,2}(t), \ldots, \sum_{i=1}^{n} p_{i,d}(t) \right)
\]

From the recursive formula can be seen, the thoughts of the “bird king” summarized all the thoughts from the zeroth generation to the \( t \) generation, but with the passage of time, the “bird king” gradually forgot the the past position, and remember the recent events. That is to say, \( M_{best}(t) \), when \( t \to t-1 \) 0, in the \( t \) generation, the “bird king” almost completely forgot the position of \( M_{best}(0) \).

4.3 Singal step search strategy

During the update process, the traditional QPSO algorithm often change the position of all the particle dimension at the same time, and according to the position of the particle, obtain a fitness value, so as to judge the degree of adaptation. This fitness can determine the overall quality of the particle, but can not judge part dimension if it move to the optimal direction.

For example, on three dimensional function Sphere \( f_1 = x_1^2 + x_2^2 + x_3^2 \), the global optimum value is \((0,0,0)\), the initial value of solutions was set \((10, 10, 10)\), the fitness value is 300. Give a random perturbation variable value \((1, 10, 1)\), the initial value is updated to the value \((11, 0, 11)\), and the fitness value is 242. The fitness (242) of the updated solution is less than the initial fitness value(300). So, we can see that in the next iteration, the solution would be to some extent to move \((1, 10, 1)\) direction. At this time, although the second dimension move to the global optimum direction, but the first and third dimension move away from the global optimum. Therefore, for high-dimensional functions, general QPSO algorithm is very difficult to juggle optimize direction of all dimension. In order to solve this problem, the search space of the particles is divided into several low dimensional space, in specific application, according to data dependence analysis, we can decide the segmentation of the data dimension.

Improved QPSO algorithm (MSQPSO) :

Initialize population: random \( X[i] \) and set \( P[i] = X[i] \);

do find out \( M_{mbest} \) using Eq. (11);
for i=1 to population size N
    if f(Xi) < f(Pi) then P[i]=X[i];
    g= min(f(P[i]));
    for j=1 to dimensionality D
        r = rand(0,1);
        p[i][j]=r*P[i][j]+(1-r)*P[g][j];
        nor= normrnd(0,2);
        Mm(i,j)=(Mm(i-1,j)+sum((Rmin+Rmax.rand).p(i,:)))/d;
        if rand(0,1) > 0.5 X[i][j]=p[i][j]-b* abs(Mm[j]-X[i][j])*ln(1/nor);
        else X[i][j]=p[i][j]+b*abs(Mm[j]-X[i][j])*ln(1/nor);
    Endif
    if f(Xi) < f(Pi) then P[i]=X[i];
endfor
endfor
Until termination criterion is met

5 Experiments

<table>
<thead>
<tr>
<th>Problem</th>
<th>name</th>
<th>Search range</th>
<th>Global Optimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>Sphere</td>
<td>[-100,100]</td>
<td>0</td>
</tr>
<tr>
<td>F2</td>
<td>Schwefels Problem 2.22</td>
<td>[-10,10]</td>
<td>0</td>
</tr>
<tr>
<td>F3</td>
<td>Schwefels Problem 1.2</td>
<td>[-100,100]</td>
<td>0</td>
</tr>
<tr>
<td>F4</td>
<td>Schwefels Problem 2.21</td>
<td>[-100,100]</td>
<td>0</td>
</tr>
<tr>
<td>F5</td>
<td>Generalized Rosenbrocks function</td>
<td>[-2.048, 2.048]</td>
<td>0</td>
</tr>
<tr>
<td>F6</td>
<td>Step function</td>
<td>[-100,100]</td>
<td>0</td>
</tr>
<tr>
<td>F7</td>
<td>Quartic function i.e. Noise</td>
<td>[-1.28,1.28]</td>
<td>0</td>
</tr>
<tr>
<td>F8</td>
<td>Generalized Griewank function</td>
<td>[-600,600]</td>
<td>0</td>
</tr>
<tr>
<td>F9</td>
<td>Generalized Rastrigins function</td>
<td>[-5.12,5.12]</td>
<td>0</td>
</tr>
<tr>
<td>F10</td>
<td>Ackleys function</td>
<td>[-32,32]</td>
<td>0</td>
</tr>
</tbody>
</table>

Experiment and simulation is using 10 benchmark functions to find the minimum value (the optimal value is 0). Experiment software platform is MATLAB 7.8 and Windows XP, machine frequency is P4 (2 10GHz), memory is 3G, hard disk is 120G. In this paper, PSO, SFLA, QPSO and WQPSO, MSQPSO were tested, in order to enhance the comparability of our experiment, all of the following public parameter settings are the same. The global iteration number T=
1000, total population number is P=50, variable dimension V=30. Weighted coefficient setting is $r_{min} = 0, r_{max} = 1$. For the SFLA algorithm, cluster number is $m=5$, frog individual number in each cluster is that $n=10$, local iteration number $r=5$. Five algorithms on 10 benchmark functions were run 50 times, and the mean optimal value, standard deviation was cosidered as the evaluation for the algorithm performance.

Table 2 lists the five algorithm to solve the optimization problem, as can be seen from table 2, in the ten benchmark functions the standard PSO algorithm present the worst performance, on the contrary, MSQPSO algorithm showed excellent performance. With regard to the F6 function, MSQPSO algorithm obtain the optimal solution 0. Table 2 proves that the MSQPSO algorithm has quick convergence speed and good robustness, improving the algorithm convergence accuracy and can effectively avoid the algorithm being trapped in local optimal solution.

Figure 1 shows the evolution curve of the 6 function by 5 kinds of algorithm, and each one
Table 2: Comparison of results for the mean and the standard deviation for PSO, SFLA, QRP-SO, WQPSO and MSQPSO

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>F1 Mean</td>
<td>7.2682</td>
<td>4.6363e-005</td>
<td>3.2064e-011</td>
<td>2.7041e-015</td>
<td>3.9237e-088</td>
</tr>
<tr>
<td>SD</td>
<td>6.2093</td>
<td>2.8314e-004</td>
<td>5.5522e-011</td>
<td>5.5206e-015</td>
<td>1.5531e-087</td>
</tr>
<tr>
<td>F2 Mean</td>
<td>0.0728</td>
<td>2.7763e-005</td>
<td>3.9173e-009</td>
<td>3.1742e-017</td>
<td>1.2659e-089</td>
</tr>
<tr>
<td>SD</td>
<td>0.0751</td>
<td>1.9305e-004</td>
<td>2.5134e-008</td>
<td>5.0767e-017</td>
<td>5.7342e-089</td>
</tr>
<tr>
<td>F3 Mean</td>
<td>8.4972e+004</td>
<td>3.8591e+004</td>
<td>2.7413e+004</td>
<td>2.8157e+004</td>
<td>6.4038e+003</td>
</tr>
<tr>
<td>SD</td>
<td>2.9328e+004</td>
<td>1.6400e+004</td>
<td>1.1609e+004</td>
<td>1.1254e+004</td>
<td>3.6590e+003</td>
</tr>
<tr>
<td>F4 Mean</td>
<td>20.5526</td>
<td>0.0322</td>
<td>3.0747</td>
<td>0.4484</td>
<td>5.4316e-019</td>
</tr>
<tr>
<td>SD</td>
<td>3.7755</td>
<td>0.0308</td>
<td>1.3299</td>
<td>0.1884</td>
<td>1.1368e-018</td>
</tr>
<tr>
<td>F5 Mean</td>
<td>41.8042</td>
<td>27.1863</td>
<td>25.7876</td>
<td>25.6391</td>
<td>25.3168</td>
</tr>
<tr>
<td>SD</td>
<td>27.6833</td>
<td>0.5196</td>
<td>0.5617</td>
<td>8.0586</td>
<td>0.1754</td>
</tr>
<tr>
<td>F6 Mean</td>
<td>11.9400</td>
<td>1.7200</td>
<td>0.1800</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>SD</td>
<td>6.6774</td>
<td>3.2642</td>
<td>0.4375</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>F7 Mean</td>
<td>0.1325</td>
<td>0.0118</td>
<td>0.0156</td>
<td>0.0076</td>
<td>4.0247e-005</td>
</tr>
<tr>
<td>SD</td>
<td>0.0453</td>
<td>0.0104</td>
<td>0.0072</td>
<td>0.0028</td>
<td>2.2399e-005</td>
</tr>
<tr>
<td>F8 Mean</td>
<td>1.0292</td>
<td>0.0084</td>
<td>0.0088</td>
<td>0.0070</td>
<td>1.6336e-004</td>
</tr>
<tr>
<td>SD</td>
<td>0.0529</td>
<td>0.0242</td>
<td>0.0117</td>
<td>0.0085</td>
<td>0.0012</td>
</tr>
<tr>
<td>F9 Mean</td>
<td>65.8825</td>
<td>59.7707</td>
<td>22.0165</td>
<td>21.3133</td>
<td>7.3489e-008</td>
</tr>
<tr>
<td>SD</td>
<td>17.8519</td>
<td>24.0783</td>
<td>5.6703</td>
<td>5.8069</td>
<td>5.1965e-007</td>
</tr>
<tr>
<td>F10 Mean</td>
<td>1.9632</td>
<td>1.1479</td>
<td>1.0347e-006</td>
<td>1.0696e-008</td>
<td>4.5830e-015</td>
</tr>
<tr>
<td>SD</td>
<td>0.6116</td>
<td>1.2890</td>
<td>1.1500e-006</td>
<td>1.0172e-008</td>
<td>7.0325e-016</td>
</tr>
</tbody>
</table>

was run for 50 times to calculate the average optimal fitness. From the evolution curve we can see, in the six test function, MSQPSO algorithm converges much faster than other algorithms. In Figure 1 (a), obviously, MSQPSO algorithm converges faster than others and the SFLA algorithm has better performance than WQPSO algorithm. In Figure 1 (b), the MSQPSO algorithm almost converges to the optimal solution after about 500 times iteratives, followed by WQPSO algorithm. For Figure 1 (d), before the 400 times iteration, SFLA algorithm has better performance than MSQPSO algorithm, but after the 400 times iterations, MSQPSO algorithm has more quick convergence speed than SFLA algorithm. Especially, the Step function in figure 1(f) shows that MSQPSO algorithm obtains the best solution at the 110th iteration. Therefore, the proposed algorithm has better convergence performance for single peak function or multi peak function, and is a kind of reliable and effective global optimization algorithm.

6 Conclusions

QPSO algorithm is a kind of swarm intelligence optimization algorithm. In recent years, many scholars has studied the algorithm. The algorithm can converge to the global optimal solution with probability 1, but in fact, at the late period of iterations, it will converge to a local optimum solution. Therefore, this paper increased the performance of the average attractor $m$ with a memory function, improved the algorithm’s searching ability; and to some extent, single step
search strategy solve the high scale of data optimization problem. Experimental results show that, the improved algorithm has strong searching ability, quick convergence speed and good stability. The function of memory can effectively avoid premature.

References