Quantum-Behaved Particle Swarm Optimization with Chaotic Search

Kaiqiao YANG∗[a], Nonmember and Hirosato NOMURA†, Member

SUMMARY The chaotic search is introduced into Quantum-behaved Particle Swarm Optimization (QPSO) to increase the diversity of the swarm in the latter period of the search, so as to help the system escape from local optima. Taking full advantages of the characteristics of ergodicity and randomicity of chaotic variables, the chaotic search is carried out in the neighborhoods of the particles which are trapped into local optima. The experimental results on test functions show that QPSO with chaotic search outperforms the Particle Swarm Optimization (PSO) and QPSO. 

key words: PSO, QPSO, chaotic search

1. Introduction

This paper proposes a new hybrid search algorithm by combining the chaotic search with the Quantum-behaved Particle Swarm Optimization (QPSO) during the latter period of the search to improve the performance of the QPSO algorithm. 

Particle swarm optimization (PSO), originally proposed by J. Kennedy and R. Eberhart [1], is a new global search technique. The underlying motivation for the development of PSO algorithm was the social behavior of animals such as bird flocking, fish schooling and swarm theory. Like genetic algorithm (GA), PSO is a population-based random search technique and is comparable in performance with GA and other evolutionary algorithms such as simulated annealing (SA) in many practical applications, particularly in non-linear optimization problems [2], [3]. PSO has become an important optimization tool since it has fewer parameters, simplicity in software programming and the relatively fast convergence rate. However, as demonstrated by Van den Bergh, it is not a global convergent algorithm [4].

Quantum-behaved Particle Swarm Optimization (QPSO) [5], [6], which was proposed by Sun, is a novel algorithm based on the PSO and quantum model. QPSO algorithm is depicted only with the position vector without velocity vector, which is a simpler algorithm. Furthermore the results show that QPSO performs better than standard PSO on several benchmark test functions and is a promising algorithm due to its global convergence guaranteed characteristic.

Although the QPSO algorithm is a viable alternative for the optimization problems, the particles of the swarm in QPSO still approach each other quickly and the loss of the diversity is inevitable. Therefore a lot of revised versions based on QPSO have emerged since the QPSO algorithm was proposed. In [7], the mechanism of Cauchy mutation is proposed to increase the diversity of the swarm in QPSO and improve the ability of global search. Furthermore in [8], an Adaptive Cauchy Mutation Operator based on annealing is further adopted to increase the self-adaptive capability of the improved algorithm. From the point of view of principle of immune system, Liu [9], [10] introduced the immune operator based on immune memory and immune regulation, and the immune operator based on vaccination into the QPSO algorithm to increase the convergent speed by using the characteristic of the problem to guide the search process. The QPSO with immune operator not only has a better capability of global search, but also guarantees the diversity and can find a more precise solution rapidly. In [11], the mechanism of Simulated Annealing is introduced into QPSO as the selection mechanism of QPSO. This hybrid algorithm could effectively employ both the ability to jump out of the local minima in Simulated Annealing and the capability of searching the global optimum in QPSO algorithm.

Chaos is a universal nonlinear phenomenon. Since Lorenz [12] found the chaos attractor in 1963, chaos theory is studied and applied in many scientific disciplines such as mathematics and computer science. Chaos behavior is complex and like random, but is orderly in the sense of being deterministic. Chaotic system usually has well defined statistics. It can traverse all the states in definite area without repetition, and is sensitive to initial condition [13]. The ergodicity and rich dynamics of the chaotic system makes the chaotic search escape more easily from the local optima than random search, and has strong search ability [14].

Considering the loss of diversity in QPSO algorithm and the ergodic characteristic of the chaotic search, this paper tries to introduce the chaotic search into QPSO to employ the strengths of both algorithms, so as to increase the global search ability and escape from the local minima.

The rest of this paper is organized as follows. In Sect. 2, PSO is introduced and QPSO is detailed described and analyzed. Section 3 shows how to introduce the chaotic search into the QPSO. Section 4 shows the experimental settings and comparable results of QPSO with chaotic search. Finally, some concluding remarks are given in Sect. 5.
2. PSO and QPSO

2.1 Overview of Particle Swarm Optimization

In the standard PSO model, considering the global optimization problem \( P \)
\[
\min\{f(X) : X \in \Omega \subset R^D\}, \quad f : \Omega \subset R^D \rightarrow R^1
\]
the set of some candidate solutions of problem \( P \) is called swarm, and each individual (candidate solution) in the swarm is treated as a volume-less particle in the \( D \)-dimensional space, with the position and velocity of the \( i \)th particle represented as \( X_i = (x_{i1}, x_{i2}, \ldots, x_{iD}) \) and \( V_i = (v_{i1}, v_{i2}, \ldots, v_{iD}) \). The particles move according to the following equation:

\[
v_{id}(t+1) = \omega * v_{id}(t) + c_1 * r_1 * (p_{id} - x_{id}(t)) + c_2 * r_2 * (g_{id} - x_{id}(t)) \tag{1}
\]

\[
x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \tag{2}
\]

where \( c_1 \) and \( c_2 \) are acceleration constants that say how much the particle is directed towards good positions, they represent a cognitive and a social component, respectively, in that they affect how much the particle’s personal best and the global best (respectively) influence its movement. \( r_1 \) and \( r_2 \) are two uniform random numbers in the range of \([0, 1]\). Vector \( P_i = (p_{i1}, p_{i2}, \ldots, p_{iD}) \) is the best previous position (the position giving the best fitness value by now) of particle \( i \) called \( p_{best} \), and vector \( G = (g_{i1}, g_{i2}, \ldots, g_{iD}) \) is the best among all the \( p_{best} \) in the swarm and called \( g_{best} \), \( x_{id}, v_{id}, p_{id} \) are the \( d \)th dimension of vectors of \( X_i, V_i, P_i \). Parameter \( \omega \) is the inertia weight introduced to accelerate the convergence speed of the PSO [15]. Usually parameter \( \omega \) decreases linearly according to the following formula:

\[
\omega = \omega_{max} - \text{iter} * (\omega_{max} - \omega_{min})/\text{iteration} \tag{3}
\]

where \( \omega_{max} \) and \( \omega_{min} \) are the maximum and minimum values of \( \omega \) respectively, which usually take values 0.9 and 0.4, \( \text{iteration} \) is the number of evolutions, \( \text{iter} \) is currently iteration of the evolution.

2.2 Concept of Quantum-Behaved PSO

Trajectory analyses [16] demonstrated that, to guarantee the convergence of the PSO algorithm, each particle must converge to its local attractor \( p = (p_1, p_2, \ldots, p_D) \), of which the coordinates are:

\[
p_d = (c_1 p_{id} + c_2 g_{id})/(c_1 + c_2) \tag{4.a}
\]
or

\[
p_d = \varphi * p_{id} + (1 - \varphi) * g_{id}, \quad 0 < \varphi < 1 \tag{4.b}
\]

From (4.a) and (4.b), \( p \) is a point between \( p_{best} \) and \( g_{best} \).

In quantum model of PSO, each particle has quantum behavior, and we can only learn the probability of the particles’ appearing in position \( x \) from the probability density function \( |\Psi(x,t)|^2 \), the form of which depends on the potential field the particle lies in. Assume that there is one-dimensional Delta potential well on each dimension with point \( p \) the center of potential. Solving the Schrödinger equation, we can get the normalized probability density function \( Q \) as follows:

\[
Q(y) = |\Psi(y)|^2 = \frac{1}{L} e^{-2y^2/L} \tag{5}
\]

where \( L \) is the most important variable, which determines search scope of each particle. Employing Monte Carlo method, and for \( y = x - p \), we can obtain the position of the particle

\[
x = p \pm (L/2) * \ln(1/u), \quad u = \text{rand}(0, 1) \tag{6}
\]

where \( u \) is a random number distributed uniformly on \([0, 1]\). The value of \( L \) is evaluated as \( L(t) = 2*\beta*|m_{best} - x(t)| \) [17], and

\[
m_{best} = \frac{1}{M} \sum_{i=1}^{M} P_i
\]

\[
= \left( \frac{1}{M} \sum_{i=1}^{M} p_{i1}, \frac{1}{M} \sum_{i=1}^{M} p_{i2}, \ldots, \frac{1}{M} \sum_{i=1}^{M} p_{iD} \right) \tag{7}
\]

where \( m_{best} \) (Mean Best Position) is defined as the mean value of all particles’ best position, and \( M \) is the population size, \( \beta \), called Contraction Expansion Coefficient, is the only parameter in QPSO algorithm. Therefore, Eq. (6) can be written as:

\[
x(t+1) = p \pm \beta * |m_{best} - x(t)| * \ln(1/u) \tag{8}
\]

2.3 Loss of Diversity in QPSO Algorithm

In QPSO algorithm, although the search space of an individual particle at iteration each is the whole feasible solution space of the problem, the loss of diversity of the population is also inevitable due to the collectiveness, as PSO and other population-based evolutionary algorithms. According to the simulation results for the benchmark functions (see Table 1) using QPSO algorithm, we can investigate that during the latter search period, the particles are clustered together and its search area is so limited that the whole swarm is prone to be trapped in a local minimum.

In order to see the relative improvement of the fitness of each particle of the swarm during the evolution process, we use the following equation to calculate the improvement for each particle in iteration each.

\[
\Delta f_i = (f_i - f_{r_{best}})/f_i \tag{9}
\]

where \( f_i \) is the fitness of the \( i \)th particle in current iteration,
and $f_{\text{best}}$ is the best fitness by present of particle $i$. If $\Delta f$ is less than zero, it shows that the particle is moving towards the optimum.

Figure 1 gives the plots of evolution processes of the particles for benchmark functions Sphere, Rosenbrock, Rastrigrin and Griewank, with population of the swarm =20, the function dimension=10 and the iteration=1000. In the figure the combination of different color and different line style denotes different particle. From the figure we can see that during the latter period of the search, the values of $\Delta f$ of all particles will tend to zero. This means that the diversity of the swarm is lost suddenly in the later period.

Seeing that the QPSO algorithm still has defects, the mechanism of chaotic search is proposed to increase the capability of jumping out of the local optimum.

3. The QPSO with Chaotic Search

3.1 Chaotic Search

In this paper, we use the Logistic map to generate the chaotic variables [18]. The logistic map is a polynomial mapping, often cited as an archetypal example of how complex, chaotic behavior can arise from very simple non-linear dynamical equations. The logistic model was originally introduced as a demographic model by Pierre François Verhulst. Mathematically this can be written as:

$$z_{j+1} = rz_j(1-z_j), \quad z_j \in [0, 1], \quad j = 0, 1, 2, \ldots$$

where $z_j$ is a number between zero and one, and represents the population at year $j$, $r$ is a positive number, and represents a combined rate for reproduction and starvation, $r$ is also called chaotic attractor, for chaotic behavior is dependent on $r$. Most values of $r$ beyond 3.57 (approximately) and less than 4 exhibit chaotic behaviors.

For chaotic search, first we randomly generate a $D$-dimensional vector $Z$ with each value between 0 and 1 as the initials of chaotic variables. Vector $Z$ iterates according to Eq. (10) and its members are always kept in $[0, 1]$. Then in each iteration, we use $\Delta f$ in Eq. (9) to see whether the particle will be trapped into a local optimum.

If $0 \leq \Delta f < \delta$ (threshold) is satisfied $N_c$ times continually in the iteration, the chaotic search is carried out on the area with the particle’s current position $x_{id}$ the center and the $x_{\text{max}}$ the radius according to the following equation.

$$x_{id}(t+1) = x_{id}(t) + x_{\text{max}}(2z_d - 1)$$

where $z_d$ is the $d$th dimension of vector $Z$, $x_{\text{max}}$ is the limit to the search space.

In the algorithm we use a vector of flags to indicate whether the chaotic search will be used for the corresponding particle. If the flag is set to true then the chaotic search will be carried out in the next iteration for the corresponding particle.

3.2 QPSO with Chaotic Search Algorithm Pseudo Code

The Quantum-behaved PSO with chaotic search is described in Fig. 2.

In Fig. 2, $\text{clamp xid}$ means that if the value of $x_{id}$ exceeds the range of $[-x_{\text{max}}, x_{\text{max}}]$, then the $x_{id}$ will be clamped to a random value in the range. Experiments show that it will be better than that $x_{id}$ only be clamped to the boundary of the range.
3.3 Convergency of QPSO with Chaotic Search Algorithm

**Hypothesis 1:** (i) The domain \( \Omega \) of problem \( P \) is the limited close area in \( \mathbb{R}^n \), (ii) Object function \( f(x) \) is the continuous function defined on area \( \Omega \).

From the hypothesis, it is easy to know that the best solution set of \( f \) in domain \( \Omega \) is not null.

**Definition 1:** For the arbitrary random initial status \( F_0 \), if \( \lim_{t \to \infty} P\{\|f(F(t))\| < 1 | F(0) = F_0\} = 1 \), then it is called that the algorithm converges to the global best solution with probability 1, or for \( \forall r > 0 \), if the random sequence \( \{\xi_n\} \) has \( P\left( \bigcap_{n=1}^{\infty} \bigcup_{k \geq n} [\xi_k - \xi| \geq r] = 0 \right) \), then it is called that the random sequence \( \{\xi_n\} \) converges to random variable \( \xi \) with probability 1.

**Theorem 1:** For the boundary constraint optimization problem, the algorithm of QPSO with chaotic search is local convergent, and converges to the global best solution with probability 1.

Proof: The process to prove that the proposed algorithm is local convergent is similar to [19], next we prove that the algorithm converges to the global best solution with probability 1.

From the Hypothesis 1, we know that the best solution set of the boundary constraint optimization problem exists. For the proposed algorithm, the Logistic sequence mutation is added to the QPSO algorithm, and it has been proved in [20] that the QPSO is local convergent, and converges to the global best solution with probability 1. Therefore in the following we only give the proof that the Logistic sequence mutation is local convergent and converges with probability 1.

Logistic sequence mutation: For \( \forall r > 0 \), the neighborhood of global best position \( x^* \) is labeled as \( B(x^*, r) = \{x \in \Omega | \|f(x) - f^*\| < r\} \). Let the random event sequence \( A_t = \{F(t) \cap B(x^*, r) \neq \emptyset\} \) which means on the \( t \)th iteration the solution comes into \( B(x^*, r) \). For a given \( r \), from the ergodicity of the Logistic sequence, it can be known that if event \( A_1 \) happens, event \( A_2 \) also has to happen, by analogy, we have \( A_1 \subseteq A_2 \subseteq \cdots \subseteq A_t \subseteq \cdots \), therefore we have \( P(A_1) \leq P(A_2) \leq \cdots \leq P(A_t) \leq \cdots \), and also because of \( 0 \leq P(A_t) \leq 1 \), the limit \( \lim P(A_t) \) exists.

Furthermore we suppose that the random variable sequence

\[
\delta_t = \begin{cases} 1 & \text{tth iteration comes into } B(x^*, r) \\ 0 & \text{otherwise} \end{cases}
\]

\( t = 1, 2, \ldots \), then we have \( A_t = \{\delta_t = 1\} \).

Let \( P(A_t) = P(\delta_t = 1) = p_t, P(\delta_t = 0) = 1 - p_t, \) and \( S_t = \frac{1}{t} \sum_{i=1}^{t} \delta_i, t = 1, 2, \ldots \), then \( E(S_t) = \frac{1}{t} \sum_{i=1}^{t} p_i \), \( D^2(S_t) = \frac{1}{t^2} \sum_{i=1}^{t} D^2(\delta_i) = \frac{1}{t^2} \sum_{i=1}^{t} p_i(1 - p_i) \leq \frac{1}{4t^2} \), where \( E(S_t), D(S_t) \) is the mathematic expectation and deviation of the sequence \( S_t \) respectively.

According to Chebyshev inequality, there is \( P\{S_t - E(S_t) \geq r\} \leq \frac{D^2(S_t)}{r^2} \leq \frac{1}{4tr^2} \), therefore \( \lim_{t \to \infty} P\{S_t - E(S_t) \geq r\} = 0 \).

Also because \( \delta_t = t \cdot S_t = (t-1) \cdot S_{t-1} \), there is \( \lim P\{\delta_t - E(\delta_t) \geq r\} = 0 \).

The above equation shows the random variable sequence \( \{\delta_t\} \) converges with probability 1, therefore the random event sequence \( \{A_t\} \) also converges with probability 1, that is \( \lim_{t \to \infty} \bigcup_{k \geq t} A_k = 0 \). Therefore the theorem is proved.

4. Experimental Settings and Results

4.1 Benchmark Functions and Experimental Settings

To test the performance of QPSO with chaotic search, four representative benchmark functions are used here for comparison with SPSO and QPSO. Table 1 gives the test functions: Sphere, Rosenbrock, Rastrigin and Griewank, mathematical expression, its initial range, minimum function value and the corresponding limits to the search space.

According to the characteristic of the function, these four functions can be classified into two groups. Function Sphere and Rosenbrock belong to unimodal functions which have only one minimum, and function Rastrigin and Griewank belong to multimodal functions which have many local minima.

As in [2], for each function, three different dimension sizes, 10, 20 and 30 are tested. The corresponding maximum iterations are 1000, 1500 and 2000 respectively. The

<table>
<thead>
<tr>
<th>fun</th>
<th>mathematic expression</th>
<th>initial range</th>
<th>( f_{\text{min}} )</th>
<th>( X_{\text{max}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sphere</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 )</td>
<td>(50, 100)</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Rosenbrock</td>
<td>( f(x) = \sum_{i=1}^{n} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2] )</td>
<td>(15, 30)</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>Rastrigin</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 - 10 \cos(2\pi x_i) + 10 )</td>
<td>(2.56, 5.12)</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>Griewank</td>
<td>( f(x) = \sum_{i=1}^{n} x_i^2 - \frac{x_i - 100}{\sqrt{v_i}} )</td>
<td>(300, 600)</td>
<td>0</td>
<td>600</td>
</tr>
</tbody>
</table>
population size is set to 20, 40 and 80. A total of 50 runs for each experimental setting are conducted.

In our experiment, for comparison with the results of QPSO with Cauchy Mutation [7] and the results of QPSO with Adaptive Mutation [8], the parameters for PSO and QPSO are selected from [7] and [8]. That is, for standard PSO algorithm, inertia weight \( w \) decreases linearly from 0.9 to 0.4, acceleration constants \( c_1 \) and \( c_2 \) are both set to 2, and \( v_{max} = x_{max} \), for QPSO algorithm, the parameter \( \beta \) decreases linearly from 1.0 to 0.5.

For QPSO with chaotic search algorithm, besides the parameter \( \beta \) which was set the same as the QPSO algorithm, there are other three parameters, that is, the threshold of \( \Delta f \), the times \( N_c \) and the coefficient \( r \) of Logistic map in Eq. (10). It is very difficult to determine the values of the three parameters and we only use empirical method to determine the values, that is, \( r = 3.618 \), \( N_c = 3 \), \( \delta = 0.0001 \).

### 4.2 Experimental Results

The mean values and standard deviations for 50 runs of each test function are recorded in Table 2 to Table 5 and the corresponding bar graphs for the mean values of each function are shown in Fig. 3 to Fig. 6. For comparison, we also list the results of the QPSO with Cauchy Mutation [7] and the results of QPSO with Adaptive Mutation [8] in the following tables for function Rosenbrock, Rastrigrin and Griewank (There are no data for function Sphere listed in [7] or [8] for the reason that there is no improvement for function Sphere with new methods). In [7] or [8], the mutation is performed on variables \( m_{best} \) and \( g_{best} \) independently. Therefore there are two results for both algorithms. For saving the space, in the following tables we only list the smallest values of the four results of the algorithm with mutations.

In the following, SPSO denotes standard PSO, QPSOC denotes the QPSO with Chaotic search, and QPSOM denotes the QPSO with Mutation operator. Because the mean values of function Sphere differ greatly for the three algorithms, for convenience, we use negative logarithmic values of the mean values to draw the bar graph. Therefore in Fig. 3, the meaning of the height of the bar is different from the other bar graphs, the higher the bar is, the smaller the corresponding value is. The numerical results and the bar graphs show that the QPSO with Chaotic Search outperforms QPSO and SPSO especially on Sphere function and has comparable performance with QPSOM.

Figures 7 to 10 give the evolution curves of mean values (for 50 runs) of the four benchmark functions per it-

<table>
<thead>
<tr>
<th>Table 2: Sphere function.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSO</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 3: Rosenbrock function.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSO</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 4: Rastrigrin function.</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPSO</td>
</tr>
<tr>
<td>P</td>
</tr>
<tr>
<td>---------------------------</td>
</tr>
<tr>
<td>10</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
<tr>
<td>30</td>
</tr>
<tr>
<td>40</td>
</tr>
<tr>
<td>20</td>
</tr>
</tbody>
</table>

Table 5: Griewank function.
Tables and figures:

Table 5: Griewank function.

<table>
<thead>
<tr>
<th>P</th>
<th>D</th>
<th>G</th>
<th>Mean</th>
<th>St.Dev</th>
<th>Mean</th>
<th>St.Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>1000</td>
<td>10000</td>
<td>0.09523</td>
<td>0.04495</td>
<td>0.0786</td>
<td>0.07639</td>
</tr>
<tr>
<td>20</td>
<td>1500</td>
<td>15000</td>
<td>0.02890</td>
<td>0.02698</td>
<td>0.02532</td>
<td>0.02503</td>
</tr>
<tr>
<td>30</td>
<td>2000</td>
<td>20000</td>
<td>0.01719</td>
<td>0.02390</td>
<td>0.008578</td>
<td>0.01224</td>
</tr>
<tr>
<td>40</td>
<td>1000</td>
<td>10000</td>
<td>0.09230</td>
<td>0.04427</td>
<td>0.06337</td>
<td>0.05807</td>
</tr>
<tr>
<td>40</td>
<td>1500</td>
<td>15000</td>
<td>0.02983</td>
<td>0.02712</td>
<td>0.02191</td>
<td>0.01786</td>
</tr>
<tr>
<td>30</td>
<td>2000</td>
<td>20000</td>
<td>0.008767</td>
<td>0.01183</td>
<td>0.01035</td>
<td>0.01378</td>
</tr>
<tr>
<td>80</td>
<td>1000</td>
<td>10000</td>
<td>0.07326</td>
<td>0.02702</td>
<td>0.04004</td>
<td>0.03524</td>
</tr>
<tr>
<td>80</td>
<td>1500</td>
<td>15000</td>
<td>0.02610</td>
<td>0.02309</td>
<td>0.01483</td>
<td>0.01953</td>
</tr>
<tr>
<td>30</td>
<td>2000</td>
<td>20000</td>
<td>0.01401</td>
<td>0.01768</td>
<td>0.01124</td>
<td>0.01245</td>
</tr>
</tbody>
</table>

QPSO and QPSOC:

<table>
<thead>
<tr>
<th>P</th>
<th>D</th>
<th>G</th>
<th>Mean</th>
<th>St.Dev</th>
<th>Mean</th>
<th>St.Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>1000</td>
<td>1000</td>
<td>0.06063</td>
<td>0.0473</td>
<td>0.0627</td>
<td>0.0557</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1500</td>
<td>0.01960</td>
<td>0.01922</td>
<td>0.0189</td>
<td>0.0175</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>2000</td>
<td>0.008564</td>
<td>0.0110</td>
<td>0.00999</td>
<td>0.0107</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>0.04886</td>
<td>0.0438</td>
<td>0.052</td>
<td>0.0401</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1500</td>
<td>0.01662</td>
<td>0.01467</td>
<td>0.0171</td>
<td>0.0177</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>2000</td>
<td>0.007517</td>
<td>0.01161</td>
<td>0.0092</td>
<td>0.0118</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td>0.03937</td>
<td>0.02213</td>
<td>0.0419</td>
<td>0.0262</td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1500</td>
<td>0.01216</td>
<td>0.0103</td>
<td>0.0123</td>
<td>0.0131</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>2000</td>
<td>0.008654</td>
<td>0.01266</td>
<td>0.0069</td>
<td>0.0094</td>
<td></td>
</tr>
</tbody>
</table>

Figures:

Fig. 3: $-\log_{10}$ values of mean values of Sphere function.

Fig. 4: Mean values of Rosenbrock function.

Fig. 5: Mean values of Rastrigrin function.

Fig. 6: Mean values of Griewank function.

Fig. 7: Mean values of Sphere per iteration with population = 20 and dimension = 10.

Discussion:

eration respectively with different population size and dimension of functions. For convenience, we use logarithmic values for the vertical axis. From the figures, we can see that the curve of QPSO with Chaotic Search almost overlaps with the curve of QPSO, and both converged quickly than traditional PSO. This is because the QPSO algorithm has a faster convergence rate than PSO, and in the earlier search period the chaotic search has not been carried out. However during the later period, we can see that QPSO with chaotic search can escape from the local optima and works better than QPSO.

For comparison, let us consider the same population of the swarm and dimension of the function. From Fig. 7, we can see that approximately after iteration 600 the curve of QPSO is becoming gradual and the chaotic search begins to
be carried out. This agrees with the situation in Fig. 1 which shows approximately after the same iterations the diversity of the swarm was lost in the QPSO algorithm.

From the evolution curves, we can also see when entering the later period the slope of curves of QPSO method becomes gradual, even though we extend the iteration times, the function values will not become smaller further. However for most test situations, at the end point of iterations the curve of QPSO with chaotic search method is sloping, therefore if we extend the iteration times, the result can be expected to become better.

Although the chaotic search can indeed help the particles in QPSO to escape from the local optima during the later period, the extent of improvements of results is not very obvious. Maybe this is because when we use Logistic sequence to control the random drift of the particles, some particles can be drifted too extremely and move to a point which is far from the gbest and the other particles. The whole swarm becomes scattered for a temporary. Furthermore we can find the performance of QPSO with chaotic search method and the one of QPSO with Cauchy mutation operator method are comparable. This means if we only use random sequence to control the drift of the particles to avoid being trapped into local optima, the extents of improvements of different improved methods for QPSO are almost the same.

5. Concluding Remarks

QPSO is a promising algorithm for the optimization problems. But in the later period in order to guarantee the convergency of the algorithm, the particles collect quickly and the performance of the algorithm is not very satisfactory. Therefore we must find other search mechanisms to improve the performance of QPSO.

In this paper, the chaotic search was introduced to improve the behavior of Quantum-behaved particle swarm optimization. According to the dynamics of chaotic system, chaotic searches are carried out to help the QPSO algorithm avoid being trapped into the local minima in the latter period of the search. Comparison with the original PSO and QPSO without chaotic search, the experimental results show that chaotic search mechanism provides an extended improvement on global search ability.

The selection of the parameters for the QPSO with Chaotic Search algorithm itself is an optimization problem. How to determine the values of the parameters is worthy of studying for the future work.

Simply using random sequence to control the drift of the particles can not get satisfactory improvements for the QPSO method, for the future work we should search other mechanisms to improve the performance of QPSO.

References


