An improved QPSO algorithm base on social learning and Lévy flights

XiaoPing Yuan\textsuperscript{a}, Peng Jin\textsuperscript{a} and Guopeng Zhou\textsuperscript{b}

\textsuperscript{a}School of Information and Electric Engineering, China University of Mining and Technology, Xuzhou, People’s Republic of China; \textsuperscript{b}Institute of engineering and technology, Hubei University of Science and Technology, Xianning, People’s Republic of China

\textbf{ABSTRACT}

The centre of the potential well of the quantum-behaviour particle swarm optimization (QPSO) is restricted to the super rectangle which is made up of the local optimal position and the global optimal position. The information sharing mechanism among particles is single, and the algorithm has the problems of premature convergence and low optimization efficiency. To solve this problem, an improved QPSO algorithm is proposed, which integrates social learning and Lévy flights (LSL-QPSO). Firstly, the social learning strategy is used to update the non-optimal particle and improve the global search ability. Then, the Lévy flights strategy is introduced to overcome the shortcoming of the low efficiency of the optimal particle in the social learning mechanism, and further improve the convergence accuracy and search efficiency of the algorithm. Finally, four typical Benchmark functions are tested. The results show that the convergence accuracy, search efficiency and universality of the LSL-QPSO algorithm are ahead of QPSO and other similar QPSO improved algorithms.

\textbf{KEYWORDS}

Quantum-behaved particle swarm optimization (QPSO) algorithm; potential well centre; social learning; Lévy flights

1. Introduction

The particle swarm optimization (PSO) is a random, parallel and distributed search strategy proposed based on foraging behaviour of colonial organisms (Kennedy & Eberhart, 2002). Due to its simple, less variable parameters and easy implementation, it is widely used in industrial, social, military and management fields (Collotta, Pau, & Maniscalco, 2017; Ghasemi, Aghaei, & Hadipour, 2017; Rahmat-Samii, Gies, & Hadipour, 2017; Tang et al., 2017).

The PSO algorithm is easy to be premature convergence, which leads to the failure of the algorithm to reliably converge to the global optimal position, and the algorithm has the problems of fast convergence in early stage, slow convergence in the later stage, and randomness in parameter selection (Couceiro & Ghamisi, 2016). In recent years, in view of the above problems, researchers have improved the convergence rate and convergence accuracy of the PSO algorithm. In 2004, Sun et al. were inspired by particles in quantum space and proposed the QPSO algorithm (Sun, Feng, & Xu, 2004). The position of each particle in quantum space is equivalent to a feasible solution, its position coordinate is determined by the wave function, and the state follows the superposition principle. It has a strong randomness and a high degree of intelligence. In order to ensure the convergence of the algorithm, Sun and others used the potential field to bind the particles to maintain the group aggregation state, so that the particle can search for any position in the space in a certain probability, but will not run to infinity, and finally collapse to the lowest point of potential energy. Due to the global search characteristic of QPSO algorithm, after it has been put forward for more than ten years, it has been recognized by many scholars at home and abroad (Li, Jiao, Shang, & Stolkin, 2015; Singh & Mahapatra, 2016; Wang, Gandomi, Alavi, & Deb, 2016).

As one of the improvement directions, the update mode of potential well centre has been widely studied in recent years. In 2004, when Sun created the QPSO algorithm, the QPSO algorithm with diversity retention ability (DCQPSO) was proposed (Sun et al., 2004). The algorithm took the average particle distance as the basis of diversity control. When the particle distance reached the lower threshold, the disturbance was artificially increased, making the potential well centre offset and increasing the later diversity of the algorithm.

In 2014, Wang proposed the double core disturbance QPSO algorithm (BCD-QPSO) (Wang, Jian-Hua, Chen, & Liu, 2014), which performed adaptive Cauchy mutation on the potential well centre and the population centre of gravity, giving full play to the synergistic guidance ability of the two in the late evolution. In 2015, Wu et al. constructed subgroups with different potential well centres, and proposed a double group interactive QPSO (DIR-QPSO) algorithm based on random evaluation (Wu, Yan, &...
Chen, 2015), which slowed down the diversity attenuation and improved the global search capability through subpopulation collaboration. In 2016, Li et al proposed an improved QPSO algorithm (CQPSO) (Li, Xuan, & Wang, 2016). In the process of particle iterative updating, it adjusts the update mode of the potential well center and introduces the adaptive adjustment mechanism to avoid the particles falling into the local optimum.

From the above literature, one can see that improving the update mode of potential well center can effectively optimize the algorithm performance, and improve the convergence rate and accuracy. Therefore, an improved QPSO algorithm is proposed, which integrates social learning and Lévy flights. Social learning mechanism can make full use of the group information to update the potential well center and improve the global searching ability of the particles. Lévy flights mechanism is introduced, act on the global optimal particle, so that it can get rid of the local optimal constraint and further improve the convergence rate and precision of the algorithm. Finally, the performance of the algorithm is verified by typical test functions. The results show that the proposed algorithm has faster search speed and better convergence accuracy.

2. Quantum-Behaved particle swarm optimization algorithm

QPSO algorithm is a global optimization algorithm based on PSO algorithm. The algorithm gets rid of the limitation of the velocity-displacement orbit model (Frans, 2006), and uses Monte Carlo algorithm to determine the location after the updating of the particle. The algorithm is more intelligent, and its evolution equation is shown in formula (1).

\[
X(t + 1) = p_{ij}(t) \pm \frac{L_{ij}(t)}{2} \ln \left(\frac{1}{u(t)}\right), \quad u(t) \in U(0, 1). \tag{1}
\]

In the formula (1), \(X(t + 1)\) is the location after the particle is updated, \(p_{ij}(t)\) is the potential well center, which determines the direction of particle optimization, \(L_{ij}(t)\) is the length of potential well, which directly restricts the search step length of the particle, and \(u(t)\) is the random distribution on \((0, 1)\). Clerc studies the behaviour of particle convergence in the PSO algorithm. The study shows that in order to ensure the convergence of the algorithm, each particle \(i\) must be convergent to the centre of each potential well \(p_{ij}\) (Clerc & Kennedy, 2002), in which the potential well center coordinate is like the formula (2).

\[
p_{ij}(t) = \frac{c_1 r_{1j}(t) p_{ij}(t) + c_2 r_{2j}(t) G_j(t)}{c_1 r_{1j}(t) + c_2 r_{2j}(t)}, \quad 1 \leq j \leq N. \tag{2}
\]

In the formula (2), \(c_1\) and \(c_2\) are respectively the local learning factor and global learning factor of particles, \(r_{1j}(t)\) and \(r_{2j}(t)\) are the random distribution on \((0, 1)\), \(P_{ij}(t)\) is the local optimal position of the \(j\)th dimension of the particle \(i\), \(G_j(t)\) is the global optimal position of the \(j\)th dimension of the particle \(i\).

To ensure the convergence of the algorithm, the formula (1) must be satisfied:

\[
\lim_{t \to \infty} L_{ij}(t) = 0. \tag{3}
\]

Therefore, how to control \(L_{ij}(t)\) becomes the key to ensure algorithm convergence. The typical \(L_{ij}(t)\) control mode is to use the distance between particles and the average optimal position of the population to regulate and control. The average optimal position coordinate is \(C(t)\), and its coordinate can be obtained by formula (4):

\[
C(t) = (C_1(t), C_2(t), \ldots, C_D(t)) = \frac{1}{N} \sum_{i=1}^{N} p_i(t) = \left(\frac{1}{N} \sum_{i=1}^{N} p_{i1}(t), \frac{1}{N} \sum_{i=1}^{N} p_{i2}(t), \ldots, \frac{1}{N} \sum_{i=1}^{N} p_{iD}(t)\right). \tag{4}
\]

Among them, \(D\) is the particle dimension, and \(N\) is the population size. At this time, the particle potential well length can be expressed as formula (5):

\[
L_{ij}(t) = 2 \cdot \alpha \cdot |C_j(t) - X_{ij}(t)|. \tag{5}
\]

Formula (5) is put into the formula (1), and the final evolutionary equation is shown in formula (6):

\[
X_{ij}(t + 1) = p_{ij}(t) \pm \alpha |C_j(t) - X_{ij}(t)| \cdot \ln\left(\frac{1}{u_{ij}(t)}\right). \tag{6}
\]

3. QPSO algorithm integrating social learning and Lévy flights

3.1. QPSO algorithm with social learning

The social learning widely exists in the biological world and plays an important role in the learning of biological behaviour. It is a performance process by which the biological individual acquires relevant information and changes its behaviour through observing and imitating other individuals’ behaviours or behaviours results (Galef & Laland, 2005). Compared with individuals, society has greater potential for storing knowledge, applying knowledge and innovating knowledge. Cheng et al introduced these characteristics into the PSO algorithm (Cheng, 2015), and proposed the PSO algorithm with social learning ability (Social Learning PSO, SL-PSO). Unlike the basic PSO algorithm, the particle can learn from any individual better than itself in the SL-PSO algorithm, strengthen the information exchange between particles, and improve...
the diversity of the population. When solving multiple maximum problems, it is difficult for particles to fall into the local optimal position.

In order to solve the problems that the information exchange between particles is insufficient and the diversity is seriously attenuated in the update process of potential well centre of QPSO algorithm, this paper introduces the social learning idea into the QPSO algorithm, which is used to adjust the mode of potential well centre update and establish the QPSO algorithm with the ability of social learning (Social Learning QPSO, SL-QPSO). The algorithm flow is as follows:

Step 1: Population initialization and parameter initialization. Generate an initial population with a dimension of $D$, and a scale of $N$. Determine the social impact factor $\varepsilon$. The parameters are determined by formulas (7) and (8):

$$N = M + \lfloor D/10 \rfloor.$$  \hspace{1cm} (7)

The performance of PSO algorithm is sensitive to the optimization problem dimension, so in order to improve the robustness of the algorithm, the relationship between the size and dimension of the population is given on the basis of the analysis of the algorithm’s flight behaviour and a large number of experiments. In the formula (7), the $M$ is the basic scale quantity. Where $M$ is the base swarm size for the SL-QPSO to work properly.

$$\varepsilon = \beta \times D/M.$$  \hspace{1cm} (8)

Population convergence requires the convergence of each dimension of each particle, indicating that there is a proportional relationship between the social learning factor $\varepsilon$ and the dimension $D$. However, the greater $\varepsilon$ may lead to the convergence of the algorithm to the average optimal position rather than the global optimal position. Therefore, through experimental analysis, it is reasonable to define the proportionality coefficient $\beta = 0.01$.

Step 2: The fitness value and the average optimal location are calculated, and the particles are sorted in descending order according to fitness values to update the local optimal and global optimal positions.

Step 3: Calculating the learning probability $P_i$ of each particle. When a particle’s random probability $p_i(t)$ satisfies $0 \leq p_i(t) \leq P_i^L \leq 1$, the particle carries out the social learning. The definition of learning probability $P_i^L$ is shown in formula (9):

$$P_i^L = \left(1 - \frac{i-1}{N}\right)^{\alpha \cdot \log\left(\left\lceil \frac{D}{N} \right\rceil\right)}.$$  \hspace{1cm} (9)

In formula (9), $i$ is the fitness ranking number. The greater $i$, the better the fitness value. From formula (9), we can see that there is a correlation between $P_i^L$ and fitness value and the dimension of the problem to be optimized: From $1 - (i - 1)/N$ item, we can know that the larger the $i$, the smaller the $P_i^L$, indicating the low ranking, that is the weaker the ability of particles with better fitness value learning from other individuals. On the other hand, $\alpha \log\left(\lceil D/M \rceil\right)$ indicates that $P_i^L$ is inversely proportional to the dimensions of the problem to be optimized, that is, the larger the $D$, the smaller the $P_i^L$. When solving large-scale optimization problems, particle social learning ability is stronger in order to maintain a certain diversity of population. $\alpha \cdot \log(\cdot)$ makes the effect of $D/M$ on learning probability more smooth, and the $\alpha$ recommended setting range is $\alpha < 1$.

In order to obtain a more intuitive understanding of the relationship between the learning probability $P_i^L$, swarm size $N$ and search dimensionality $D$, a number of curves showing the relationship between the learning probability and the search dimensionality varying from $D \leq 100$ to $D = 2000$ are plotted in Figure 1. It can be seen that, when the dimensionality is not large, e.g. $D \leq 100$, the learning probability keeps constant at 1 for all particles. By contrast, when the dimensionality becomes larger, the learning probability decreases as the fitness value increases (a higher index in the sorted swarm) or as the dimensionality becomes higher. It could also be noticed that, under the influence of $\alpha \cdot \log(\cdot)$ function, the probability curves decrease sharply at the very beginning whilst more gently with the increase of $D$.

Step 4: The random probability $p_i(t)$ of each particle is calculated, and the non-optimal particle potential well centre is updated by social learning strategy, as shown in (10) formula:

$$p_i(t) = \begin{cases}
    r_1(t) \cdot p_{ij}(t) + \Delta p_{ij}(t), & p_i(t) \leq P_i^L \\
    p_{ij}(t - 1), & \text{others}
\end{cases}.$$  \hspace{1cm} (10)
According to the formula (9), it can be seen that when the random probability \( p_i(t) \) is less than the learning probability \( \frac{P_i}{r_i} \), the centre of the potential well of the particle \( i \) is updated, otherwise, it will not be updated. Among them, the learning offset \( \Delta p_{ij}(t) \) of the potential well centre is determined by (11) formula:

\[
\Delta p_{ij}(t) = r_2(t) \cdot l_{ij}(t) + r_3(t) \cdot \varepsilon \cdot W_{ij}(t) \tag{11}
\]

And there is:

\[
\begin{align*}
& l_{ij}(t) = P_{kl}(t) - P_{ij}(t), \\
& C_{ij}(t) = C(t) - P_{ij}(t).
\end{align*}
\tag{12}
\]

Among them, \( r_1(t), r_2(t) \) and \( r_3(t) \) take the random numbers between \((0,1)\), and the position of the potential well centre after updating is determined by three parts: The first part is similar to the traditional QPSO algorithm, which is the function of the local optimal position \( P_{ij}(t) \) of the particle. In the second part, \( l_{ij}(t) \) represents the behavior learning from the better particle \( k \), and the local learning ability is measured by the distance between the local optimal positions between the two. In the third part, \( \varepsilon \cdot W_{ij}(t) \) represents the behavior learning from the whole group. The global learning ability is measured by the distance from the local optimal position of individual to the average optimal position of the group. The average optimal position coordinates are calculated from (4) formula.

Step 5: By substituting the results of the formula (10) into (6) formula, the positions of the particles after evolution can be obtained.

Step 6: Determine whether the precision requirement or termination conditions are met, if so, step 7 is executed, otherwise jump to step 2.

Step 7: Stop the search and output the results.

### 3.2. SL-QPSO algorithm integrating Lévy flights

In the 3.1 section, the social learning mechanism is introduced into the QPSO algorithm, which makes full use of the social information of the population to improve the update mode of the potential well centre, and improve the diversity of the population. However, there are still the following problems: Each iterative evolutionary process of SL-QPSO algorithm only updates the non-optimal particles, but ignores the guiding effect of the optimal particles. Therefore, in order to solve the above problems, the Lévy flight strategy is introduced into the SL-QPSO algorithm to update the global optimal particles and further improve the performance of the algorithm.

Lévy flight, proposed by Paul Lévy (Ghaemi, Zabihin-pour, & Asgari, 2009), is a non-Gaussian random process, and is a kind of random walk process with Markov property which is characterized by long-range hopping. The large step in the early stage is favourable for particles to jump out of the local optimum for global search, and the small step in the later stage is favourable for the local search of particles. The step length of Lévy flight obeys the Lévy distribution, that is \( \text{Lévy}(\lambda) \sim t^{-\alpha} \), in which for the index part, \( 1 < \lambda < 3 \). The mathematical description of Lévy flight is as follows:

\[
\text{Lévy}(s, \gamma, \mu) = \begin{cases} \\
\sqrt{\frac{\gamma}{2\pi}} \exp\left[-\frac{\gamma}{2(s-\mu)}\right] \frac{1}{(s-\mu)^{s-1/2}}, & 0 < \mu < s < \infty; \\
0, & s \leq 0
\end{cases}
\tag{13}
\]

Among them, \( \mu \) is the displacement parameter, and \( \gamma \) is a scale parameter and determines the distribution scale.

In order to apply the Lévy flight to the global optimal particle update in the SL-QPSO algorithm, first of all, it is necessary to discretize the Lévy flight. After discretization of formula (13), it is:

\[
x_g^\alpha(t) = x_g(t) + \alpha \oplus \text{Lévy}(\lambda).
\tag{14}
\]

Among them, \( x_g(t) \) is the global optimal particle position for the \( t \)th generation, \( x_g^\alpha(t) \) is the global optimal particle location updated by Lévy flight, \( \alpha = \alpha_0 \times (x_g(t) - x_{\text{worst}}) \) is the step length control factor, and \( x_{\text{worst}} \) is the worst particle position of the current generation. The search area will be more uniform by flying to the original small probability exploration area with this large and small step length. \( \text{Lévy}(\lambda) \) is a random search path, and \( \oplus \) represents the point multiplication operation.

The Lévy distribution is very complex and cannot be realized, so the Mantegna algorithm is used to simulate its flight path. Its mathematical expression is shown in formula (15):

\[
\text{Lévy}(\lambda) = \frac{\mu}{|u|^3}
\tag{15}
\]

Among them, \( \mu \) and \( u \) obey normal distribution, and are defined as follows:

\[
\begin{align*}
& \mu \sim N(0, \sigma_\mu^2) \\
& u \sim N(0, \sigma_u^2)
\end{align*}
\tag{16}
\]

Among them, the variance \( \sigma_\mu \) and \( \sigma_u \) are determined by formula (17):

\[
\begin{align*}
& \sigma_\mu = \left( \frac{\Gamma(1 + \chi)}{\Gamma\left(\frac{1+\chi}{2}\right)} \cdot \chi \cdot 2^{\frac{-1}{\chi}} \right)^{\frac{1}{2}} \\
& \sigma_u = 1
\end{align*}
\tag{17}
\]

In formula (17), \( \Gamma \) is a gamma function. \( \chi = 1.5 \) (Mantegna, 1994).
To illustrate the superiority of Lévy flight, Figure 2 gives the simulation experiment in two-dimensional space and records the flight path map of Lévy flight for 1000 generations.

The simulation results show that the Lévy flight process is the form of large and small intervals. This kind of flight path can increase the diversity of the population, enlarge the search range of particles, strengthen the activity and jumping ability of particles, avoid the algorithm falling into local optimum and improve the efficiency of the algorithm search.

The formulas (15)–(17) are put into formula (14), then the Lévy flight update formula can be obtained:

\[
x^l_g(t) = x_g(t) + \alpha_0 \odot \frac{\mu}{|v|^\beta}[x_g(t) - x_{worst}] .
\]

Although the Lévy flight can make particles get rid of local optimum, it cannot guarantee that the updated particle position is superior to the original position. Therefore, the greedy algorithm is used to decide whether to update the optimal particle position, that is, when the updated position is superior to the original position, the position is updated, otherwise, the original position is retained.

\[
x^\text{new}_g(t) = \begin{cases} x^l_g(t), & f(x^l_g(t)) < f(x_g(t)), \\ x_g(t), & f(x^l_g(t)) > f(x_g(t)). \end{cases}
\]

The greedy based evaluation strategy enables the improved algorithm to use each generation of optimal particles in the evolutionary process to guide other particles to search, so that the algorithm can achieve better convergence speed.

In conclusion, applying the Lévy flight to the SL-QPSO algorithm solves the problem that the optimal particles are not updated in the SL-QPSO algorithm. The QPSO algorithm integrating social learning and Lévy flight (LSL-QPSO) algorithm has improved the exploration ability and mining ability. The LSL-QPSO algorithm flow is as follows:

According to Figure 3, the specific execution flow of the LSL-QPSO algorithm is as follows:

**Step 1:** Initialize the basic parameters, including the population size \( N \), particle dimension \( D \), maximum number of iterations \( G_{\text{max}} \), particle search space, initialization space, etc;

**Step 2:** Initialize the population by random principle in the problem space; In the problem space, the population is initialized by random principle;

**Step 3:** Calculate the particle fitness value, update the local optimal position and global optimal position;

**Step 4:** Calculate the average optimal position, and sort the particles in descending order according to fitness values;

**Step 5:** To determine whether the particle is the optimal particle, if the particle is the optimal particle, Lévy flight is used to update the particle position, and greedy algorithm is used to determine whether to accept the
updated particle position; if the particle is not the optimal particle, social learning mechanism is used to update the particle position; 

Step 6: Determine whether all particles are updated. If there are other particles for updating, then jump to step 5.

Step 7: When all particles are updated, the transboundary particles are processed.

Step 8: Repeat step 3–7 until a certain loop end condition is satisfied.

4. Experimental analysis and simulation

4.1. Test function

In order to verify the performance of LSL-QPSO algorithm, four standard test functions were used to carry out simulation experiments. According to the number of extremums, it can be divided into unimodal functions and multimodal functions. The initialization and search scope of each standard test function are shown in Table 1.

4.2. Parameter setting

The LSL-QPSO algorithm was compared with QPSO, DCQPSO and SL-QPSO respectively. In order to fully explain the optimization effect, the particle size \(N\) took 20, 50, and 80, and the dimension \(D\) took 10, 20, and 30 dimensions respectively, and the iterative algebra \(G_{\text{max}}\) took 1000, 1500 and 2000 generations to carry out experiments. Setting comparing the characteristic parameters of the algorithm: In the QPSO algorithm, the CE coefficient adopted the linear decreasing strategy: the initial value took 1.0 and the final value took 0.5. The proportionality coefficient \(\beta\) = 0.01.

4.3. Result analysis

Each case runs 50 times separately. The results were recorded and the mean value of the optimal solution (Mean) and the Standard Deviation (Std) were calculated. The convergence curves of each algorithm when the particle numbers \(N = 20\), the dimension and the iteration numbers \(G_{\text{max}} = 2000\) were given. The test results are as follows.

According to Table 2, it can be seen that for the simple unimodal optimization problem similar to Sphere, the LSL-QPSO algorithm has obvious advantages, and the convergence precision is far superior to the other contrast algorithms, and the standard deviation is very small, indicating that the stability of the algorithm is good. From the convergence curve in Figure 4(a), it can be obtained that the convergence rate of LSL-QPSO algorithm is the fastest, indicating that when solving simple unimodal optimization problem, it is easier to find the global optimal point by using group information and Lévy flights. From Table 3, it can be found that for the complex unimodal optimization problem similar to Rosenbrock, the LSL-QPSO algorithm still has some advantages, and with the increase of dimension, the advantage becomes more significant, indicating that the algorithm has advantages in dealing with high dimensional problems. From Figure 4(b), we can get that the QPSO algorithm has the fastest convergence rate, however, its diversity loss is serious, and it falls into local optimum in the 800th generation. Compared with the DCQPSO and SL-QPSO algorithms, LSL-QPSO has faster convergence rate in the early stage, and the better solution is found in the 1000th generation, and the convergence accuracy of the algorithm is also better than the other three algorithms.

According to Table 4, it can be found that for the multimodal optimization problem similar to Ackley, the
convergence accuracy of the LSL-QPSO algorithm is much better than that of the other three comparison algorithms, and the standard deviation shows that the algorithm has better stability. From Figure 5(a), QPSO algorithm is difficult to obtain better solution. Although the DCQPSO and SL-QPSO algorithms have some data mining capability in the early stage, when it comes to the 1400th generations, most of the particles have fallen into local optimum. The convergence curve of LSL-QPSO shows that the algorithm not only has a faster convergence rate in the early stage, the final convergence accuracy is also far superior to other algorithms. It can be seen from Table 5 that for the complex multimodal optimization problem similar to Schwefel, the performance advantage of LSL-QPSO algorithm has decreased, but it is still better than other algorithms. As shown in Figure 5(b), QPSO, SL-QPSO and LSL-QPSO algorithms converge in the 800th generation, but LSL-QPSO algorithm has the highest convergence accuracy. When in the 1100th generation, although human disturbance can cause population to spread for a short time, cause the potential well centre offset and cause the restoration of divergence, the DCQPSO algorithm will fall into local optimum quickly.

Analysing the experiment results of the above 4 functions, we can get the basic reason of the performance advantages of the LSL-QPSO algorithm: all non-optimal particles in the population were involved in the update process. It can be seen from Figure 5(b), the QPSO, SL-QPSO and LSL-QPSO algorithms converge well during the 800th generation, but LSL-QPSO algorithm has some data mining capability in the early stage, when it comes to the 1400th generations, most of the particles have fallen into local optimum. The convergence curve of LSL-QPSO shows that the algorithm not only has a faster convergence rate in the early stage, the final convergence accuracy is also far superior to other algorithms. It can be seen from Table 5 that for the complex multimodal optimization problem similar to Schwefel, the performance advantage of LSL-QPSO algorithm has decreased, but it is still better than other algorithms. As shown in Figure 5(b), QPSO, SL-QPSO and LSL-QPSO algorithms converge in the 800th generation, but LSL-QPSO algorithm has the highest convergence accuracy. When in the 1100th generation, although human disturbance can cause population to spread for a short time, cause the potential well centre offset and cause the restoration of divergence, the DCQPSO algorithm will fall into local optimum quickly.

Analysing the experiment results of the above 4 functions, we can get the basic reason of the performance advantages of the LSL-QPSO algorithm: all non-optimal particles in the population were involved in the update process.

Table 3. Comparison of Rosenbrock simulation results.

| N | D | Gmax | QPSO Mean | QPSO Std | DCQPSO Mean | DCQPSO Std | SL-QPSO Mean | SL-QPSO Std | LSL-QPSO Mean | LSL-QPSO Std |
|---|---|------|-----------|---------|-------------|-----------|-------------|-----------|--------------|-------------|
| 20 | 10 | 1000 | 1.75E+01 | 3.15E+01 | 7.35E+00 | 1.54E+00 | 5.53E+00 | 4.56E+00 | 2.26E+00 | 9.93E-02 |
| 30 | 1500 | 1.15E+02 | 1.01E+02 | 6.75E+01 | 5.71E+01 | 2.86E+01 | 1.61E+01 | 1.07E+01 | 1.12E-01 |
| 50 | 2000 | 1.38E+04 | 8.64E+03 | 2.96E+02 | 4.46E+02 | 1.42E+02 | 8.35E+01 | 4.78E+01 | 1.10E-01 |

Table 4. Comparison of Ackley simulation results.

| N | D | Gmax | QPSO Mean | QPSO Std | DCQPSO Mean | DCQPSO Std | SL-QPSO Mean | SL-QPSO Std | LSL-QPSO Mean | LSL-QPSO Std |
|---|---|------|-----------|---------|-------------|-----------|-------------|-----------|--------------|-------------|
| 20 | 10 | 1000 | 9.10E-01 | 8.13E-01 | 2.66E-15 | 1.87E-15 | 1.01E-15 | 1.32E-15 | 8.88E-16 | 5.27E-18 |
| 30 | 1500 | 4.35E-00 | 6.03E-01 | 2.78E-06 | 3.26E-06 | 1.04E-06 | 2.03E-06 | 2.59E-15 | 5.02E-16 |
| 50 | 2000 | 6.34E+00 | 8.07E-01 | 4.39E-03 | 3.82E-03 | 4.99E-05 | 4.85E-05 | 2.66E-15 | 1.32E-16 |
| 50 | 10 | 1000 | 5.08E-01 | 7.49E-01 | 2.66E-15 | 0.00E+00 | 1.76E-13 | 2.16E-13 | 8.88E-16 | 8.54E-16 |
| 30 | 1500 | 2.98E-00 | 5.55E-01 | 1.70E-13 | 2.75E-12 | 1.70E-13 | 2.75E-12 | 2.45E-15 | 8.52E-16 |
| 50 | 2000 | 5.59E+00 | 5.65E-01 | 1.03E-07 | 4.73E-07 | 1.03E-07 | 4.73E-07 | 2.66E-15 | 2.37E-15 |
| 80 | 10 | 1000 | 4.59E-00 | 5.65E-01 | 2.66E-15 | 2.66E-15 | 2.66E-15 | 2.66E-15 | 4.88E-16 | 8.88E-16 |
| 30 | 1500 | 2.76E-00 | 5.76E-01 | 9.16E-14 | 3.79E-14 | 9.16E-14 | 3.79E-14 | 2.66E-15 | 2.66E-15 |
| 50 | 2000 | 4.08E+00 | 6.52E-01 | 2.07E+01 | 5.02E-02 | 2.07E+13 | 5.02E-13 | 5.66E-15 | 7.66E-15 |
4.4. Computational complexity

In order to further discuss the universality of the LSL-QPSO algorithm for high-dimensional problems, the time complexity of the algorithm is analysed.

From the algorithm flow, it can be seen that the time complexity of LSL-QPSO algorithm is restricted by three parts: sorting of fitness values, social learning and Lévy flights.

In the LSL-QPSO algorithm, the sorting of fitness values is a typical sort problem. Considering the worst case, for the population with the size of $N$, the time complexity $T_s$ in the execution process of sorting fitness values is determined by the formula (20):

$$T_s = O(N^2).$$  \hfill (20)

The sorting of fitness values in LSL-QPSO algorithm can be realized by quick sorting algorithm. According to the formula (8), it can be known that when $D \leq 100$, $N$ is a fixed value. When $D > 100$, for example $D = 200$, $N = M + 20$ can be obtained according to the formula (8). Therefore, it can be seen that the time complexity in the execution process of sorting fitness values will not increase significantly in the form of square as the dimension of the optimization problem increases.

The time complexity in the execution process of the social learning is similar to that of other swarm intelligence algorithms (Kendal et al., 2018). If the population size is $N$ and the dimension is $D$, the time complexity $T_c$ in the execution process of the social learning is determined by the formula (21):

$$T_c = O(ND).$$  \hfill (21)

Similarly, the time complexity $T_l$ in the execution process of the Lévy flights is:

$$T_l = O(ND).$$  \hfill (22)
In order to illustrate the relationship between the time complexity of the algorithm and the dimension of the optimization problem, the change curve of the time complexity in the execution process of the fitness value sorting, social learning and Lévy flights is drawn in Figure 5 at the dimension of 1-500.

From the Figure 6, it can be seen that the time complexity of the LSL-QPSO algorithm is \( \max(T_s, T_c, T_l) \). When \( D > 111 \), \( T_c > T_l \), and the time complexity of the LSL-QPSO algorithm is \( T_c = T_l = O(ND) \). However, the time complexity of the basic QPSO algorithm is also \( O(ND) \), and the time complexity is in the same order, therefore, the LSL-QPSO algorithm does not increase significantly in time complexity due to the introduction of improved ideas. At the same time, the above analysis also explains why the performance of LSL-QPSO algorithm does not decay as the optimization problem dimension increases.

5. Conclusions

In this paper, the update method of potential well centre in QPSO algorithm is studied. In the proposed LSL-QPSO algorithm, the social learning helps to strengthen the information sharing among particles, and the Lévy flights helps to produce better initial population, and can enhance the guiding role of the optimal particle. The combination of the two learning mechanisms and their complementary advantages make the algorithm achieve a good balance in terms of convergence precision and search efficiency. The advantages of LSL-QPSO in convergence precision and search efficiency are demonstrated by experimental comparison and simulation. Compared with the similar algorithms, the algorithm has strong universality, and the increase of dimension will not cause the fast loss of algorithm performance, which has a great application prospect on large-scale and complex optimization problems. Next, we will consider the combination of algorithm and practice to solve practical engineering problems.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work is partially supported by the Research on Bipolar Analysis Theory and Method Oriented to Big Chemical Data in NSFC-Zhejiang Joint Fund Project U1509217 2016.01-2019.12.

References

Cheng, R., & Jin, Y. (2015). A social learning particle swarm optimization algorithm for scalable optimization. Information Sciences, 291(6), 43–60.

Clerc, M., & Kennedy, J. (2002). The particle swarm – explosion, stability, and convergence in a multidimensional complex space. IEEE Transactions on Evolutionary Computation, 6(1), 58–73.

Collotta, M., Pau, G., & Maniscalco, V. (2017). A fuzzy logic approach by using particle swarm optimization for effective energy management in WSNs. IEEE Transactions on Industrial Electronics, PP(99), 1–1.

Couceiro, M., & Ghamisi, P. (2016). Particle swarm optimization. Fractional order Darwinian particle swarm optimization. Berlin: Springer International Publishing.

Frans, V. D. B. (2006). An analysis of particle swarm optimizers. Phd Thesis.

Galef, B. G., & Laland, K. N. (2005). Social learning in animals: Empirical studies and theoretical models. Bioscience, 55(6), 489–499.

Ghaemi, M., Zabihinpour, Z., & Asgari, Y. (2009). Computer simulation study of the Levy flight process. Physica A: Statistical Mechanics and its Applications, 388(8), 1509–1514.

Ghasemi, M., Aghaei, J., & Hadipour, M. (2017). New self-organising hierarchical pso with jumping time-varying acceleration coefficients. Electronics Letters, 53(20), 1360–1362.

Kendal R, L., Boogert N, J., Rendell, L., Laland, N., Webster, M., & Jones, L. (2018). Social learning Strategies: Bridge-Building between fields[J]. Trends in Cognitive Sciences, 22(7), 651–665.

Kennedy, J., & Eberhart, R. (2002). Particle swarm optimization. IEEE International Conference on Neural Networks, 1995. Proceedings (Vol.4, pp. 1942–1948). IEEE.

Li, Y., Jiao, L., Shang, R., & Stolkin, R. (2015). Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Information Sciences, 294, 408–422.

Li, S. J., Yuan, P. H., & Wang, X. D. (2016). Improved quantum behaved particle swarm optimization algorithm. Chinese Control and Decision Conference (pp. 774–776).

Mantegna, R. N. (1994). Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes. Physical Review E Statistical Physics Plasmas Fluids & Related Interdisciplinary Topics, 49(5), 4677–4677.
Rahmat-Samii, Y., Gies, D., & Robinson, J. (2017). Particle swarm optimization (PSO): a novel paradigm for antenna designs. *Ursi Radio Science Bulletin, 76*(3), 14–22.

Singh, M. R., & Mahapatra, S. S. (2016). *A quantum behaved particle swarm optimization for flexible job shop scheduling*. New Delhi: Pergamon Press, Inc.

Sun, J., Feng, B., & Xu, W. (2004). Particle swarm optimization with particles having quantum behavior. *Evolutionary Computation, 2004. CEC2004. Congress on* (Vol. 70, pp. 325-331 Vol.1). IEEE.

Tang, J., Wang, D., Wang, X., Jia, H., Wang, C., & Huang, R. (2017). Study on day-ahead optimal economic operation of active distribution networks based on kriging model assisted particle swarm optimization with constraint handling techniques. *Applied Energy, 204*, 143–162.

Wang, G. G., Gandomi, A. H., Alavi, A. H., & Deb, S. (2016). A hybrid method based on krill herd and quantum-behaved particle swarm optimization. *Neural Computing and Applications, 27*(4), 989–1006.

Wang, A. L., Jian-Hua, H. E., Chen, S., & Liu, H. Y. (2014). Research on QPSO algorithm of double core disturbance. *Computer Engineering, 40*(7), 193.

Wu, T., Yan, Y. S., & Chen, X. (2015). Improved dual-group interaction qpsq algorithm based on random evaluation. *Kongzhi Yu Juece/Control & Decision, 30*(3), 526–530.