

A Quantum Glowworm Swarm Optimization Algorithm based on Chaotic Sequence

Du Pengzhen, Tang Zhenmin and Sun Yan

*College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
h.k@foxmail.com*

Abstract

The standard Glowworm Swarm Optimization (GSO) has poor global search ability and easily trap into local optimum. In order to solve these problems, a Quantum Glowworm Swarm Optimization Algorithm based on Chaotic Sequence (QCSGSO) is proposed in this paper. Firstly, chaotic sequence is generated to initialize the population, which has higher probability to cover more local optimal areas, and provides a good condition for further optimization and tuning. Then, quantum behavior is applied to elite population, which makes individuals locate in any position of the solution space randomly with a certain probability, greatly enhances the algorithm's capability of global searching and local optimum jumping. Finally, QCSGSO adopts single dimension loop swimming rather than the original fixed step movement mode, which not only improves the solution precision and convergence speed, but also solves GSO's problem about too sensitive to the step-size, and enhances the robustness of the algorithm indirectly. The results of simulation experiments show that the proposed method is feasible and effective.

Keywords: *Glowworm Swarm Optimization; Quantum Behavior; Chaotic Sequence; Single Dimension Swimming; Function Optimization*

1. Introduction

The Glowworm Swarm Optimization (GSO) [1], mimicking the courtship behavior of glowworm, is proposed by Krishnanand and Ghose in 2005. It has been successfully used in various applications such as multi-robot localization [2], multi-signal source localization [3], multi-modal functions optimization [4] and wireless sensor deployment [5]. Different from general intelligent optimization algorithms, the greatest characteristic of GSO is that it can find multiple optima of multi-modal objective functions. However, the No Free Lunch (NFL) theorem assumes that for any algorithm, any elevated performance over one class of problems is offset by performance over another class. Similarly, GSO is weak in global search ability and convergence speed, especially in the process of optimization of complex problems.

In order to avoid these drawbacks, a lot of improved methods have been proposed to enhance the performance of GSO. Wu and Qian [8] put forward three kinds of movement method, which enhanced the algorithm's global search ability and convergence speed. He and Tong [9] replaced the worst individual by middle individual to speed up evolution. Zhang and Ma [10] improved the precision of the algorithm by adopting adaptive step size. Zhou and Wang [11] introduced Lévy flights with higher randomness into population, and greatly improved the basic algorithm in terms of overall and convergence. Zhou and Huang [13] proposed the discrete glowworm swarm

optimization algorithm that successfully been used to solve TSP, expanded the application range of GSO.

The GSO completes evolution through small scale search in multiple local optimal areas. As its movement mode depends only on local information, compared with other intelligent optimization algorithms such as particle swarm optimization algorithm (PSO) and artificial bee colony algorithm (ABC), it lacks a direct way of global search. Meanwhile, GSO is too sensitive to step-size to balance the ability of searching and tuning. In the view of mentioned above, we proposed a quantum glowworm swarm optimization algorithm based on chaotic sequence (QCSGSO). First, initialize the population with chaotic sequences, in order to cover more local optimal areas with great probability. Then, all the local optimal individuals in each iteration are classified as elite population. On the basis of this, the Delta potential well model of elite population is constructed in a quantum space, adopting the concept of quantum behavior in reference [14]. The Monte Carlo stochastic sampling method is used to ensure the individuals could appear anywhere within the solution space. Finally, the original movement mode of fixed step-size is replaced by single dimension loop swimming to further complete the high precision tuning in local area and make the GSO not too sensitive to the step-size. A number of simulation experiments show that the proposed method QCSGSO is better than the standard GSO in terms of solution quality, convergence speed and solution precision.

2. Standard Glowworm Swarm Optimization

In GSO, a glowworm represents a feasible solution. Each of them carries a light called luciferin value, and has its own vision range called local-decision range. At the beginning of the algorithm, each individual searches for lighter neighbors within its local-decision range to built neighborhood set and then flies towards them randomly. After the movement phase, each individual changes its local-decision range depends on the number of neighbors, and updates the luciferin according to the fitness of its position. Finally, all the individuals get together at multiple lighter individuals, complete the optimization process. Specific process is as follows.

Step1: Construct $m \in N$ feasible solutions randomly in the search space as x_1, x_2, \dots, x_m . Meanwhile, for each individual, evaluate the initial luciferin value l_0 according to the fitness of its position.

Step2: Construct the neighborhood set N_i for glowworm i ($i \leq m$) according to Eq. (1).

$$N_i(t) = \{ x_j \mid l_i(t) < l_j(t), \|x_j(t) - x_i(t)\| < r_i(t) \} \quad (1)$$

where $l_i(t)$ represents the luciferin value of glowworm i at time t , $r_i(t)$ represents the variable neighborhood range associated with glowworm i at time t , $\|\cdot\|$ represents the Euclidean norm operator.

Step3: Let glowworm i select a glowworm j to move toward by roulette wheel method based on the probability p_{ij} that is given as Eq. (2). The movement can be stated as Eq. (3).

$$p_{ij}(t) = \frac{l_j(t) - l_i(t)}{\sum_{u \in N_i(t)} l_u(t) - l_i(t)}, j \in N_i(t) \quad (2)$$

$$x_i(t+1) = x_i(t) + s \times \left(\frac{x_j(t) - x_i(t)}{\|x_j(t) - x_i(t)\|} \right) \quad (3)$$

Where s is the step-size ($s > 0$).

Step4: The local-decision range of glowworm i is updated by using the rule given below:

$$r_i(t+1) = \min\{r_s, \max\{0, r_i(t) + \beta(N_i - |N_i(t+1)|)\}\} \quad (4)$$

where r_s is the maximum local-decision range of the luciferin sensor, β is a decision domain update constant, N_i is a threshold used to control the number of neighbors, $|\cdot|$ is the size of a set.

Step5: Glowworm i updates its luciferin value according to the rule as given by Eq. (5).

$$l_i(t+1) = (1 - \rho)l_i(t) + \gamma J(x_i(t)) \quad (5)$$

where $\rho \in (0, 1)$ is the luciferin decay constant, $\gamma \in (0, 1)$ is the luciferin enhancement constant, and $J(x_i(t))$ represents the fitness of individual i at time t .

Step6: If the specified iteration number is reached or the current optimal solution is desirable, the algorithm finishes. Otherwise turn to Step 2.

3. Proposed Algorithm

In this section, three strategies are conducted to improve the standard GSO.

3.1. Chaotic Sequence

Chaos is a kind of seeming random and irregular movement, which is a common phenomenon in nonlinear system. It has the characteristic of randomness and ergodicity so that it could ergodic all the states in limited space without repetition. The general way of random initialization is hard to ergodic solution space non-repeatedly under the arbitrary precision. But for the chaotic sequence, it is feasible. The population, initialized by chaotic sequence, is likely to distribute in more local optimal areas, which provides a prerequisite for the algorithm to obtain a high solution quality and precision.

In this paper, the chaotic sequence is generated by using Logistic equation as given in Eq. (6), and is mapped to the solution space by the way of carrier.

$$z_{i+1} = \mu z_i(1 - z_i), \quad 1 < i \leq m - 1 \quad (6)$$

where $z_i \in [0, 1]$, $z_i \neq 0.25$ and $z_i \neq 0.75$, μ is the control parameter of chaotic state, m is the population size. While $\mu = 4$, the system is in complete chaotic state.

3.2. Elite Population and Quantum Behavior

In this paper, quantum behavior is applied to elite population. Firstly, the elite population is defined as follows:

$$elite(t) = \{x_i \mid N_i(t) = \emptyset\} \quad (7)$$

In the standard GSO, the elite population keeps still, only non-elite population move around them, details see Eq. (3). The essence of such movement mode is that the population scatters in multiple local optimal areas, and do small-scale search centered on their respective optimal individuals in the areas, which effectively guarantee the GSO's capability to cover multiple optima. However, such movement mode is not an effective way of global searching, because of depending only on local information. So the standard GSO has poor global search ability.

Based on the analysis above, quantum behavior make sure the elite population have the capability of appearing anywhere in the solution space at a certain probability, which could largely enhance the GSO's search ability. Specific process is as follows:

Step1: According to reference [15], if want to ensure the algorithm's convergence, the position of a random individual i has to converge to local attractor P_i , as given in Eq.(8).

$$P_i(t) = \phi x_i^p + (1 - \phi)x_g(t) \quad (8)$$

where ϕ is a random variable subjects to uniformly distribution in the interval [0,1], x_i^p is the best position that x_i has searched, x_g is the optimal individual of the population.

Step2: Restrict the population within a Delta potential well model to keep the population in aggregation state. In such situation, the probability density function of individual position could be deduced.

As in quantum space, particle position and velocity can not be measured at the same time. Its state could be depicted by wavefunction $\Psi(Y, t)$. $|\Psi|^2$ is the probability density function of particle position. Therefore, on the basis of step 1, Delta potential well model is built for a random dimension of P_i . Its potential energy function is as follows:

$$V(Y) = -\gamma\delta(Y) \quad (9)$$

$$Y = x_i - P_i \quad (10)$$

Plug the potential energy function into the Schrodinger equation:

$$\left(-\frac{\hbar^2}{2\mu} \frac{\partial^2 \Psi(Y, t)}{\partial Y^2} + V(Y) \right) \Psi(Y, t) = i\hbar \frac{\partial \Psi(Y, t)}{\partial t} \quad (11)$$

By solving the Eq. (11) could get the probability density function.

$$Q(Y) = |\Psi|^2 = \frac{e^{-2|Y|/L}}{L} \quad (12)$$

$$L = 2b(t) \times \left| \frac{1}{m} \sum_{i=1}^m x_i^p - x_i(t) \right| \quad (13)$$

where \hbar is the Planck Constant, L is the characteristic length of Delta potential well, $|\cdot|$ is the absolute value computation, $b(t)$ a control variable that linear decrease from 1.0 to 0.5 with the increase of iteration times.

Step3: On the basis of probability density function, the individual motion equation is simulated by the Monte Carlo method, as given by Eq. (14). Let $u = e^{-2|V|/L}$, then u is a random number uniformly distributed on (0,1).

$$x_i(t+1) = P_i(t) \pm \frac{L}{2} \ln(1/u) \quad (14)$$

Above all, the location update formula is as follows:

$$x_i(t+1) = \phi x_i^p + (1-\phi)x_g(t) \pm \ln u \times b(t) \times \left| \frac{1}{m} \sum_{j=1}^m x_j^p(t) - x_i(t) \right| \quad (15)$$

In Eq. (15), ϕx_i^p is the individual cognitive, $(1-\phi)x_g(t)$ is the population cognitive, $\sum_{j=1}^m x_j^p(t) / m - x_i(t)$ is the difference between individual location and mean population optimal location. If an individual's location is quite different from the mean population optimal location, the other individuals in this population will wait for it to catch up. Meanwhile, other individuals will conduct a large range search around $\phi x_i^p + (1-\phi)x_g(t)$, which represents the individual cognitive and population cognitive. Such improved method not only greatly enhances the global search ability, but also effectively avoids the premature.

3.3. Single Dimension Swimming

In the standard GSO, step-size is an important control parameter, see Eq. (3). If the step-size is big, the algorithm has strong global search ability, but its precision is low and easy to miss the optimal position. On the contrary, it has better tuning ability, but lack global search ability and easy to trap into local optimum.

In order to improve the precision of standard GSO, a motion mode of single dimension swimming is introduced. Each individual keeps on swimming by a step size s , as Eq.(17), until there is no improved fitness value or reach the search space boundary. See Eq. (16).

$$x_{id}(t+1) = x_{id}(t) + s \quad (16)$$

$$s = rand(0, 1)(x_{jd} - x_{id}) \quad (17)$$

where $rand(0, 1)$ represents a random number between 0 and 1, x_{id} is the d th dimension of individual i , and j is calculated by Eq. (2).

3.4. The Whole Process of the Proposed Algorithm

QCSGSO shares the same framework with the standard GSO. Fig. 1 shows the flowchart of the improved algorithm.

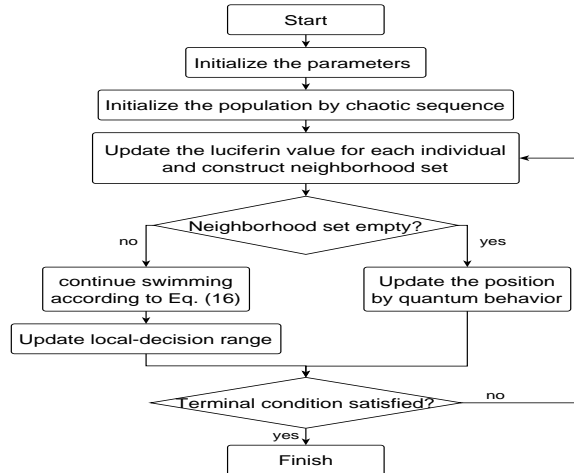


Figure 1. Algorithm Flowchart

4. Experiments and Discussions

Most of the existing improved intelligent optimization algorithm showed good performance on the uni-modal and low-dimensional problems, but for multi-modal and high-dimensional functions, they often cannot achieve ideal results.

Table 1. Benchmark Functions

| Funciton | Function Formula | Domain | Type |
|---------------|---|--------------|------|
| Sphere | $f_1(x) = \sum_{i=1}^n x_i^2$ | [-100,100] | UM |
| SumSquares | $f_2(x) = \sum_{i=1}^n ix_i^2$ | [-10,10] | UM |
| Quartic | $f_3(x) = \sum_{i=1}^n ix_i^4 + random[0,1)$ | [-1.28,1.28] | UM |
| Schwefel 2.22 | $f_4(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $ | [-10,10] | UM |
| Schwefel 1.2 | $f_5(x) = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$ | [-100,100] | UM |
| Rosenbrock | $f_6(x) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$ | [-30,30] | UM |
| Dixon-Price | $f_7(x) = (x_1 - 1)^2 + \sum_{i=2}^n i(2x_i^2 - x_{i-1})^2$ | [-10,10] | UM |
| Griewank | $f_8(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$ | [-600,600] | MM |
| Ackley | $f_9(x) = -20 / \exp(\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)) + 20 + e$ | [-32,32] | MM |
| Rastrigin | $f_{10}(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$ | [-5.12,5.12] | MM |

*UM: Uni-modal, MM:Multi-modal

In order to fully verify the QCSGSO, some experiments have been done to evaluate its performance. We select ten widely used benchmark functions for test. All the functions are to be minimized and their global optimal values are 0. The function names, function formulas, domain and the type are summarized in Table 1. Where $f_1 \sim f_7$ are uni-modal functions used for

testing the algorithm's precision and convergence speed, and $f_8 \sim f_{10}$ are multi-modal functions used for testing the ability of searching and jumping out of local optimum. For the sake of comparison, the best results obtained by the algorithms for every function are denoted by bold. Values that less than E-100 are reported as 0. The simulation platform is Windows XP, AMD CPU 2.9GHz, 2GB memory, VS2010 programming suit.

4.1. Optimization Performance Comparison

In order to verify the optimization performance of the proposed algorithm QCSGSO, it was first compared with standard GSO and improved GSO from reference [8], called AGSO and PGSO. Then it was compared with other intelligent optimization algorithm such as genetic algorithm (GA) [16], particle swarm optimization (PSO) [17] and artificial bee colony algorithm (ABC) [18].

4.1.1. Comparison with Standard GSO and Improved GSO: Third-order headings, as in this paragraph, are discouraged. However, if you must use them, use 11-point Times New Roman, boldface, initially capitalized, flush left, preceded by one blank line, followed by a colon and your text on the same line.

In this section, we compare the proposed QCSGSO with standard GSO and two modified algorithms call AGSO and PGSO from reference [8], in which a number of experiments about complex functions with high-dimension were carried out. The dimensions of all the benchmark functions are set to 10.

In the experiment, the population sizes of all the algorithms are set $NP = 100$, the max iteration is set $MCN = 5000$. For QCSGSO, the luciferin decay constant $\rho = 0.4$, the luciferin enhancement constant $\gamma = 0.6$, the neighborhood change rate $\beta = 0.08$ and the neighborhood threshold $N_t = 5$. For GSO, the step-size $s = 0.8$, other parameters are the same as QCSGSO. The parameters of AGSO and PGSO refer to the reference [8]. Each benchmark function in Fig. 1 is independently run with every algorithm for 20 times. The best, mean and standard deviation of the algorithms' test results are compared in Table 2. The best value measures the solution quality and precision, mean value measures the global performance and convergence, and the standard deviation measures the stability of the algorithm.

The results in Table 2 show that the QCSGSO outperforms other algorithms in terms of solution quality and precision. It obtain better results than the others on all benchmark functions, especially on $f_1 \sim f_4$, f_8 and f_{10} . The mean value contrast result shows that QCSGSO owns high solution quality and convergence speed, and global performance is better than others. For QCSGSO, all the standard deviation values are remain in a lower level, which means it has strong stability.

It should be noted that the optimum value of function f_6 is in a narrow area. It's often used to test an algorithm's search ability and the ability of jumping out of local optimum, as it is easy to premature convergence. However, QCSGSO gets ideal results in the test of function f_6 , which fully demonstrate its strong ability to jump out of local optimum.

Table 2. Comparison with GSO, AGSO and PGSO

| | Best (Mean/Dev) | | | |
|----------|-------------------------------|---|---|---|
| | GSO | AGSO | PGSO | QCSGSO |
| f_1 | 1.58E-06 4.72E-06/2.64E-06 | 9.17E-17 2.55E-16/9.25E-17 | 8.19E-23 4.84E-20/7.24E-20 | 0.00E+00 0.00E+00/0.00E+00 |
| f_2 | 4.12E-06 1.90E-05/1.14E-05 | 3.68E-14 5.47E-13/3.29E-13 | 7.35E-18 6.33E-16/1.37E-15 | 0.00E+00 0.00E+00/0.00E+00 |
| f_3 | 2.16E-13 2023E-11/2.44E-11 | 1.08E-03 3.38E-03/1.53E-03 | 7.52E-07 3.82E-05/3.31E-05 | 0.00E+00 0.00E+00/0.00E+00 |
| f_4 | 1.98E-03 6.18E-03/3.50E-03 | 0.00E+00 0.00E+00/0.00E+00 | 0.00E+00 0.00E+00/0.00E+00 | 0.00E+00 0.00E+00/0.00E+00 |
| f_5 | 2.26E-06 1.75E-05/1.70E-05 | 5.57E-16 2.54E-05/6.16E-05 | 3.84E-22 6.32E-20/9.61E-20 | 4.28E-38 2.65E-32/2.34E-33 |
| f_6 | 3.67E-01 9.31E+00/1.97E+01 | 1.99E-05 3.80E+00/2.70E+00 | 2.02E-04 5.35E+00/1.77E+00 | 1.65E-08 6.10E-02/5.87E-02 |
| f_7 | 6.70E-01 6.77E-01/3.52E-03 | 3.00E-16 9.36E-08/3.84E-07 | 1.58E-24 5.82E-22/1.14E-21 | 2.47E-31 4.56E-30/8.90E-30 |
| f_8 | 9.73E-01 2.86E+00/8.76E-01 | 1.11E-16 3.89E-16/3.21E-16 | 0.00E+00 0.00E+00/0.00E+00 | 0.00E+00 0.00E+00/0.00E+00 |
| f_9 | 1.02E-03 2.92E-03/1.39E-03 | 2.96E-15 6.46E-14/2.50E-13 | — | 2.33E-15 2.11E-14/1.01E-14 |
| f_{10} | 2.05E+00 9.18E+00/3.86E+00 | 0.00E+00 1.36E+00/3.25E+00 | 0.00E+00 0.00E+00/0.00E+00 | 0.00E+00 0.00E+00/0.00E+00 |

4.1.2. Comparison with Other Algorithms: In this section, the proposed QCSGSO is compared with the results from reference [18] of genetic algorithm (GA), particle swarm optimization (PSO) and artificial bee colony algorithm (ABC). In this experiment, dimensions are set to 30 and values that less than E-12 are reported as 0 to consistent with reference [18]. All the algorithms are simulated 20 times, each time has 5000 iterations.

Table 3. Comparison with Other Algorithms

| | Mean(Dev) | | | |
|----------|--------------------|---------------------------|---------------------------|---------------------------|
| | GA | PSO | ABC | QCSGSO |
| f_1 | 1.11E+03(7.42E+01) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_2 | 1.48E+02(1.24E+01) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_3 | 1.81E-01(2.71E-02) | 1.16E-03(2.76E-04) | 3.00E-02(4.87E-03) | 0.00E+00(0.00E+00) |
| f_4 | 1.10E+01(1.39E+00) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_5 | 7.40E+03(1.14E+03) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_6 | 1.96E+05(3.85E+04) | 1.51E+01(2.42E+01) | 8.88E-02(7.74E-02) | 7.05E-02(1.15E-02) |
| f_7 | 1.22E+03(2.66E+02) | 6.67E-01(1.00E-08) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_8 | 1.06E+01(1.16E+00) | 1.74E-02(2.08E-02) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_9 | 1.47E+01(1.78E-01) | 1.65E-01(4.94E-01) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |
| f_{10} | 5.29E+01(4.56E+00) | 4.40E+01(1.17E+01) | 0.00E+00(0.00E+00) | 0.00E+00(0.00E+00) |

The mean value and the standard deviation are given in Table 3. Take a global look at Table 3, we can see that QCSGSO provide better performance than all the other contrast

algorithms, and ABC is better than GA and PSO. All the algorithms have no good performance on f_3 except QCSGSO. From the result of test on the function f_6 , we observe that the performance of QCSGSO is close to but better than ABC, while GA and PSO demonstrate worse performance.

Overall, the introduction of quantum behavior and single swimming greatly enhanced the QCSGSO's search ability, and its stability and global optimization performance is superior to contrast algorithms.

4.2. Comparison of Convergence Speed

In this section, four widely used benchmark function, Schwefel 1.2, Rosenbrock, Griewank and Ackley are chosen to test the convergence speed of QCSGSO. The dimension of each function is set 10. Related parameters are the same as Section 4.1, and the results are shown in Figure 2 whose axis-y represents log of the optimal solution and axis-x represents the iterative number.

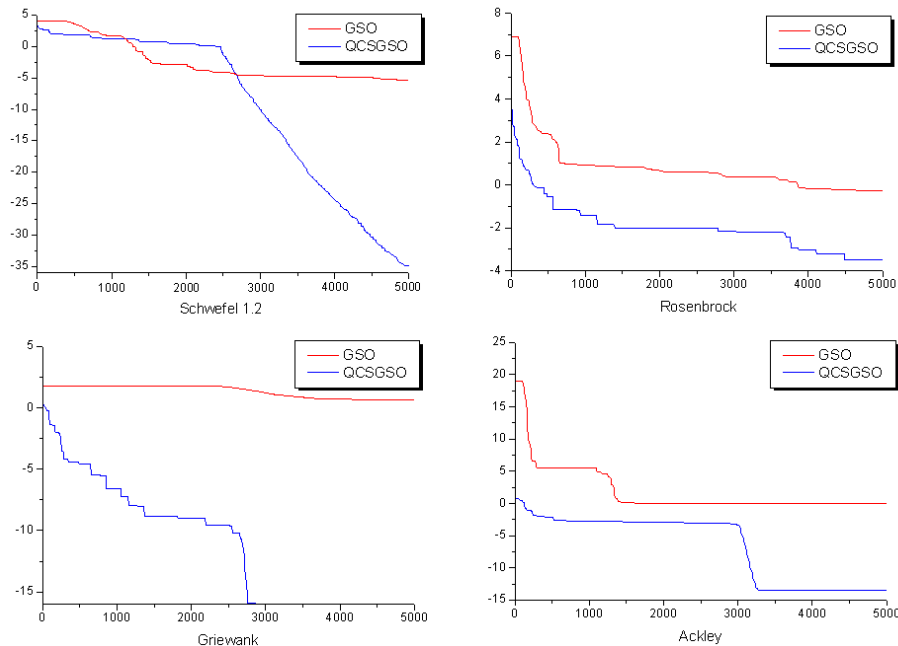


Figure 2. Convergence Speed Comparison

It can be seen from Figure 2, the curve of QCSGSO is generally under that of GSO. That means, with the same iteration number, the solution quality and convergence speed of QCSGSO are higher than GSO.

It should be noted that all of the initial optimal solutions of QCSGSO are better than GSO, especially on function Ackley, the difference between them close to 20 orders of magnitude. It can be explained that the population of QCSGSO is initialized by chaotic sequence, which makes the population more likely to close to or reach the optimal solution area at the beginning. Such phenomenon is particularly obvious in multi-modal function test. Both of Griewank and Ackley are typical multi-modal functions. On the tests of them, the QCSGSO's initial population initialized by chaotic sequence reaches the order of magnitude as E+00, which is far better than GSO.

As can be seen from Figure 2, the curve of GSO is quite gentle in the late iterations while that of QCSGSO still remain a steep range. This should be due to quantum behavior, which makes the population keep on wide range of search around the individual optimal position and population optimal position. Also, the single dimension loop swimming mode can not only make high precision tuning in local optimal region, but also jump out of local optimum with strong possibility to achieve better solution.

4.3. Population Diversity Analysis

While the population hasn't reach the global optimal area, a high population diversity means the population has high global search ability and more likely to obtain the global optimal solution. But while the population has reached it, a high population diversity means that the population has more possibility to jump out of local optima and avoid prematurity.

In order to fully verify QCSGSO, it is compared with standard GSO about population diversity. In this test, the diversity measure for population is the "distance-to-average-point" measure [19], defined as Eq.(18).

$$diversity(P) = \frac{2}{mL} \sum_{i=1}^m \sqrt{\sum_{d=1}^D (x_{id} - \bar{x}_d)^2} \quad (18)$$

where P is the population whose size is m , L is the diagonal length of the D dimension solution space, x_{id} is the d th dimension of individual i , \bar{x}_d is the average of the d th dimension of population P .

Through the analysis of Figure 3, three points can be concluded that:

1) The initial population diversity of QCSGSO is lower than GSO. That is because the chaotic sequence could be regarded as a ergodic random sequence. And compared to ordinary random sequence, there is a big possibility for the chaotic sequence to have a small "distance-to-average-point". Therefore, the result is reasonable, and this point indirectly indicates that the chaotic sequence is more likely to cover the global optimal area directly.

2) Different from GSO, the population diversity of QCSGSO reaches a low level in a few iterations. That is because the high population diversity represents this population distributes throughout the whole solution space, while the low population diversity represents this population locates in a relatively narrow area. At the beginning, the algorithm makes a wide global search in the whole solution space. After some iterations, the population enter a relatively narrow area to start tuning. Combined with the results of Table 2 shows that the population of QCSGSO achieves the global optimal area only in a few iterations, while GSO needs several times and even hundreds times of iteration numbers more than QCSGSO. This point indicates that the convergence speed of QCSGSO is high and its overall performance is much better than GSO.

3) In the later iterations, the diversity curve of GSO almost remains unchanged, but that of QCSGSO occurs small sharp fluctuations. That is because after finishing the global search, the whole population will distribute in a relatively narrow optimal area. It is conducive to obtain a higher precision if all the individuals search in small area at this time, but easy to trap into local optimum and lead to premature. From the curve's small sharp fluctuations we can see that some individuals in the QCSGSO's population

make a wide range of search beyond the 'optimal area' and meanwhile others make a small range of tuning within the 'optimal area'. This point fully shows that QCSGSO has strong ability to jump out of local optimum while maintaining high precision.

From view of the diversity comparison results, QCSGSO owns high global search ability and convergence speed. Meanwhile, it could effectively avoid precocious.

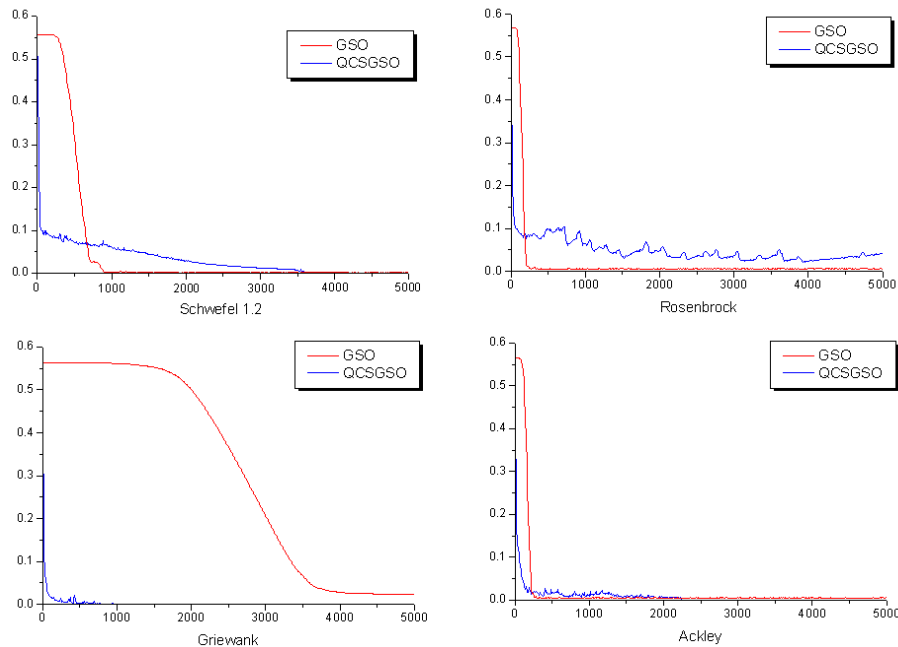


Figure 3. Diversity Compare

5. Conclusion

On the basis of the research about GSO, a Quantum Glowworm Swarm Optimization Algorithm based on Chaotic Sequence (QCSGSO) was proposed in this paper. Chaotic sequence was introduced to initialize the population to make it has more chance to cover more optimal areas, and provided good precondition to obtain high quality and precision solution. The local optimal individual of standard GSO keeps still so that its global search ability is poor. To overcome this problem, in QCSGSO, Delta potential well model of elite population was constructed in quantum space and Monte Carlo stochastic sampling method was used to make the individuals could appear in any position of solution space. At the same time, the improved algorithm could effectively avoid the prematurity and had strong ability to jump out of local optima. Finally, single dimension swimming mode was put forward to further enhance the solution precision and has solved the problem of standard GSO that sensitive to step-size, indirectly improved the robustness of the algorithm. In order to verify the feasibility and the performance of the proposed algorithm, the proposed QCSGSO was compared with some existing methods on 10 well-known uni-modal/multi-modal benchmark functions. The simulation results show that the optimization ability, solution precision and convergence speed of QCSGSO are significantly better than contrast algorithms. Our future work will apply the proposed algorithm to solve more complex and high-dimensional engineering problems.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grant No. 91220301 and 61371040.

References

- [1] K. N.Krishnanand and D.Ghose,“Detection of multiple source locations using a glowworm metaphor with applications to collective robotics”, Swarm Intelligence Symposium, Proceedings 2005 IEEE, (2005), pp. 84-91.
- [2] K. N. Krishnanand and D. Ghose,“Theoretical foundations for rendezvous of glowworm-inspired agent swarms at multiple locations”, Robotics and Autonomous Systems, vol. 56, no. 7, (2008), pp. 549-569.
- [3] K. N.Krishnanand and D. Ghose,“A glowworm swarm optimization based multi-robot system for signal source localization”, Design and Control of Intelligent Robotic Systems. Springer Berlin Heidelberg, (2009), pp. 49-68.
- [4] K. N.Krishnanand and D. Ghose,“Glowworm swarm optimisation: a new method for optimising multi-modal functions”, International Journal of Computational Intelligence Studies, vol. 1, no. 1,(2009), pp. 93-119.
- [5] W. H. Liao, Y. Kao, Y. S. Li,“A sensor deployment approach using glowworm swarm optimization algorithm in wireless sensor networks”, Expert Systems with Applications, vol. 38, no. 10,(2011), pp. 12180-12188.
- [6] K. N.Krishnanand and D.Ghose,“Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions”, Swarm intelligence, 2009, 3(2): 87-124.
- [7] D. H.Wolpert and W. G. Macready,“No free lunch theorems for optimization”, Evolutionary Computation, IEEE Transactions on, vol. 1, no. 1, (1997),pp. 67-82.
- [8] B. Wu, C. Qian, W. Ni, *et al.*,“The improvement of glowworm swarm optimization for continuous optimization problems”, Expert Systems with Applications, vol. 39, no. 7, (2012),pp. 6335-6342.
- [9] L.He, X.Tong and S.Huang,“A Glowworm Swarm Optimization Algorithm With improved movement rule”, Intelligent Networks and Intelligent Systems (ICINIS), 2012 Fifth International Conference on. IEEE, (2012),pp. 109-112.
- [10] Y. L. Zhang, X. P.Ma, Y.Gu, *et al.*,“A modified glowworm swarm optimization for multimodal functions”, Control and Decision Conference (CCDC), 2011 Chinese. IEEE, (2011), pp. 2070-2075.
- [11] Y. Zhou, Y. Wang, S. He, *et al.*,“A Novel Double Glowworm Swarm Co-Evolution Optimization Algorithm based Levy Flights”, Appl. Math, vol. 8, no. 1, (2014), pp. 355-361.
- [12] X. S. Yang and S.Deb,“Cuckoo search via Lévy flights”, Nature & Biologically Inspired Computing, 2009.NaBIC 2009.World Congress on. IEEE, (2009), pp. 210-214.
- [13] Y.-Q Zhou, Z.-X. Huang, H.-X. Liu, “Discrete Glowworm Swarm Optimization Algorithm for TSP Problem”,Acta Electronic Sinica, vol. 40, no. 6,(2012), pp. 1164-1170.
- [14] J. Sun, B. Feng and W.Xu,“Particle swarm optimization with particles having quantum behavior”, Evolutionary Computation, 2004. CEC2004. Congress on. IEEE, vol. 1,(2004), pp. 325-331.
- [15] M.Clerc and J.Kennedy,“The particle swarm-explosion, stability, and convergence in a multidimensional complex space”, Evolutionary Computation, IEEE Transactions on, (2002), vol. 6, no. 1, pp. 58-73.
- [16] J. H.Holland,“Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence”, U Michigan Press, (1975).
- [17] J.Kennedy,“Particle swarm optimization”, Encyclopedia of Machine Learning, Springer US, (2010),pp. 760-766.
- [18] D. Karaboga and B.Akay, “A comparative study of artificial bee colony algorithm[J]. Applied Mathematics and Computation, vol. 214, no. 1, (2009), pp. 108-132.
- [19] J.Sun, W.Fang, V. Palade, *et al.*, “Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point”, Applied Mathematics and Computation, vol. 218, no. 7, (2011), pp. 3763-3775.

Author



Du Pengzhen received the B.S. degrees in Nanjing University of Science and Technology, Nanjing, China, in 2005. He is currently a Ph.D. student in College of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, from 2006. His research interests include robot and intelligent optimization.

