Research Article

Multiscale Quantum Harmonic Oscillator Algorithm for Multimodal Optimization

Peng Wang,¹ Kun Cheng,^{2,3} Yan Huang¹,^{4,5,6} Bo Li,^{2,3} Xinggui Ye,^{2,3} and Xiuhong Chen⁶

¹School of Computer Science and Technology, Southwest University for Nationality, Chengdu, China

²Chengdu Institution of Computer Application, China Academy of Science, Chengdu, China

³University of Chinese Academy of Sciences, Beijing, China

⁴School of Computer Science and Technology, Huaiyin Normal University, Huaian, China

⁵School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China

⁶Jiangsu Key Laboratory of Media Design and Software Technology, Jiangnan University, Wuxi, China

Correspondence should be addressed to Yan Huang; hep128@qq.com

Received 2 February 2018; Accepted 3 April 2018; Published 13 May 2018

Academic Editor: Massimo Panella

Copyright © 2018 Peng Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper presents a variant of multiscale quantum harmonic oscillator algorithm for multimodal optimization named MQHOA-MMO. MQHOA-MMO has only two main iterative processes: quantum harmonic oscillator process and multiscale process. In the two iterations, MQHOA-MMO only does one thing: sampling according to the wave function at different scales. A set of benchmark test functions including some challenging functions are used to test the performance of MQHOA-MMO. Experimental results demonstrate good performance of MQHOA-MMO in solving multimodal function optimization problems. For the 12 test functions, all of the global peaks can be found without being trapped in a local optimum, and MQHOA-MMO converges within 10 iterations.

1. Introduction

Many real-world optimization problems are multimodal optimization problems, such as classification problems in machine learning [1] and inversion of teleseismic waves [2]. Multimodal optimization problems always contain several high quality global or local solutions which have to be identified and the most appropriate solution should be chosen. Global optimization of a continuous multimodal function aims at finding its several global minima or the most appropriate solution, without being trapped in a local optimum. When facing complex multimodal optimization problems, traditional optimization methods, such as gradient descent, quasi-Newton method, and Nelder-Mead's simplex methods, which may exploit all local information in an effective way, can easily be trapped into the local optimum. If a point-by-point classical optimization approach is used for this task, the approach must have to be applied several times, each time hoping to find a different optimal solution. There are two main reasons for us to find such optima as many as

possible. Firstly, an optimal solution currently favorable in the future may not remain to be so. With the knowledge of another optimal solution for the problem, users can simply switch to this new optimal solution when such a predicament occurs. Secondly, the sheer knowledge of multiple optimal solutions in the search space may provide useful insights to the properties of optimal solutions of the problem. Evolutionary algorithms (EAs) and particle swarm optimization (PSO) are used to tackle multimodal optimization problems.

Due to the population-based approach, EAs have natural advantage over classical optimization techniques. EAs maintain a population of candidate solutions, which are processed in every generation. If several distinct solutions can be preserved over all these generations, we will get multiple good solutions, rather than the only best solution. In recent years, there are several attempts to improve EAs so as to deal with multimodal fitness landscapes. Niching methods are widely used in genetic algorithms (GA), differential evolution (DE), and other evolutionary algorithms for multimodal optimization [1, 3–16].

Similar to EAs, PSO is also an iterative, population-based optimization technique. The principle of PSO is that each particle has learning ability. It can learn from itself (pbest) and its best neighbor (gbest). According to the learning approaches of particles, PSO can be divided into two models. One is the global model, the other one is the local model. In the local PSO model, each particle learns from the best particle in its neighborhood while in the global model every particle learns from the best particle in the whole population. To ensure different particles in the population converge into different optima in the solution space, the way of choosing neighborhood topology structure is crucial. This property leads to the application of PSO for multimodal optimization problems in recent years [17, 18]. Owing to PSO's features of easy-to-implement and robust adaptability, the PSO converges quickly. But once it gets stuck into the local optimum, it will be very difficult to get out from the local optimum. To overcome this problem, quantum theories are introduced into PSO system. Quantum behaved Particle Swarm Optimization (QPSO) is the quantum model of a PSO. In QPSO, individual particles have quantum behavior [19, 20]. Instead of position and velocity, wavefunction $\psi(x, t)$ [21, 22] is used to depict the state of a particle in QPSO [23]. Though QPSO performs better in global optimization than standard PSO, it also has the problem of premature convergence.

A novel optimization algorithm named multiscale quantum harmonic oscillator algorithm (MQHOA) is proposed in 2013 [24]. The population parameter and sampling parameter are researched in [24]. The uncertainty principle, zero energy and quantum tunnel effect of MQHOA are researched in [25]. MQHOA was inspired by the wavefunction of quantum harmonic oscillator. It transforms the optimization problems to find the low energy state of potential $V(x) = f(x)$. The complex objective function's second order Taylor approximation is Harmonic oscillator potential. According to quantum theory, the wavefunction of quantum harmonic oscillator represents the distribution of optimal solution. Different spring coefficients in quantum harmonic oscillator correspond with different search scales. Different spring coefficients vary inversely with search scales.

MQHOA's structure is elegant and pithy. It only includes two iteration processes: Quantum harmonic oscillator process (QHO process) and multiscale process (M process). The goal of optimization problem is searching the lowest energy position $f(x_{\text{best}})$ (where x_{best} is global minimum position). QHO process simulates the quantum harmonic oscillator annealing from high energy level to ground state. In M process, MQHOA chooses σ decreasing with a series of 1/2 to get an increased series spring coefficient. With the same σ , in QHO process, MQHOA defines a new wavefunction to get sufficient sampling points in the global optimal area. The new wavefunction is defined as the summation of Gaussian probability-density functions. MQHOA's wavefunction in scale σ_s is the sum of k Gaussian probability-density functions which take k_i as centers. It depicts the probability distribution of optimal solutions in domain. The equation can be written as

$$\psi^s(x) = \sum_{i=1}^k \frac{1}{\sqrt{2\pi}\sigma_s} e^{-(x-k_i)^2/2\sigma_s^2}. \quad (1)$$

The experimental results of 15 typical two-dimensional test functions show that MQHOA performs well in finding global optima [24].

In this paper, we present a variant of MQHOA for multimodal optimization named MQHOA-MMO. Similar to PSO's local version, in the proposed MQHOA-MMO, for each scale, every sampling point just needs to compare with the sampling points which are of the same Gaussian distribution.

This paper is organized as follows. Section 2 describe the framework of MQHOA-MMO. Test functions and comparison algorithms are presented in Section 3. The results of experiments are discussed in Section 4. Finally, Section 5 concludes the paper.

2. The Framework of MQHOA-MMO

This section presents the framework of MQHOA-MMO. We define the symbols as follows:

- (i) k is the number of swarms and Gaussian distributions.
- (ii) m is the number of sampling points of each Gaussian distribution.
- (iii) $\sigma_{\min}$ is the accuracy of optimization.
- (iv) σ is the standard deviation for all x_i .
- (v) σ_k is the standard deviation for all new x_i .
- (vi) $\Delta\sigma$ is the absolute value of the difference between σ and σ_k .
- (vii) σ_s is the current scale for iteration, the initial value is defined as the domain length.
- (viii) $S = x_1, \dots, x_i, \dots, x_k$ is the swarm of k particles such that x_i indicates particle i ($i = 1, 2, 3, \dots, k$). x_i is randomly generated in domain. For every x_i , generate m x_i^q ($q = 1, 2, 3, \dots, m$), which are following probability distribution $N(x_i, \sigma_s^2)$. $k \times m$ sampling positions are needed by every iteration. k optimal positions are stored in x_i .
- (ix) x_i^{best} is the optimal position selected from m sampling positions x_i^q .
- (x) x'_{best} is the optimal position the algorithm has found.

MQHOA-MMO includes just two nested iteration processes: QHO process and M process. In MQHOA-MMO, the QHO process is nested inside the M process. The convergence conditions of QHO process and M process are $\Delta\sigma < \sigma_s$ and $\sigma_s \leq \sigma_{\min}$ respectively. The framework of MQHOA-MMO is described in Algorithm 1.

The elaborate interpretation of the framework of MQHOAMMO is as follows:

- (1) Initialize $\sigma_{\min} = 0.00001$, $d_{\min} = -10$, $d_{\max} = 10$. So $\sigma_s = 20$. Here, we choose $k = 20$, $m = 200$, the influence of the value k will be discussed in Section 4.1.
- (2) Randomly generate x_i ($i = 1, \dots, 20$) in $[-10, 10]$. Calculate the standard deviation σ for all x_i .

```

begin
(1) initialize:  $k, m, \sigma_{\min}, \sigma_s = d_{\max} - d_{\min}$ 
(2) randomly generate  $x_i$  ( $i = 1, \dots, k$ ) in domain
(3) calculate the standard deviation  $\sigma$  for all  $x_i$ 
(4) do
(5)   do
(6)      $\forall x_i$ , generate  $x_i^q$  ( $q = 1, \dots, m$ )  $\sim N(x_i, \sigma_s^2)$ 
(7)      $\forall x_i$ , select  $x_i^{\text{best}}$  from all  $x_i^q$ 
(8)     update  $x_i: x_i \leftarrow x_i^{\text{best}}$ 
(9)     calculate the standard deviation  $\sigma_k$  for all new  $x_i$ 
(10)     $\Delta\sigma = |\sigma_k - \sigma|$ 
(11)    update  $\sigma: \sigma \leftarrow \sigma_k$ 
(12)    while: ( $\Delta\sigma > \sigma_s$ )
(13) update  $\sigma_s: \sigma_s \leftarrow \sigma_s/2$ 
(14) while ( $\sigma_s > \sigma_{\min}$ )
(15) output  $x'_{\text{best}}, f_{\text{best}}$ 
end

```

ALGORITHM 1: The framework of MQHOA-MMO.

- (3) For each x_i , generate x_i^q ($q = 1, \dots, 200$), which are following probability distribution $N(x_i, \sigma_s^2)$.
- (4) Choose the optimal position x_i^{best} from all x_i^q ($q = 1, \dots, 200$) for each x_i .
- (5) For each x_i , $x_i = x_i^{\text{best}}$. Calculate the standard deviation σ_k for all new x_i .
- (6) Calculate $\Delta\sigma = |\sigma_k - \sigma|$.
- (7) Compare $\Delta\sigma$ and σ_s . If $\Delta\sigma > \sigma_s$, return to step (3). If $\Delta\sigma < \sigma_s$, $\sigma_s = \sigma_s/2$.
- (8) Compare $\sigma_{\min}$ and σ_s . If $\sigma_s > \sigma_{\min}$, return to step (3). If $\sigma_s < \sigma_{\min}$, return $x'_{\text{best}}, f_{\text{best}}$.

According to the framework above, only two parameters (k and m) need to be set. The selection of k is discussed in Section 4.1.

In framework the superposition of k Gauss sampling areas constructs the wavefunction. Wavefunction written as equation (1) depicts the probability distribution of optimal solutions in domain. The changes of wavefunction in iterations are showed in Figure 5. In order to reduce the energy of system, k optimal positions x_i^{best} are retained from $k \times m$ sampling positions. In QHO process, $\sigma_s = \sigma_s/2$ transforms the system from high energy state at scale σ_s to ground state at scale $\sigma_s/2$.

For high dimensional test functions, MQHOA-MMO can use two-dimensional array x_{ij} to store the k high dimensional central positions of Gauss sampling area ($N(x_{ij}, \sigma_s)$). Where i is dimension, j is the number of Gauss sampling areas. For every dimension, MQHOA-MMO calculates the value of σ_k and $\Delta\sigma$. The QHO process at scale σ_s will end until $\Delta\sigma \leq \sigma_s$ in every dimension.

3. Experimental Setup

In this section, we give a brief description of benchmark functions and comparison algorithms. Experimental setup is present at the end of this section.

3.1. Test Functions. The benchmark functions we choose are widely used in multimodal optimization. Some test functions have various characteristics, such as irregular landscape, symmetric or equal distribution of optima. The goals are thus to evaluate the ability to tackle a complicated problem, to validate its capacity to detect all the global peaks of a function. A brief description of the functions is listed in Table 1.

3.2. Comparasion Algorithms. To evaluate the performance of MQHAO-MMO, MQHAO-MMO is compared with the following standard multimodal evolutionary algorithms. MQHOA-MMO is marked as AL0.

- AL1 (IWO- σ -GSO [26]): the vasion weed optimization- σ -group search optimizers;
- AL2 (scma [27]): CAM-ES with Self-Adaptive Niche Radius;
- AL3 (cde [28]): the original crowding DE;
- AL4 (sde [29]): speciation-based DE;
- AL5 (ferpso [30]): fitness-Euclidean distance Ratio;
- AL6 (spso [31]): speciation-based PSO;
- AL7 (r2pso [31]): a lbest PSO with a ring topology, each member interacts with only its immediate member to its right;
- AL8 (r3pso [31]): a lbest PSO with a ring topology, each member interacts with only its immediate member to its left;
- AL9 (r2psolhc): the same as r2pso [32], but with no overlapping neighborhoods;
- AL10 (r3psolhc): the same as r3pso [32], but with no overlapping neighborhoods.

3.3. Experimental Environment and Criteria. MQHOA-MMO is coded in Matlab R2014 and the simulations are run on i5 CPU 2.9 GHz with 8 GB memory. Results are averaged over 30 independent runs. If the difference between a computed solution and a known global optimum is less than ϵ , the peak is considered to be found. The performance of all multimodal algorithms is measured in terms of the following two criteria:

- (1) Success rate: The percentage of runs in which an algorithm can detect all the global peaks.
- (2) Average peak number: Peak number found over 30 runs for each function.

4. Experimental Studies

The experimental studies and analyses are presented in this section. MQHOA-MMO had been run until $\sigma_s < \sigma_{\min}$ or the maximum number of function evaluation was exhausted.

4.1. Parameter Experiment. In this section, we examine the effectiveness and efficiency of MQHOA-MMO by applying it to selected four benchmark functions: Six-Hump Camel Back

TABLE 1: Benchmark function.

Function name	D	Range	Benchmark function	Optima
E1-F1: Equal maxima	1	$x \in [0 \quad 1]$	$F1(x) = \sin^6(5\pi x)$	5
E1-F2: Uneven maxima	1	$x \in [0 \quad 1]$	$F2(x) = \sin^6(5\pi(x^{3/4} - 0.05))$	5
E1-F3: Himmelblau's	2	$x \in [-4 \quad 4]$ $y \in [-4 \quad 4]$	$F3(x, y) = 200 - (x^2 + y - 11)^2 - (x + y^2 - 7)^2$	4
E1-F4: Six-Hump Camel	2	$x \in [-1.9 \quad 1.9]$ $y \in [-1.1 \quad 1.1]$	$F4(x, y) = \left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + xy + (-4 + 4y^2)y^2$	2
E1-F5: Shekel's foxholes	2	$x \in [-65.54 \quad 65.54]$ $y \in [-65.54 \quad 65.54]$	$F5(x, y) = 500 - \frac{1}{0.002 + \sum_{i=0}^{24} \left(\frac{1}{1+i+(x-a(i))^6 + (y-b(i))^6} \right)}$	1
E1-F6: Branin RCOS	2	$x \in [-5 \quad 10]$ $y \in [0 \quad 15]$	$F6(x, y) = \left(y - \frac{5.1}{4\pi^2}x^2 + \frac{x}{\pi} - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x) + 10$	3
E1-F7: Root function	2	$x = x_1 + ix_2$ $x \in [-2 \quad 2]$	$F7(x) = \frac{1}{1 + x^6 - 1 }$	6
E1-F8: Hansen	2	$x \in [-10 \quad 10]$ $y \in [-10 \quad 10]$	$F8(x, y) = \sum_{i=1}^5 i \cos((i-1)x - i) \cdot \sum_{j=1}^5 j \cos((j+1)y + j)$	9
E1-F9: Holder Table	2	$x \in [-10 \quad 10]$	$F9(x, y) = -\left \sin(x_1) \cos(x_2) \cdot \exp\left(1 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right) \right $	4
E1-F10: Rescaled Six-Hump Camel	2	$x \in [-1.9 \quad 1.9]$ $x \in [-1.1 \quad 1.1]$	$F10(x, y) = -\left(\left(4 - 2.1x^2 + \frac{x^4}{3}\right)x^2 + 10xy + (-4 + 4(10y)^4)\right)$	2
E1-F11: 2D Inverted Shubert	2	$x_1 \in [-10 \quad 10]$ $x_2 \in [-10 \quad 10]$	$F11(\vec{x}) = -\prod_{i=1}^D \sum_{j=1}^5 j \cos[(j+1)x_i + j]$	18
E1-F12: Cross-In-Tray	2	$X_i \in [-10 \quad 10]$	$F12(x) = -0.0001 \left(\left \sin(x_1) \sin(x_2) \exp\left(100 - \frac{\sqrt{x_1^2 + x_2^2}}{\pi}\right) + 1 \right \right)^{0.1}$	4

FIGURE 1: Relation of k and optimal solution with $m = 200$, $\sigma_{\min} = 10e - 6$, and repeat time = 30.

Function (E1-F4), Himmelblaus's Function (E1-F3), Hansen Function (E1-F8) and 2D Inverted Shubert Function (E1-F18). The numbers of solution individuals are 2, 4, 9, and 18, respectively.

In MQHOA-MMO, we generate initial population using two parameters: k and m . We choose different initial parameter k to test the impact on the ability of finding global optima. We run over 30 times for each k while $m = 200$. Figure 1 contains four relation figures between the parameter k and the number of global optima which MQHOA-MMO can find. The initial k value begins at 5 and increases by 5 each time. According to Figures 1(c) and 1(d), while k is smaller than

the number of global optima, MQHOA-MMO can only find part of global optima. With the increasing of k , we can find most global optima when the initial k is close to the number of global optima. When the number of optimal solutions is small, for example, less than 10, the increased k can find all the optimal solutions. But when the number of global optima is larger (more than 10), with the increase of k , the number of global optima which MQHOA-MMO can find also increases. When k increases to a certain number, the number of global optima which MQHOA-MMO can find is stable. When the domain is large, k should increase in a large scale more than ten times. We can make a conclusion that to find all the

FIGURE 2: Convergence of $F1-F4$, where $k = 50$, $m = 200$, $\sigma_{\min} = 10e - 6$, repeat time = 30.

global optima the initial value of k should be larger than the number of the global optima. Besides, along with the increase of domain, the number of k should increase correspondingly.

4.2. Convergence Experiment. To be more intuitive, we apply 12 benchmark test functions to demonstrate the convergence and verify the effectiveness of MQHOA-MMO. Figures 2, 3, and 4 show the relationship between fitness and iteration times. Fitness is the average of k optimum function values of test functions. The value of iteration times is the number of QHO processes. For the 12 benchmark test functions, we set parameters as follows: $k = 50$, $m = 200$, and $\sigma_{\min} = 10e - 6$. Results are averaged over 30 independent runs.

From Figures 2–4, we find that almost all the functions have converged to several small areas before the tenth iteration. Some functions, such as $F1$, $F2$, $F3$, $F5$, $F6$, $F9$, and $F12$, can converge to several small areas at the fifth iteration. Table 2 presents the iteration times of 12 test functions while $\sigma_s < \sigma_{\min}$ or the maximum number of function evaluations was exhausted. The results mean that MQHOA-MMO has a fast convergent ability, and for most test functions there is only one QHO process in each M process.

4.3. Changes of Wavefunction. In this section, we choose Hansen function (E1-F8) to present the changes of wavefunction in MQHOA-MMO. We set $k = 35$, $m = 200$,

$d_{\min} = -10$, and $d_{\max} = 10$. The function is written as follows:

$$F(x, y) = \sum_{i=1}^5 i \cos((i-1)x - i) \sum_{j=1}^5 j \cos((j+1)y + j). \quad (2)$$

MQHOA-MMO's wavefunction is written as (1). To describe wavefunction clearly, we have defined three notions. First notion is incipient centers of k swarms, which are x_i used in QHO's first iteration for each σ_s . Second notion is incipient wavefunction, which is the wavefunction of x_i used in QHO's first iteration for each σ_s . The last notion is last wavefunction, which is the wavefunction of x_i used in QHO's last iteration for each σ_s . After the last QHO iteration for each σ_s , σ_s will be cut half to $\sigma_s/2$.

Figure 5 presents the change of wavefunction in iterations. Figures 5(a)–5(d) show the incipient centers of k swarms with different σ_s . Figures 5(e)–5(h) show the incipient wavefunctions with different σ_s . Figures 5(i)–5(l) show the last wavefunctions with different σ_s . With $\sigma_s = 20$, Figure 5(a) shows the incipient centers of swarms and Figure 5(e) shows the incipient wavefunction while the last wavefunction is shown in Figure 5(i). With $\sigma_s = 2.5$, Figure 5(b) shows the incipient centers of swarms, Figure 5(f) shows the incipient wavefunction, and the last wavefunction is shown in Figure 5(j). With $\sigma_s = 1.25$, Figure 5(c) shows the incipient

FIGURE 3: Convergence of F5–F8, where $k = 50$, $m = 200$, $\sigma_{\min} = 10e - 6$, repeat time = 30.

FIGURE 4: Convergence of F9–F11, where $k = 50$, $m = 200$, $\sigma_{\min} = 10e - 6$, repeat time = 30.

TABLE 2: Iteration times (N) for 12 test functions, where $k = 50$, $m = 200$, $\sigma_{\min} = 10e - 6$, $\sigma_s = d_{\max} - d_{\min}$, and repeat time = 30.

$F1$	$F2$	$F3$	$F4$	$F5$	$F6$	$F7$	$F8$	$F9$	$F10$	$F11$	$F12$
20	20	20	20	27	24	22	25	25	22	25	25

TABLE 3: Success rates for test functions; the parameters for MQHOA-MMO are as follows: $k = 50$, $m = 200$, $\sigma_{\min} = 10e - 6$, and repeat time = 30; the domain of definition is different with different functions.

Func	ϵ	AL0	AL1	AL2	AL3	AL4	AL5	AL6	AL7	AL8	AL9	AL10
$F1$	0.000001	100	100	92	28	72	84	88	92	88	100	92
$F2$	0.000001	100	100	88	28	60	100	92	88	72	92	92
$F3$	0.0005	100	88	72	0	72	72	0	24	28	24	24
$F4$	0.000001	100	60	60	0	100	96	0	60	56	52	60
$F5$	0.00001	100	100	88	52	32	100	56	88	76	72	60
$F6$	0.1	100	92	74	82	78	76	70	72	66	60	62

TABLE 4: Average number of peaks found for the test functions; the parameters for MQHOA-MMO are as follows: $k = 50$, $m = 200$, $\sigma_{\min} = 10e - 6$, and repeat time = 30, and the domain of definition is different with different functions.

Func	ϵ	AL0	AL1	AL2	AL3	AL4	AL5	AL6	AL7	AL8	AL9	AL10
$F1$	0.000001	5	5	4.92	3.84	4.72	4.84	4.88	4.92	4.88	5	4.92
$F2$	0.000001	5	5	4.88	3.96	4.6	5	4.92	4.88	4.72	4.92	4.88
$F3$	0.0005	4	3.88	3.72	0.32	3.72	3.68	0.84	2.92	2.76	3	3.12
$F4$	0.000001	2	1.6	1.6	0.04	2	1.96	0.08	1.44	1.56	1.56	1.48
$F5$	0.00001	1	1	0.88	0.52	0.32	1	0.56	0.88	0.76	0.72	0.6
$F6$	0.1	3	2.88	2.56	2.76	2.72	2.64	2.48	2.52	2.44	2.36	2.40

centers of swarms, Figure 5(g) shows the incipient wavefunction, and the last wavefunction is shown in Figure 5(k). With $\sigma_s = 0.3125$, Figure 5(d) shows the incipient centers of swarms, Figure 5(h) shows the incipient wavefunction, and the last wavefunction is shown in Figure 5(l).

Figures of the same σ_s show the changes of wavefunctions in different QHO iterations. For example, Figures 5(g) and 5(k) show the changes in QHO iterations with $\sigma_s = 1.25$. As mentioned in Section 4.2 the number of QHO iterations is small in each M iteration, so the change of wavefunctions is not obvious in different QHO iterations with the same σ_s . Figures of different σ_s give us the differences of wavefunctions in different M iterations. For example, Figures 5(i), 5(j), 5(k), and 5(l) show the changes of wavefunctions in M iterations. According to our definition, Figures 5(j) and 5(g) have the same centers of k swarms and the different σ_s . From the wavefunctions, we also can find that if there is a large probability of the optimal solution at a certain point, the point will be more attracted in this area, according to Gauss sampling law. It means that the higher probability is that the optimal solution is in the region. With the decrease of σ_s , the probability distribution of particles is more and more concentrated. This is the principle that MQHOA-MMO uses multiscale to implement the precision of the algorithm.

4.4. Comparison Experiments. In this section, we will present a detailed discussion on the performance of various algorithms that were chosen in the comparative study. We use 6 challenging functions of various characteristics to evaluate the MQHOA-MMO's performance. MQHOA-MMO runs

until $\sigma_s < \sigma_{\min}$ or the maximum number of function evaluations was exhausted. The experimental results of other algorithms are quoted from [26]. All performances of MQHOA-MMO are calculated and averaged over 30 independent runs with $k = 50$, $m = 200$, and $\sigma_{\min} = 10e - 6$. From Table 3, we can find that MQHOA-MMO's success rate can reach up to 100%. For some complex test functions, such as $F3$, $F5$, and $F6$, MQHOA-MMO can also get 100% success rate while some other algorithms even cannot find the global optima.

Table 4 shows the average number of global peaks detected by the MQHOA-MMO and other ten evolutionary multimodal optimization algorithms on test functions. Table 4 further indicates that MQHOA-MMO is able to detect the global optima in the test cases, and MQHOA-MMO can yield a good level of accuracy. IWO- σ -GSO is a new excellent combination algorithm. It can generate acceptable results over the test functions. SCMA can also generate good results over simple and low dimension. Their performance gradually becomes poor when the dimension increases. SDE algorithm shows very poor performance if the number of peaks has high accuracy. Other SDE could not be able to generate satisfactory solution. FERPSO is able to generate relatively satisfactory results in many test functions. For MQHOA-MMO, given suitable parameters, it can get all global optima.

5. Conclusion

In this paper, we proposed a multimodal optimization algorithm named MQHOA-MMO. We used wave function to locate the possibility positions of the optimal solutions. The

FIGURE 5: Continued.

FIGURE 5: Changes of wavefunction in iterations.

experimental study undertook 12 distinct test functions with number of global peaks varying from 2 to 25. Through comparison of results which were obtained from optimization of several benchmark functions using MQHOA-MMO and other optimization algorithms with two criteria and results obtained from performance experiments it is revealed that MQHOA-MMO can detect all the global optimum in a fast effective, controllable, and higher accuracy. Furthermore, this algorithm can find the global optimum of multifunction without being trapped in local optimum. The experimental study clearly indicated that, in most of the test cases, performance of MQHOA-MMO remains statistically better than all the other algorithms compared with it. For some complex functions, MQHOA-MMO could not have a good success rate. Future research of MQHOA-MMO will be focused on the optimization of such complex functions and higher-dimensional functions.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (60702075), Fundamental Research Funds for the Central Universities of China (2017NZYQN27), Science and Technology Planning Project of Guangdong Province, China (2016B090918062), and Jiangsu Key Laboratory of Media Design and Software Technology Foundation.

References

- [1] S. W. Mahfoud, "Nicheing methods for genetic algorithms," *Urbana*, vol. 51, no. 95001, pp. 62–94, 1995.
- [2] K. D. Koper, M. E. Wyssession, and D. A. Wiens, "Multimodal function optimization with a niching genetic algorithm: A seismological example," *Bulletin of the Seismological Society of America*, vol. 89, no. 4, pp. 978–988, 1999.
- [3] A. Della Cioppa, C. De Stefano, and A. Marcelli, "Where are the niches? Dynamic fitness sharing," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 4, pp. 453–465, 2007.
- [4] S. Dasa, S. Maity, B.-Y. Qu, and P. N. Suganthan, "Real-parameter evolutionary multimodal optimization-A survey of the state-of-the-art," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 71–78, 2011.
- [5] B. Sareni and L. Krähenbühl, "Fitness sharing and niching methods revisited," *IEEE Transactions on Evolutionary Computation*, vol. 2, no. 3, pp. 97–106, 1998.
- [6] G. Singh and K. Deb, "Comparison of multi-modal optimization algorithms based on evolutionary algorithms," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference 2006*, pp. 1305–1312, USA, July 2006.
- [7] K. A. D. Jong, *An analysis of the behavior of a class of genetic adaptive systems [Ph.D. thesis]*, 1975.
- [8] D. E. Goldberg and J. Richardson, "Genetic algorithms with sharing for multimodal function optimization," in *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*, pp. 41–49, Lawrence Erlbaum, Hillsdale, NJ, 1987.
- [9] G. R. Harik, "Finding multimodal solutions using restricted tournament selection," *ICGA*, pp. 24–31, 1995.
- [10] J.-P. Li, M. E. Balazs, G. T. Parks, and P. J. Clarkson, "A species conserving genetic algorithm for multimodal function optimization," *Evolutionary Computation*, vol. 10, no. 3, pp. 207–234, 2002.
- [11] S. W. Mahfoud, "Crowding and preselection revisited," *Urbana*, vol. 51, Article ID 61801, 1992.
- [12] D. Beasley, D. R. Bull, and R. R. Martin, "A Sequential Niche Technique for Multimodal Function Optimization," *Evolutionary Computation*, vol. 1, no. 2, pp. 101–125, 1993.
- [13] M. Bessao, A. Pérowski, and P. Siarry, "Island Model Cooperating with Speciation for Multimodal Optimization," in *Parallel Problem Solving from Nature PPSN VI*, vol. 1917 of *Lecture Notes in Computer Science*, pp. 437–446, Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
- [14] K. E. Parsopoulos and M. N. Vrahatis, "On the computation of all global minimizers through particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 211–224, 2004.
- [15] X. Yin and N. Gernay, "A fast genetic algorithm with sharing scheme using cluster analysis methods in multimodal function optimization," in *Artificial neural nets and genetic algorithms*, pp. 450–457, Springer, 1993.
- [16] A. Petrowski, "Clearing procedure as a niching method for genetic algorithms," in *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation, ICEC'96*, pp. 798–803, May 1996.
- [17] Y.-T. Juang, S.-L. Tung, and H.-C. Chiu, "Adaptive fuzzy particle swarm optimization for global optimization of multimodal functions," *Information Sciences*, vol. 181, no. 20, pp. 4539–4549, 2011.
- [18] L. Liu, S. Yang, and D. Wang, "Force-imitated particle swarm optimization using the near-neighbor effect for locating multiple optima," *Information Sciences*, vol. 182, pp. 139–155, 2012.
- [19] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the Congress on Evolutionary Computation (CEC '04)*, pp. 325–331, Portland, Ore, USA, June 2004.
- [20] J. Sun, W. Xu, and B. Feng, "Adaptive parameter control for quantum-behaved particle swarm optimization on individual level," in *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, pp. 3049–3054, October 2005.
- [21] W. Schweizer, *Numerical quantum dynamics*, vol. 9, Springer Science & Business Media, 2001.
- [22] F. S. Levin, *An introduction to quantum theory*, Cambridge University Press, Cambridge, 2002.
- [23] L. D. S. Coelho, "A quantum particle swarm optimizer with chaotic mutation operator," *Chaos, Solitons & Fractals*, vol. 37, no. 5, pp. 1409–1418, 2008.
- [24] P. Wang, Y. Huang, C. Ren, and Y. M. Guo, "Multi-scale quantum harmonic oscillator for high-dimensional function global optimization algorithm," *Chinese Journal of Electronics*, vol. 41, no. 12, pp. 2468–2473, 2013.

- [25] P. Wang and Y. Huang, "Physical model of multi-scale quantum harmonic oscillator optimization algorithm," *Journal of Frontiers of Computer Science and Technology*, vol. 9, no. 10, pp. 1271–1280, 2015.
- [26] S. Roy, S. M. Islam, S. Das, and S. Ghosh, "Multimodal optimization by artificial weed colonies enhanced with localized group search optimizers," *Applied Soft Computing*, vol. 13, no. 1, pp. 27–46, 2013.
- [27] O. M. Shir, M. Emmerich, and T. Bäck, "Adaptive niche radii and niche shapes approaches for niching with the CMA-ES," *Evolutionary Computation*, vol. 18, no. 1, pp. 97–126, 2010.
- [28] D. Shen and Y. Li, "Multimodal optimization using crowding differential evolution with spatially neighbors best search," *Journal of Software*, vol. 8, no. 4, pp. 932–938, 2013.
- [29] X. Li, "Efficient differential evolution using speciation for multimodal function optimization," in *Proceedings of the GECCO 2005 - Genetic and Evolutionary Computation Conference*, pp. 873–880, usa, June 2005.
- [30] J.-H. Seo, C.-H. Im, C.-G. Heo, J.-K. Kim, H.-K. Jung, and C.-G. Lee, "Multimodal function optimization based on particle swarm optimization," *IEEE Transactions on Magnetics*, vol. 42, no. 4, pp. 1095–1098, 2006.
- [31] X. Li, "Adaptively choosing neighbourhood bests using species in a particle swarm optimizer for multimodal function optimization," *Lecture Notes in Computer Science (including sub-series Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Preface*, vol. 3102, pp. 105–116, 2004.
- [32] X. Li, "Niching without niching parameters: particle swarm optimization using a ring topology," *IEEE Transactions on Evolutionary Computation*, vol. 14, no. 1, pp. 150–169, 2010.