IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 16, 2018, accepted December 13, 2018,
date of publication January 18, 2019, date of current version February 20, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2893200

Multi-Scale Quantum Harmonic Oscillator Algorithm With Truncated Mean Stabilization Strategy for Global Numerical Optimization Problems

XINGGUI YE[1,2,3], (Senior Member, IEEE), PENG WANG[4], GANG XIN[1,2], JIN JIN[1,2], AND YAN HUANG[5]

[1]University of Chinese Academy of Sciences, Beijing 100049, China
[2]Chengdu Institution of Computer Application, Chinese Academy of Sciences, Chengdu 610041, China
[3]Cloud Computing and Big Data Center, China Unicom Fujian Branch, Fuzhou 350007, China
[4]School of Computer Science and Technology, Southwest Minzu University, Chengdu 610225, China
[5]School of Computer Science and Technology, Huaiyin Normal University, Huai'an 223300, China

Corresponding author: Peng Wang (qhoalab@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 60702075, in part by the Fundamental Research Funds for the Central Universities of China under Grant 2018NQN55, and in part by the Natural Science Foundation of Huaian under Grant HAB201828.

ABSTRACT A multi-scale quantum harmonic oscillator algorithm (MQHOA) is a quantum population-based algorithm proposed recently. It utilizes the quantum wave function to locate the global optimum of a global numerical optimization problem. As the MQHOA employs the elitism to replace the worst particle in each iteration cycle, it reduces one of the particles in each run, which will cripple the diversity of the population and slow down the convergence speed. Therefore, the particles will be easily trapped into local optima. In this paper, we suggest a new MQHOA with truncated mean stabilization (TS-MQHOA) policy to alleviate the above-mentioned problems. The theoretical and experimental analyses indicate that the truncated mean stabilization strategy helps to diversify the populations and improve the convergence efficiency. The proposed TS-MQHOA is evaluated on a number of dimensionwise unimodal and multimodal CEC benchmark functions, and the computational results are compared with several popular population-based algorithms. The experimental results on complex test problems demonstrate that the proposed TS-MQHOA, in most function evaluations, is able to obtain better convergence toward the global optimum compared with several renowned heuristic algorithms based on swarm intelligence. Meanwhile, the comparative results reveal the competitiveness and superiority of the proposed algorithm, especially on high-dimensional function evaluations.

INDEX TERMS Multi-scale quantum harmonic oscillator algorithm, swarm intelligence, population-based algorithm, stochastic algorithm, truncated mean stabilization strategy.

I. INTRODUCTION

Swarm intelligence has attracted extensive attentions for many decades. Although the intelligence is either from a living population or from an inanimate group, the swarm intelligence often generates stunning effects on tackling global optimization problems. Typically, since Eberhart and Kennedy proposed the particle swarm optimization (PSO) in 1995 [1], tens of thousand of researchers have followed them to inherit and develop the PSO technique. In recent years, thousands of algorithms have been proposed, either proposing improvements of PSO, or developing new techniques inspired by PSO. Some of these algorithms have been proved to be superior to the original PSO and become branches of swarm intelligence theory.

Typically, the algorithms based on swarm intelligence are including standard particle swarm optimization (SPSO) [2], [3], particle swarm optimization with Levy flight (LPSO) [4], comprehensive learning PSO (CLPSO) [5], quantum-behaved particles swarm optimization (QPSO)

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Ji.

[6], [7], artificial bee colony algorithm (ABC) [8], [9], ant colony optimization (ACO) [10], bat algorithm (BA) [11], fireworks algorithm (FWA) [12], [13], multi-scale quantum harmonic oscillator algorithm (MQHOA) [14], [15] and etc. Most of these algorithms consist of a decentralized population of simple agents interacting with one another locally within a given search space. The agents follow very simple rules to interact with each other without anyone dictating how to behave. The interaction between the agents leads to the emergence of swarm intelligence and global behavior.

MQHOA is a recent proposed quantum-behaved and population-based metaheuristic algorithm. The algorithm utilizes a population of particles (quantum swarm) to search for the ground energy in quantum system. The convergence process of MQHOA in function evaluation is analogized to the transformation process of particles from a high energy level to the ground energy level. Although the structure of MQHOA is concise, it is found effective and efficient to solve unimodal and multimodal problems [15], [16]. Meanwhile, it has been proved to be more effective and efficient when an individual stabilization strategy is introduced to the original MQHOA (IS-MQHOA) in the course of the function evaluation [17]. However, sometimes the arithmetic mean position applied to IS-MQHOA is too closed to the local optimum that the algorithm still can not avoid premature stagnation and time consuming. This paper proposes a new MQHOA based on truncated mean stabilization policy and some mechanisms. Theoretical and experimental analyses of the proposed TS-MQHOA are conducted in this paper. The comparative results between the TS-MQHOA and some state-of-the-art population-based algorithms reveal the competitiveness and superiority of the proposed algorithm.

The remainder of this paper is organized as follows. Section II briefly introduces the related works about the research work with the truncated mean rule. Followed by the introduction of the original MQHOA in brief in Section III. The multi-scale quantum harmonic oscillator algorithm with truncated mean stabilization strategy (TS-MQHOA) is demonstrated in Section IV. Section V elaborates the experiments and compares the computational results with several popular population-based algorithms. Finally, the conclusion and our future work are outlined in Section VI.

II. RELATED WORKS

The truncated mean rule is not only widely used to deal with statistical problems [18]–[20], but also universally applied to engineering problems, such as filter design [21], [22], hotel reservation forecasting [23], intrusion detection system development [24] and etc.

In [24], the truncated mean is suggested and applied to estimate the class mean vector in the *Linear Discriminant Analysis* modeling. The author designed a density estimator based on truncated mean and found to obtain smaller mean squared errors compared with the classical estimator when estimating the tails of gamma and normal distributions. The empirical results indicate the superiority of the proposed technique based on truncated mean rule to effectively develop an intrusion detection system based on truncated mean rule. In [25], a new method based on the truncated mean of specific energy loss is developed to apply to Cherenkov detector. The new method was found a 26% improvement in measuring muon energy. In their experiments, the author divided the muon track into several segments with separate values. It found that to eliminate the highest results in an overall energy data was more closely correlated to the real muon energy. The truncated mean mechanism employed in [26] was to develop a new filter, and it was found effectively helpful to suppress the additive and exclusive noise. The authors also found that by trimming and truncating the samples, the new filters were more efficient to attenuate the mixed additive and exclusive noise. Meanwhile, they experimentally found that the new method based on the truncated mean rule effectively saved the computational time and theoretically lowered the computational complexity.

In this paper, the truncated mean stabilization strategy is utilized in the proposed TS-MQHOA to eliminate the extreme particles which are located in local optima. Truncating the particles with the highest and the lowest fitness can theoretically reduce the probability of falling into local optima, and hence improve the search performance of the population. Meanwhile, the expansion of the search space when the algorithm stagnates for a long period is helpful to escape from local optima, to some extent. The main contribution of this paper can be summarized as follows.

Firstly, a new multi-scale quantum harmonic oscillator algorithm based on truncated mean stabilization policy is proposed for global optimization problems.

Secondly, a search space expansion mechanism is proposed, and the expansion coefficient is determined according to a large number of several empirical trials.

Thirdly, the performance of the proposed TS-MQHOA is validated to be mostly better than the IS-MQHOA in the CEC benchmark function evaluations.

Fourthly, the performance of the proposed TS-MQHOA is compared to several popular population-based algorithms, the comparative results demonstrate the superiority of TS-MQHOA in most high-dimensional function evaluations.

Fifthly, the empirical results indicate the effectiveness of truncated mean strategy on the improvement of convergence performance in TS-MQHOA, which implies a promising method to improve other population-based techniques.

III. MULTI-SCALE QUANTUM HARMONIC OSCILLATOR ALGORITHM

An optimization problem $f(x)$ in this paper is designated as follows.

$$\textit{Minimize} \quad f(x) \quad \textit{subject to } x_i \in [x_l, x_u]^D \tag{1}$$

where $f(x)$ is the objective function, x_i is the D-dimensional decision variable, x_l and x_u are the lower and upper bounds respectively.

In quantum space, every non-relativistic particle moves randomly in the electric field where there are different energy levels between high potential energy states E_i, $i = 1, 2, \ldots, n$ and the ground state (*the Zero State*) E_0. The higher of the energy level, the more active and unstable of the particles. The course of transition for particles from E_i to E_0 is a convergence process, which is similar to the converging process of an algorithm in a function evaluation. The probability of particles appears in the quantum space can be demonstrated by the time-independent Schrödinger equation as follows [27], [28].

$$E\psi(x) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x)\right)\psi(x) \quad (2)$$

(2) is an eigenvalue equation, where E is the system energy of stationary state $\psi(x)$ ($\psi(x)$ is probability amplitude) and $|\psi|^2$ designates the probability distribution of the particles in the quantum space. $\hbar = h/2\pi$ (h is the Planck constant), $V(x)$ is the potential energy and a bound in the quantum space.

Inspired by the quantum theory [29] and quantum annealing method [30], an optimization problem can be in analogy with particles from high energy states searching for the ground state under a potential well $V(x)$. Accordingly, (2) can be rewritten as the following form:

$$E\psi(x) = \left(-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + f(x)\right)\psi(x) \quad (3)$$

where the $V(x)$ in (2) is replaced by $f(x)$ in (3). As $|\psi_n(x)|^2$ implies particle distribution probability in the nth energy level in the quantum space [27], we employ it in the global numerical optimization, where the objective is transformed into searching for the minimum of an objective function $f(x)$.

According to the Taylor's expansion, an objective function $f(x)$ and the potential well $V(x)$ can be written in the following way.

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{1}{2}f''(x_0)(x - x_0)^2 + \ldots \quad (4)$$

where $f(x_0)$ is a constant.

We substitute (4) into (3) and obtain the wavefunction $\psi(x)$ as the form [28]:

$$\psi_n(x) = \sqrt{\frac{1}{2^n n!}}(\frac{m\omega}{\pi\hbar})^{\frac{1}{4}} \cdot exp(-\frac{m\omega x^2}{2\hbar}) \cdot H_n(\sqrt{\frac{m\omega}{\hbar}}x) \quad (5)$$

where n represents the nth energy level, other parameters have the same meanings as described in above-mentioned equations. Accordingly, the distributional probability density of wavefunction can be rewritten as:

$$|\psi_n(x)|^2 = \frac{1}{2^n n!}(\frac{m\omega}{\pi\hbar})^{\frac{1}{2}} \cdot exp(-\frac{m\omega x^2}{\hbar}) \cdot |H_n(\sqrt{\frac{m\omega}{\hbar}}x)|^2 \quad (6)$$

when $n \to 0$, (6) is equal to :

$$|\psi_0|^2 = (\frac{m\omega}{\pi\hbar})^{\frac{1}{2}} \cdot exp(-\frac{m\omega x^2}{\hbar}) \quad (7)$$

where (7) is a form of Gaussian equation:

$$\psi(x) = \frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{(x-\mu)^2}{2\sigma^2}) \quad (8)$$

Accordingly, (5) can be rewritten as follow:

$$\psi_n(x) = \sum_{i=1}^{n}\psi(i) = \sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}exp(-\frac{(x-\mu_i)^2}{2\sigma^2}) \quad (9)$$

where μ is the average of the optimal solutions, σ is the standard deviation of the current optimal solutions. The smaller of the σ, the narrower of the search space.

It can be seen in (6) and (7), from high energy levels to the ground state, the wavefunction of quantum harmonic oscillator changes from an intertwined n Gaussian functions in (6) to an overlapped Gaussian function in (7).

The structure of MQHOA is concise, including quantum harmonic oscillator process (QHO process) and multi-scale process (M process). In QHO process, particles explore new neighbor fields to exploit better optimal solutions. While in M process, the search domain is narrowed by half. The framework of MQHOA is depicted in Fig. 1 [15].

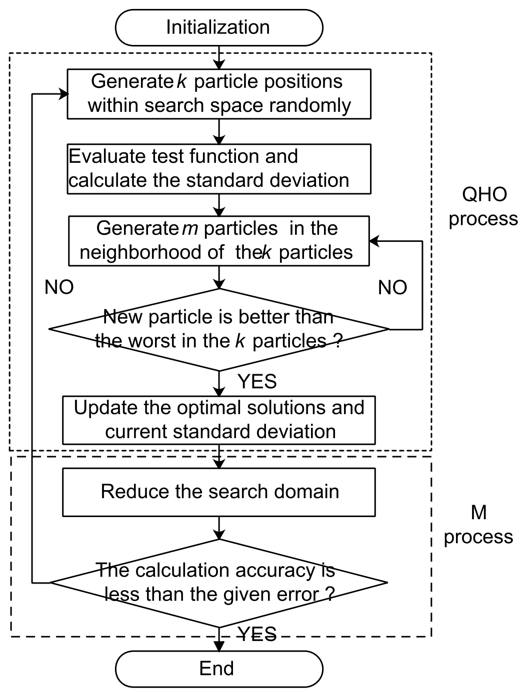


FIGURE 1. **The framework of MQHOA.**

Wavefunction plays an important role in quantum-behaved algorithms. In QPSO, the ground state wavefunction δ potential well works as a sampling probability density function [6], [31]. While in MQHOA, wavefunction of harmonic oscillator potential is employed as a sampling probability density function. Wavefunction reflects the convergence process of the proposed algorithm. For simple object functions, few iterations are needed for particles to escape from high energy levels to the ground state. While for sophisticated functions, a large number of running cycles will be required.

IV. MQHOA WITH TRUNCATED MEAN STABILIZATION STRATEGY

In this section, an improved MQHOA with truncated mean stabilization strategy (TS-MQHOA) is proposed.

A. TRUNCATED MEAN STABILIZATION STRATEGY

Truncated mean or trimmed mean is a statistical measure of central tendency. It calculates the average after discarding the given parts at the high and low ends, typically removing an equal amount of both. The number of points to be removed is usually a percentage or a fixed number of points. It is proved that the median is mostly robust, and high efficiency for mixed distributions [24], [26]. For most statistical applications, 5 to 25 percent of the ends are discarded. In this paper, we discard the highest and the lowest points from the particles, that is 10% of the total number (20) of the population. The truncated stabilization strategy in TS-MQHOA can be depicted as in Fig. 2.

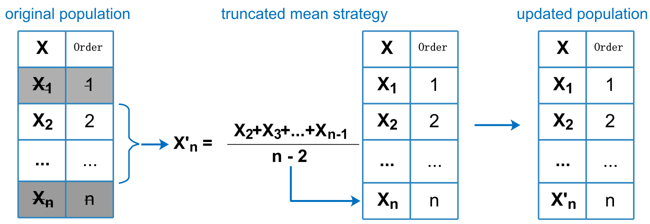


FIGURE 2. The truncated mean stabilization strategy to generate a new population.

In Fig .2, the truncated mean is generated by eliminating the particles whose fitness values are sorted in ascending order, from the lowest to the highest. The average of the remaining position is applied to replace the particle who obtains the largest fitness value (the worst particle).

B. SEARCH SPACE EXPANSION MECHANISM

If there are many local optima in the search space, it is easy for the particles to stagnate and fall in local optima. To overcome this problem, we introduce a search space expansion mechanism. When the algorithm stagnates for a large number of iteration cycles, it implies the particles are falling into local optima. In this case, the search space expansion mechanism is activated, the current search space is enlarged to help the particles to escape from local optima. The expansion coefficient is defined by evaluating a number of benchmark functions. The definition of expansion coefficient is demonstrated in Section V-B2.

C. PSEUDO CODE OF TS-MQHOA

The pseudo code of the proposed TS-MQHOA can be demonstrated as follows.

In Algorithm 1, the parameters k, $X \in [d_{min}, d_{max}]^D$, ε, λ and c denote the particle number, the location of particles within the search space $[d_{min}, d_{max}]$, the computational accuracy, the scale contraction coefficient and the scale expansion factor, respectively. σ_s and σ_k are the current search scale and the standard deviation independently. σ_s is obtained by $|d_u - d_l|/\lambda$ (d_u and d_l are the current upper bound and lower bound). X_w and X_{tm} represent the worst particle (particle obtains the largest fitness value) and the new position generated by truncated mean stabilization strategy, respectively.

Algorithm 1 TS-MQHOA Pseudocode

Input: $k, X \in [d_{min}, d_{max}]^D, \varepsilon, \lambda, c$
Output: the global optimum f_{best}, the optimal position X_{best}
initialization;
evaluate test function and obtain fitness value $f_i = f(X_i)$ and the current minimum $f_{best} = min(F)$
while ($\sigma_s > \varepsilon$) **do**
  while ($\sigma_k > \sigma_s$) **do**
    $\forall X_i \in X$, generate $X_i' \sim N(X_i, \sigma_s^2)$
    $\forall X_i$ and X_i', if $f(X_i') < f(X_i)$ then $X_i = X_i'$
    update **X** by $X_w = X_{tm}$
    update σ_k;
    if $\sigma_k < \sigma_s$ **then**
      finish the iteration cycle
    else
      $\sigma_k = c\sigma_k$
    end
  end
  $\sigma_s = \sigma_s/\lambda$
end

V. EMPIRICAL RESULTS AND DISCUSSION

In this section, the effectiveness and efficiency of the proposed algorithm are fully researched by evaluating well-defined multi-dimensional CEC benchmark functions. Meanwhile, the proposed TS-MQHOA is compared with several state-of-the-art algorithms inspired by swarm intelligence.

A. BENCHMARK FUNCTIONS

To reveal the performance of the proposed algorithm, several multi-dimensional benchmark functions are employed for evaluations from different aspects. In Table 1, $f_1 - f_7$ are unimodal functions and $f_8 - f_{12}$ are multimodal functions [32], [33].

B. PARAMETER SETTING

1) COMMON PARAMETER

The parameters of the comparative algorithms applied in this paper are from the specialized literatures. For all of the population-based algorithms, the group number or the population size is defined $k = 20$. The maximum iteration cycle is set according to the rule used in CEC2017 [34], the maximal iteration generation is defined $maxFE = 10000 * Dimension$, e.g., if the function dimension is 10, the maximal iteration is 100000. The search space $[d_{min}, d_{max}]^D$ for each benchmark

TABLE 1. Benchmark functions.

Function Name	Benchmark Function	D	Range	Optimum
Sphere	$f_1 = \sum_{i=1}^{n} x_i^2$	n	[-5.12,5.12]	$f(0,...,0)=0$
Sum Squares	$f_2 = \sum_{i=0}^{n-1} ix_i^2$	n	[-10,10]	$f(0,...,0)=0$
Rotated Hyper-Ellipsoid	$f_3 = \sum_{i=1}^{n}(\sum_{j=1}^{i} x_i)^2$	n	[-65.54,65.54]	$f(0,...,0)=0$
Ellipsoidal	$f_4 = \sum_{i=1}^{n}(x_i - i)^2$	n	[-100,100]	$f(1,2,...n)=0$
Sum of Different Power	$f_5 = \sum_{i=1}^{n} \lvert x_i \rvert^{i+1}$	n	[-1,1]	$f(0,...,0)=0$
Zakharov	$f_6 = \sum_{i=1}^{n} x_i^2 + (\sum_{i=1}^{n} 0.5ix_i)^2 + (\sum_{i=1}^{n} 0.5ix_i)^4$	n	[-5,10]	$f(0,...,0)=0$
High Conditioned Elliptic	$f_7 = \sum_{i=1}^{n}(10^6)^{\frac{i-1}{n-1}} x_i^2$	n	[-10,10]	$f(0,...,0)=0$
Ackley	$f_8 = -20exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}) - exp(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i)) + 20 + e$	n	$[-32.77, 32.77]$	$f(0,...,0)=0$
Griewank	$f_9 = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	n	[-100,100]	$f(0,...,0)=0$
Levy	$f_{10} = sin^2(\pi\omega_1) + \sum_{i=1}^{n-1}(\omega_i - 1)^2[1 + 10sin^2(\pi\omega_i + 1)] + (\omega_n - 1)^2[1 + sin^2(2\pi\omega_n)]$, where $\omega_i = 1 + \frac{x_i - 1}{4}$,for all $i = 1, ..., n$	n	[-10,10]	$f(1,...,1)=0$
Rastrigin	$f_{11} = 10n + \sum_{i=1}^{n}[x_i^2 - 10\cos(2\pi x_i)]$	n	[-5.12,5.12]	$f(0,...,0)=0$
Modified Schwefel	$f_{12} = 418.9829 \times D - \sum_{i=1}^{n} g(z_i) \quad z_i = x_i + 420.9687462275036, where$	n	[-5.12,5.12]	$f(0,...,0)$
	$g(z_i) = \begin{cases} z_i sin(\lvert z_i \rvert^{1/2}) & if \quad \lvert z_i \rvert \le 500 \\ (500 - mod(z_i, 500))sin(\sqrt{(}\lvert 500 - mod(z_i, 500)\rvert) - \frac{(z_i - 500)^2}{10000n} & if \quad z_i > 500 \\ (mod(\lvert z_i \rvert, 500) - 500)sin\sqrt{(}\lvert mod(\lvert z_i \rvert, 500) - 500\rvert) - \frac{(z_i + 500)^2}{10000n} & if \quad z_i < -500 \end{cases}$			=0.000012727*D

function is set according to Table 1. For SPSO [35], [36], the learning factor $c1$ and $c2$ are both 1.4962, the inertia weight declines linearly from 0.9 to 0.4. For SPSO2011 [3] the inertia weight $\omega = 1/2log(2)$, learning factor $c1 = c2 = 0.5 + log(2)$. The parameters used in CLPSO [5] are inertia weight linearly declines from 0.9 to 0.2, accelerate constant $c1 = c2 = 1.49445$. For QPSO [6], [7], the contraction-expansion coefficient α increases linearly from 0 to 0.5. For DE [37], [38], the crossover probability $pc = 0.2$, the lower and upper bound scaling factors are 0.2 and 0.8 respectively. For ABC [8], [9], the size of the food sources is set half of the colony. The limit trial number is 100, The probability to choose a food source is defined as:

$$Prob = 0.9 * Fitness./max(Fitness) + 0.1 \tag{10}$$

where Fitness is a vector holding fitness (quality) values associated with food sources. For IS-MQHOA and TS-MQHOA, the contraction coefficient $\lambda = 2.0$.

Meanwhile, the stopping criteria for all of the algorithms are uniformly defined as: the computational accuracy is less than 1e-6 or the maximal running generation is larger than 10000*D. All of the comparative methods are coded in Matlab R2016a and executed on the same personal computer with an Intel core(TM) i5-4200U 64 bit, 2.3 GHz and windows 7 operation system.

2) EXPANSION COEFFICIENT

As TS-MQHOA stagnates sometimes when there are several local optima, it takes a long period for particles to jump out or even get stuck in local optima. To solve this problem, we introduce an expansion coefficient to change the current search space and help the population to jump out the local optima. To maintain the current search region within the defined domain, the coefficient can be considered within the range 0.8-2.0. The experiments are carried out by evaluating functions with different coefficients to reveal which coefficient is preferred.

As demonstrated in Fig. 3, the different expansion coefficients affect the performances of the algorithm. For $f_1, f_2, f_3, f_5, f_8, f_9, f_{11}$ and f_{12} the algorithm with the coefficient $c = 1.2$ performs better than others. While for function f_4 and f_{10}, the performance of the algorithm is improved as the decreasing of the coefficient, but the improvement is not significant. Without loss of generality, we define the coefficient c as the *Expansion Coefficient* and $c = 1.2$.

C. EFFECTIVENESS EVALUATION

In this section, the effectiveness of the proposed algorithm is evaluated by applying it to deal with several well-defined and multi-dimensional benchmark functions. Then, the performances of the proposed method is compared with several population-based algorithm inspired by swarm intelligence.

1) SUCCESS PROPORTION

In order to reveal the effectiveness of the proposed algorithm, we carried out a large number of function evaluations from 4 to 100 dimensions. We counted the successful number if any algorithm found the global optimum (the fitness value satisfied the computational accuracy). To reduce the measure error, each benchmark function was evaluated for 50 independent trials. The computational results are recorded in Table 2.

Table 2 demonstrates the successful proportion every algorithm obtains in 4, 10, 30, 50 and 100 dimensional function evaluations, each function was run 50 times independently. The statistic results show that within the 60 function evaluations (12 functions $\times$ 5 dimensions), the ABC algorithm obtains 52, the largest number of 100% (finding the global optimal every time in the 50 independent trials for each function.). TS-MQHOA obtains 49 times of 100%, just ranks behind the best performer ABC, and DE algorithm gains 46 times of 100% followed by TS-MQHOA. While SPSO2011, IS-MQHOA, QPSO, CLPSO and SPSO find 36, 34, 31, 13 and 5 times of 100%, respectively.

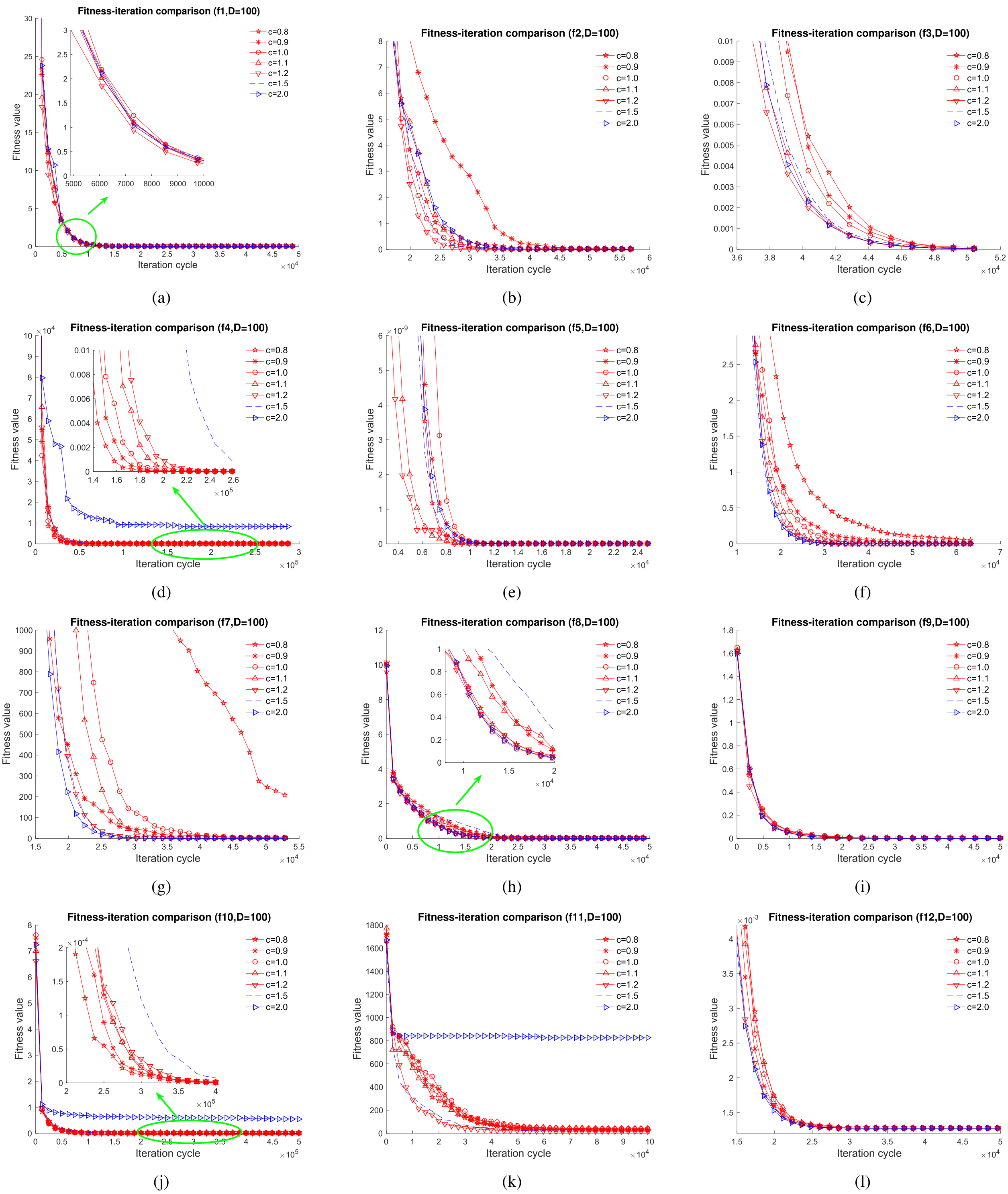


FIGURE 3. **Function evaluation with different coefficients. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_5. (f) f_6. (g) f_7. (h) f_8. (i) f_9. (j) f_{10}. (k) f_{11}. (l) f_{12}.**

For unimodal functions ($f_1 - f_7$), most algorithms perform excellent when the function dimension is not more than 30. However, when the dimension increases to 50 and above, most of the algorithms cannot find the global optima with 100% successful proportion in the 50 individual trials. For instance, in the evaluation of 100-dimensional function f_5 and f_6, only ABC algorithm and TS-MQHOA are able to find the global optima every time, while the other algorithms are not able to find one global minimal in the 50 independent experiments. Meanwhile, in the evaluation

TABLE 2. The success proportion of finding the global optima in 50 independent trials.

Function	Dim	SPSO	SPSO2011	CLPSO	QPSO	ABC	DE	IS-MQHOA	TS-MQHOA
f_1	4	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
	100	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
f_2	4	98.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	22.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
	100	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
f_3	4	74.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
	100	0.00%	0.00%	0.00%	0.00%	100.00%	86.00%	0.00%	100.00%
f_4	4	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
	100	0.00%	100.00%	0.00%	0.00%	100.00%	98.00%	100.00%	100.00%
f_5	4	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	0.00%	0.00%	100.00%	100.00%	86.00%	0.00%	100.00%
	30	16.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	100	0.00%	0.00%	0.00%	0.00%	100.00%	0.00%	0.00%	100.00%
f_6	4	96.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	0.00%	0.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%
	100	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	100.00%
f_7	4	6.00%	0.00%	100.00%	100.00%	100.00%	100.00%	0.00%	100.00%
	10	0.00%	0.00%	0.00%	100.00%	100.00%	100.00%	0.00%	100.00%
	30	0.00%	0.00%	0.00%	100.00%	100.00%	100.00%	0.00%	100.00%
	50	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	0.00%	100.00%
	100	0.00%	0.00%	0.00%	0.00%	100.00%	98.00%	0.00%	100.00%
f_8	4	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	96.00%	0.00%	100.00%	100.00%	100.00%	92.00%	100.00%
	50	0.00%	38.00%	0.00%	0.00%	100.00%	100.00%	30.00%	100.00%
	100	0.00%	0.00%	0.00%	0.00%	100.00%	100.00%	0.00%	100.00%
f_9	4	0.00%	0.00%	0.00%	0.00%	100.00%	84.00%	0.00%	16.00%
	10	0.00%	54.00%	0.00%	0.00%	94.00%	90.00%	10.00%	32.00%
	30	0.00%	72.00%	0.00%	0.00%	70.00%	100.00%	86.00%	82.00%
	50	0.00%	82.00%	0.00%	2.00%	100.00%	100.00%	82.00%	78.00%
	100	0.00%	82.00%	0.00%	0.00%	100.00%	100.00%	82.00%	88.00%
f_{10}	4	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	10	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	98.00%	100.00%
	50	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	94.00%	100.00%
	100	0.00%	96.00%	0.00%	0.00%	100.00%	100.00%	98.00%	96.00%
f_{11}	4	4.00%	46.00%	0.00%	82.00%	100.00%	100.00%	24.00%	68.00%
	10	0.00%	0.00%	0.00%	2.00%	100.00%	86.00%	0.00%	0.00%
	30	0.00%	0.00%	0.00%	0.00%	100.00%	50.00%	0.00%	0.00%
	50	0.00%	0.00%	0.00%	0.00%	100.00%	10.00%	0.00%	0.00%
	100	0.00%	0.00%	0.00%	0.00%	94.00%	0.00%	0.00%	8.00%
f_{12}	4	18.00%	94.00%	48.00%	56.00%	58.00%	36.00%	92.00%	100.00%
	10	0.00%	100.00%	6.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	30	0.00%	100.00%	0.00%	100.00%	100.00%	100.00%	100.00%	100.00%
	50	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
	100	0.00%	100.00%	0.00%	0.00%	100.00%	100.00%	100.00%	100.00%
	frequency	5/60	36/60	13/60	31/60	52/60	46/60	34/60	49/60

The **frequency** denotes the number of 100% each algorithm gains.

of 100-dimensional function f_6, only TS-MQHOA is able to obtain the global minimal with 100% successful proportion. Statistically, in the unimodal function evaluations, TS-MQHOA performs the best, finding every global optimal solution with 100% success proportion in the 50 independent trials. The performance of ABC is closed to TS-MQHOA, obtaining 31 times of 100%, following by DE, obtaining 28. SPSO2011 and IS-MQHOA perform similarly in the evaluations of function ($f_1 - f_7$), both gaining 26 times of 100%. QPSO, CLPSO and SPSO find 22, 11 and 4 global optima, individually.

In the evaluation of multimodal functions f_8-f_{12}, the ABC algorithm performs the best, obtaining 21 times of 100% in the 25 function evaluations (5 functions × 5 dimensions), following by DE and TS-MQHOA, which gain 18 and 14 times of 100% respectively. The rest of the algorithms are able to find 10, 9, 8, 2 and 1 times of 100% for SPSO2011, QPSO, IS-MQHOA, CLPSO and SPSO separately.

Similar to the performances of function evaluation in the function $f_1 - f_7$, most algorithms perform well in the evaluation of 4-dimensional, 10-dimensional and 30-dimensional function f_8, f_{10} and f_{11}. While in the comparison of 30 and 50 dimensional function f_8 evaluations, the successful proportion obtained by QPSO, SPSO2011 and IS-MQHOA drop from 100% to 0, 96% to 38% and 92% to 30% respectively. While in the 100-dimensional function evaluations, only TS-MQHOA, ABC and DE are able to find the global optimal with 100% successful proportion in the 50 independent trials. Interestingly, in the evaluation of multi-dimensional function f_9 and f_{12}, most algorithms perform weaker in the 4-dimensional function evaluations, comparing with higher dimensional function evaluations. For example, in the evaluation of function f_9, only ABC, DE and TS-MQHOA are able to find the global minimal in the 4-dimensional function evaluations with 100%, 84% and 16% successful proportion, respectively. While in the evaluation of 100-dimensional function f_9, more algorithms are able to find the global optimal solutions, e.g., the successful proportion increase from 0 to 82% for both of SPSO2011 and IS-MQHOA. DE and TS-MQHOA find the global minimal with higher successful rate, increasing from 84% and 16% to 100% and 88%, respectively. In the evaluation of 4-dimensional function f_{12}, all of the algorithms are not able to find the global minimum with 100% successful proportion, except for TS-MQHOA. While in the 30-dimensional function evaluations, all of the algorithms find the global optimal solution with full success. It should be noticed that, in the evaluation of 4-dimensional function f_{12}, only TS-MQHOA is able to find the global optimal with 100% successful proportion.

2) DISTRIBUTION OF GLOBAL OPTIMA

In Table 2, though the algorithms find the global optima with the same success proportion, the distribution of the global optima may differ from each other. In fact, the distribution of the global optima found by an algorithm in 50 independent trials reflects the robustness of the algorithm. The less vibration of the results, the more robust of the algorithm. To reveal the distribution of the optimal solutions, we put the data into boxplot, which are exhibited in Fig. 4. As the algorithms obtain the most times of finding the global optima with 100% proportion, we carried out the experiments on the 10-dimensional function evaluations.

As demonstrated in Fig. 4, for most of the algorithms, their global optima vibrate mildly at the center of the boxes, such as the boxes in Fig. 4(a), (b), (c), (d), (h), (j) and (k), for these functions, the algorithms perform excellently with slight vibrations except for SPSO. It has sharp vibrations in all of the functions, which is in accord with its disability to find the global optimum with 100% proportion in any function evaluation in Table 2. Meanwhile, for algorithms which are not able to find the global optima, their performances may vary a lot. As demonstrated in Fig.4(i), although none of the algorithms (except for DE) find the global optimum in the 10-dimensional function f_9 evaluation, their boxes of best fitness values are different from each other. The vibration of TS-MQHOA in Fig.4(i) is milder than SPSO2011, CLPSO, QPSO, ABC and IS-MQHOA, which is in line with the second best algorithm in Table 2, obtaining 94% successful proportion.

D. EFFICIENCY EVALUATION

Although the referred algorithms may obtain the same success proportion in Section V-C, they may differ from the computational precision, the iteration cycle and the total run time during the convergence process. In this section, evaluations among the referred techniques are conducted on the computational accuracy of fitness values, the total iteration cycles, the run time in the course of convergence and the fitness-iteration relations during the convergence process.

1) COMPUTATIONAL PRECISION

Although the results obtained in Section V-C reflect the performances of an algorithm by the success proportions they obtain, we cannot distinguish their performance if they get the same proportion. In order to further reveal the search ability of the referred algorithms, we list and make comparisons of the computational results by the best fitness, mean fitness, averaged standard deviation of the fitness values, the iteration cycles and total run time they obtain in 100-dimensional function evaluations. The results are the average from the 50 independent experiments which are demonstrated in Table 3.

Table 3 illustrates the detailed performances of the algorithms on the best fitness value, the average value, the standard deviation of the fitness values obtained by the population, the total iteration cycle and the CPU run time in function evaluations. The best fitness value is obtained by the smallest fitness value when the program finish its run, the mean value and the standard deviation are denoted as the average values and the standard deviation of the fitness values obtained by the population when the program ends running. The **iterNum** and **time** are defined by the iteration cycles and

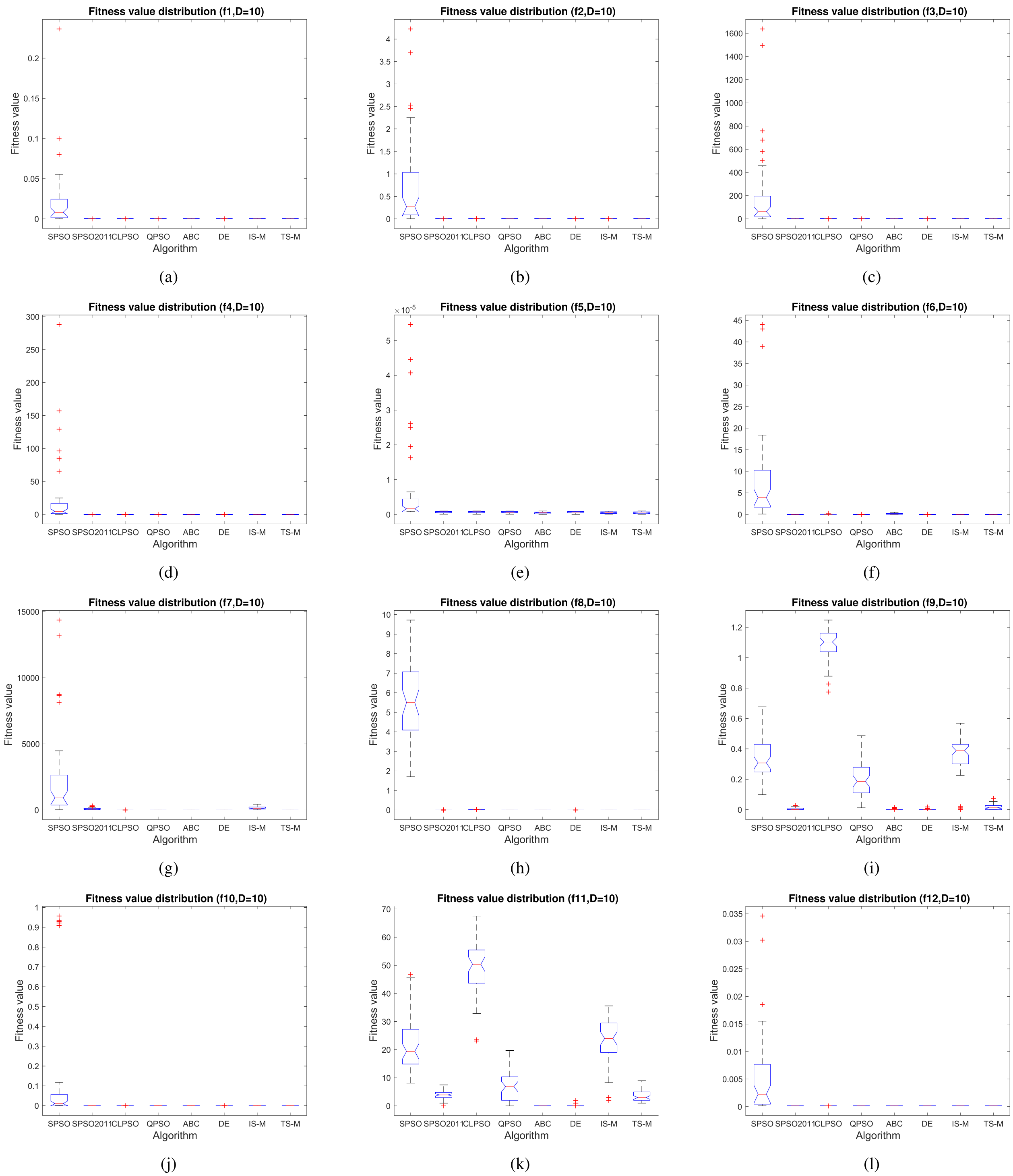


FIGURE 4. Box plot of the best fitness value from 50 independent trials. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_5. (f) f_6. (g) f_7. (h) f_8. (i) f_9. (j) f_{10}. (k) f_{11}. (l) f_{12}.

CPU run time from starts to finish running of the program (satisfying the stopping criteria, that is when the iteration cycle reaches to maxFE or the computational precision meets the given error).

As the bold data shown, TS-MQHOA finds all of the global optima within the given computational precision, except for f_{11}. Meanwhile, in most cases, TS-MQHOA obtains the smallest value in the comparisons of the mean value (average fitness), requiring the least iteration cycles and CPU run time. Moreover, in evaluation of function f_2, f_6, f_7 and f_8, TS-MQHOA is several order of magnitude better than IS-MQHOA and some algorithms on the best fitness value,

TABLE 3. Detailed computational results obtained by SPSO, SPSO2011, CLPSO, QPSO, ABC, DE, IS-MQHOA and TS-MQHOA. The records are the average from 50 independent trials.

Function	Item	SPSO	SPSO2011	CLPSO	QPSO	ABC	DE	IS-MQHOA	TS-MQHOA
f_1	best	6.456E+01	9.985E-07	7.712E-01	1.092E+02	**6.037E-07**	9.577E-07	6.163E-07	9.737E-07
	mean	6.794E+01	1.196E-06	3.268E+01	1.515E+02	1.976E-02	1.286E-06	**6.387E-07**	1.196E-06
	std	4.195E+00	1.084E-07	3.904E+01	1.621E+00	3.155E-03	1.575E-07	**1.673E-08**	3.358E-05
	iterNUM	1.000E+06	7.224E+04	1.000E+06	1.000E+06	3.944E+04	3.784E+04	4.646E+04	**7.801E+03**
	time	2.888E+02	3.810E+01	4.375E+02	3.964E+02	1.616E+01	1.872E+01	1.759E+01	**3.091E+00**
f_2	best	1.225E+04	2.790E-05	5.719E+01	9.287E+03	**5.163E-07**	9.728E-07	1.103E-04	8.605E-07
	mean	1.227E+04	2.802E-05	3.394E+03	1.684E+04	5.400E+00	1.230E-06	1.124E-04	**9.873E-07**
	std	5.816E+01	**7.900E-08**	3.905E+03	2.577E+00	2.906E-03	1.309E-07	1.180E-06	2.933E-06
	iterNUM	1.000E+06	1.000E+06	1.000E+06	1.000E+06	4.850E+04	4.796E+04	1.000E+06	**1.262E+04**
	time	7.413E+01	1.687E+02	1.207E+02	1.135E+02	5.250E+00	7.169E+00	1.019E+02	**1.559E+00**
f_3	best	3.807E+07	9.623E-07	5.564E+05	7.961E+07	9.966E-07	9.472E-07	**6.498E-07**	7.225E-07
	mean	3.814E+07	1.195E-06	2.464E+07	1.029E+08	6.008E-01	1.237E-06	**7.103E-07**	9.732E-07
	std	1.481E+05	1.034E-07	2.494E+07	2.024E+01	3.362E-04	1.481E-07	**3.033E-08**	5.743E-07
	iterNUM	1.000E+06	1.265E+05	1.000E+06	1.000E+06	6.416E+04	6.288E+04	8.126E+04	**1.410E+04**
	time	1.291E+02	5.662E+01	1.800E+02	1.634E+02	1.091E+01	1.283E+01	1.224E+01	**2.445E+00**
f_4	best	2.584E+05	9.944E-07	4.959E-03	6.796E+03	6.923E-07	9.882E-07	**3.039E-07**	6.032E-07
	mean	2.585E+05	1.188E-06	3.962E-01	1.016E+04	5.103E-01	1.222E-06	**3.207E-07**	4.686E+01
	std	7.630E+01	1.015E-07	5.036E-01	1.362E+01	1.400E-02	1.293E-07	**1.444E-08**	5.793E-01
	iterNUM	1.000E+06	1.080E+05	1.000E+06	1.000E+06	5.294E+04	**4.700E+04**	7.132E+04	6.938E+04
	time	2.896E+02	6.405E+01	4.947E+02	3.941E+02	**2.063E+01**	2.346E+01	2.602E+01	2.914E+01
f_5	best	9.145E-07	9.924E-07	7.970E-07	1.397E+00	4.579E-07	8.692E-07	6.781E-07	**3.347E-07**
	mean	2.545E-06	3.954E-02	1.825E-01	1.397E+00	1.587E-01	7.431E-06	3.778E-06	**1.059E-06**
	std	**4.517E-06**	1.387E-01	3.441E-01	5.414E-01	2.687E-01	7.938E-06	4.614E-06	6.594E-03
	iterNUM	1.201E+05	1.062E+04	3.270E+04	1.000E+06	8.402E+04	1.398E+04	2.121E+03	**1.521E+03**
	time	2.319E+01	3.603E+00	9.642E+00	2.944E+02	2.313E+01	4.647E+00	5.561E-01	**4.941E-01**
f_6	best	2.346E+03	1.343E-02	1.887E+02	7.247E+02	1.203E+03	5.526E+01	1.155E-05	**2.013E-07**
	mean	2.361E+03	7.047E-01	4.935E+09	3.029E+09	4.672E+14	7.967E+01	1.172E-05	**2.248E-07**
	std	3.924E+01	1.667E+00	1.440E+10	2.388E+00	4.064E+00	1.091E+01	**4.707E-08**	3.295E-06
	iterNUM	1.000E+06	1.000E+06	1.000E+06	1.000E+06	1.000E+06	1.000E+06	1.000E+06	**2.992E+04**
	time	1.902E+02	3.343E+02	2.798E+02	2.341E+02	2.407E+02	2.763E+02	2.239E+02	**7.541E+00**
f_7	best	5.529E+06	1.288E+03	1.101E+01	5.350E+04	**3.366E-07**	9.410E-07	1.929E+03	6.247E-07
	mean	5.591E+06	1.290E+03	2.370E+03	3.453E+05	3.627E-01	1.161E-06	1.935E+03	**6.897E-07**
	std	1.490E+05	1.607E+00	3.613E+03	1.658E+00	4.390E-05	1.174E-07	1.630E+00	**6.067E-08**
	iterNUM	1.000E+06	1.000E+06	1.000E+06	1.000E+06	6.494E+04	5.366E+04	1.000E+06	**3.374E+04**
	time	2.407E+02	3.400E+02	2.856E+02	2.608E+02	1.829E+01	1.796E+01	2.400E+02	**8.366E+00**
f_8	best	1.414E+01	2.478E+00	4.211E+00	1.511E+01	9.423E-07	9.977E-07	2.289E+00	**5.958E-07**
	mean	1.415E+01	2.478E+00	9.972E+00	1.660E+01	2.749E-05	1.109E-06	2.289E+00	**6.893E-07**
	std	8.076E-03	4.619E-15	4.319E+00	8.488E+00	2.369E-06	6.479E-08	**8.189E-16**	7.142E-08
	iterNUM	1.000E+06	1.000E+06	1.000E+06	1.000E+06	8.800E+04	6.724E+04	1.000E+06	**1.562E+04**
	time	3.003E+01	2.066E+02	9.004E+01	6.860E+01	6.169E+00	7.726E+00	5.072E+01	**1.075E+00**
f_9	best	4.420E+00	9.870E-07	1.065E+00	9.797E+00	**1.680E-07**	9.590E-07	4.260E-07	2.500E-07
	mean	4.427E+00	1.100E-06	3.691E+00	1.264E+01	1.185E-01	1.170E-06	4.380E-07	**3.120E-07**
	std	2.079E-02	5.650E-08	2.918E+00	2.598E+01	5.552E-01	1.220E-07	**6.720E-09**	1.324E-04
	iterNUM	1.000E+06	1.610E+05	1.000E+06	1.000E+06	5.666E+04	4.028E+04	1.290E+05	**1.144E+04**
	time	2.953E+02	6.394E+01	3.438E+02	3.146E+02	1.866E+01	1.558E+01	3.610E+01	**3.879E+00**
f_{10}	best	1.953E+01	9.906E-07	1.515E-01	2.403E+01	**4.602E-07**	9.556E-07	8.528E-07	5.359E-07
	mean	1.971E+01	1.216E-06	7.018E+00	3.220E+01	2.844E-04	1.182E-06	**8.981E-07**	9.605E-04
	std	4.804E-01	1.094E-07	7.661E+00	2.766E+00	2.766E-03	1.061E-07	**3.600E-08**	1.147E-02
	iterNUM	1.000E+06	7.174E+04	1.000E+06	1.000E+06	3.986E+04	**3.584E+04**	4.782E+04	4.048E+04
	time	3.388E+02	3.477E+01	3.985E+02	3.631E+02	1.345E+01	1.408E+01	1.356E+01	**1.245E+01**
f_{11}	best	6.255E+02	6.069E+01	3.943E+02	1.047E+03	**2.281E-07**	2.628E+02	7.866E+02	1.990E+01
	mean	6.598E+02	8.740E+01	8.714E+02	1.120E+03	**1.023E+01**	2.911E+02	8.288E+02	2.044E+01
	std	4.524E+01	1.522E+02	3.350E+02	1.851E+00	2.028E-01	1.307E+01	2.463E+01	**2.283E-03**
	iterNUM	1.000E+06	1.000E+06	1.000E+06	1.000E+06	**6.871E+05**	1.000E+06	1.000E+06	9.185E+05
	time	2.143E+02	3.448E+02	2.977E+02	2.979E+02	**1.992E+02**	3.296E+02	2.619E+02	2.610E+02
f_{12}	best	6.754E+00	**1.273E-03**	1.004E-01	1.101E+01	**1.273E-03**	**1.273E-03**	**1.273E-03**	**1.273E-03**
	mean	6.795E+00	**1.273E-03**	4.432E+00	1.653E+01	**1.274E-03**	**1.273E-03**	**1.273E-03**	**1.273E-03**
	std	1.190E-01	5.451E-09	4.724E+00	1.311E+00	1.032E-04	5.322E-09	9.062E-10	**2.413E-05**
	iterNUM	1.000E+06	7.536E+04	1.000E+06	1.000E+06	4.692E+04	3.906E+04	4.974E+04	**9.021E+03**
	time	3.476E+02	2.236E+03	4.123E+02	3.845E+02	1.743E+01	1.685E+01	1.818E+01	**3.680E+00**

The bold mark indicates that they are the best results among all algorithms.

iteration cycle and CPU run time. It should be noticed that in the evaluation of function f_6, TS-MQHOA is the only one which is able to obtain the global optimum satisfying the computational accuracy.

2) FITNESS-ITERATION

The success proportion in Table 2, the distribution of the global optima in Fig. 4, the best fitness value, the mean best fitness, the standard deviation, the iteration cycle and the run time in Table 3 reflect specific convergence properties of the referred algorithms to some extent. The smaller of the values in Table 3, the more superior of the algorithm. However, these values are not convincing enough to fully reflect the performance of an algorithm. There is a possibility that an algorithm may require a large number of iteration cycles to converge but cost less CPU run time. Another case is that it is impossible to know how do the algorithms perform in detail in the course of function evaluations. The fitness-iteration chart overcomes these disadvantages and helps to objectively demonstrate the whole process when fitness values obtained by an algorithm in every iteration cycle. The fitness-iteration relation also reflects the convergent efficiency of an algorithm as it dynamically records the trend of convergence in the course of function evaluation. Without loss of generality, the fitness-iteration experiments were carried out on the 100-dimensional benchmark functions. The maximal iteration cycle is defined by the least iteration cycles in a function evaluation in Table 3. The fitness-iteration results of the referred algorithms are exhibited in Fig. 5.

From an overall perspective of the subfigures in Fig. 5, the fitness-iteration curves of TS-MQHOA are universally bottomed much lower than other algorithms, which indicates its more efficient convergence performance. As demonstrated in Fig. 5(a), the curves of TS-MQHOA and IS-MQHOA are much lower than that of other algorithms. Although SPSO2011 does not perform better than SPSO in the first 1000 iteration cycles (the curve of SPSO is lower than SPSO2011), it outperforms SPSO after that (SPSO converges very slowly after 1000 iteration cycles). DE and ABC perform similarly, though converge slower than the TS-MQHOA, IS-MQHOA and SPSO2011, their curves are much lower than that of QPSO and CLPSO. Similar situation happens to function f_2 and f_3, except that DE and ABC converge to the global optimum within 12000 iteration cycles in Fig. 5(b) and they converge closely to the global optima within 14000 iterations in Fig. 5(c).

In Fig. 5(d), DE and ABC perform much better than the other methodologies, they find the global optima within about 20000 iteration cycles. Following by TS-MQHOA, IS-MQHOA and SPSO2011, their fitness-iteration curves lay lower than that of SPSO, QPSO and CLPSO. In the evaluation of function f_5 (Fig. 5(e)), the superiority of TS-MQHOA is obvious, it converges to the global optimum within 100 iterations, while the other methods do not find the global best solutions within 500 iteration cycles. The chart of SPSO is much lower than that of CLPSO, ABC, QPSO and DE in Fig. 5(e). The partially enlarged figure in Fig. 5(f) demonstrates that TS-MQHOA is much more efficient to converge to the global optimum within 5000 iteration cycles, while the other algorithms are not able to find the global optimum within 30000 iterations.

Although the curves in Fig. 5(g) are decreasing quickly in the 30000 iterations, they perform differently. As shown in the enlarged part, the chart of TS-MQHOA lays the lowest, which indicates much more efficient than the other algorithms referred. The fitness-iteration chart of TS-MQHOA in Fig. 5(h) indicates the significant performance of the algorithm to find the global optimal solution within 6000 iterations, while the other methods cannot find the best solution within 15000 iteration cycles. Similar to Fig. 5(a)-(c), in Fig. 5(i), the curve of TS-MQHOA keeps excellent from beginning until meet the stopping criteria, converging to the global minimum within 6000 iteration cycles. In Fig. 5(j), TS-MQHOA, IS-MQHOA, SPSO2011, ABC and DE quickly converge to the global optimum at around 15000 iterations, while SPSO, CLPSO and QPSO converge much slower.

Similar to Fig. 5(j), in Fig. 5(k), TS-MQHOA and ABC perform significantly better than other algorithms. Especially for TS-MQHOA, the fitness value declines from 1800 to 20 within 20000 iteration cycles, while most of the algorithms are unable to find the global optimum before satisfying the stopping criteria. ABC converges quickly to the global minimum within around 40000 iterations. The performances on the evaluation of function f_{12} is similar to Fig. 5(a)-(c), the chart of TS-MQHOA lays the lowest which indicates the superiority of it. SPSO converges faster than SPSO2011 at the first 1500 iterations, it stops to convergence after that. Though the fitness-iteration of DE and ABC converge slowly, they find the global optimum within 9000 iterations, but QPSO and CLPSO are not able to find the global optimal solutions.

E. BRIEF DISCUSSION

Theoretically, the truncated mean strategy removes the extreme particles, which helps to prevent trapping into local optima and improve the diversity of the particles [39]. Experimentally, from an overall perspective of the experimental results in Section V, TS-MQHOA outperforms the techniques referred from several aspects as follows: 1) it gains the high-frequency of finding the global optima in Table 2 and; 2) it has the least vibration of fitness value for 50 independent trials (Fig. 4) and; 3) it achieves high computational accuracy with less iteration cycles and CPU run time in Table 3 and; 4) it obtains the lowest fitness-iteration curves in Fig. 5. The superiority of TS-MQHOA is significant in high-dimensional function evaluations. The good performance of TS-MQHOA can be due to the reasons as follows.

Firstly, the mechanism multi-scale variation of search space (multi-scale quantum harmonic oscillator process in MQHOA) ensures the computational precision of the fitness value when the stopping criterion is satisfied. While the techniques based on particle swarm optimization are stopped by the maximum generation, which may lead to premature

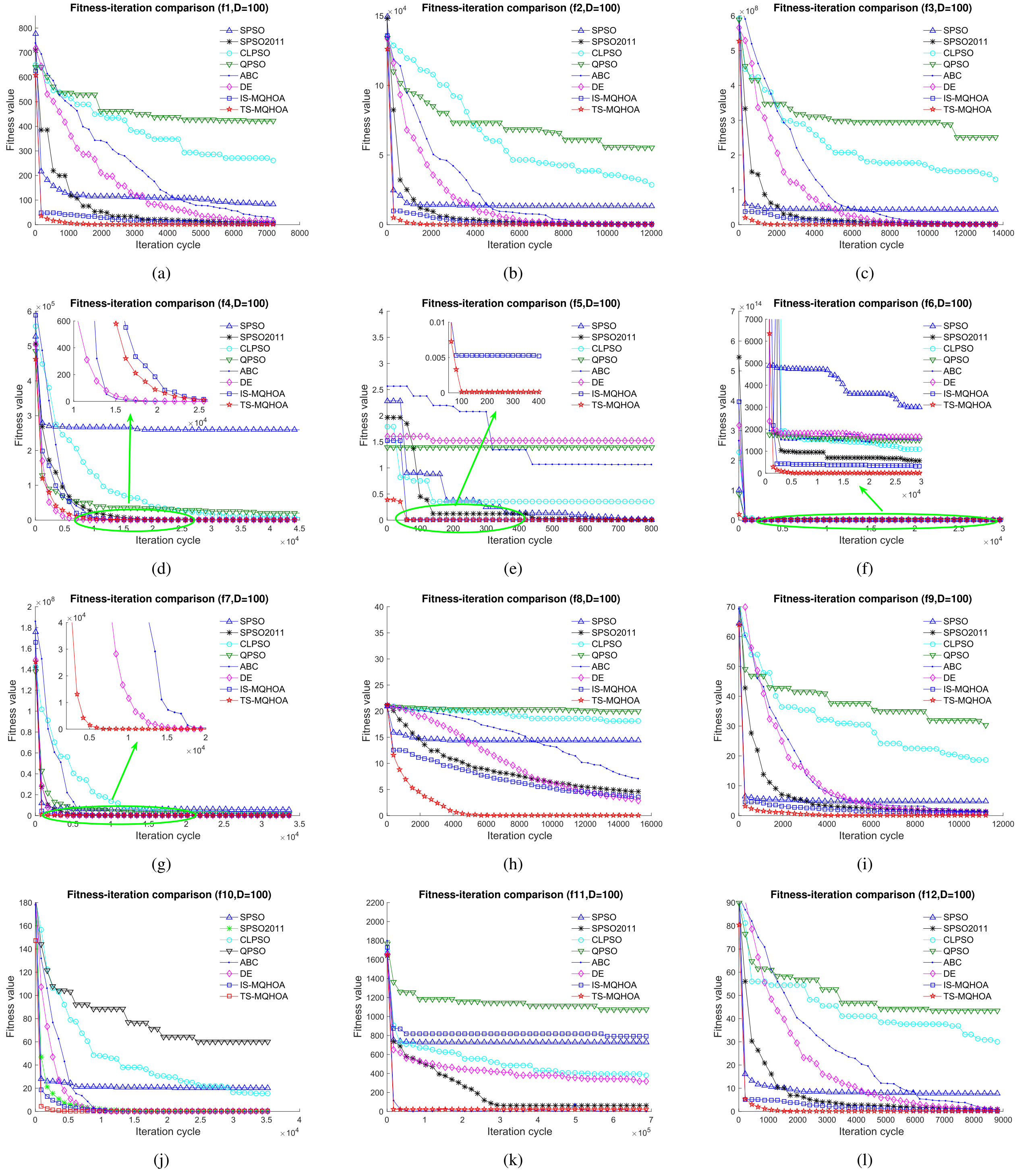


FIGURE 5. The fitness-iteration comparison for 100-dimensional function evaluations. (a) f_1. (b) f_2. (c) f_3. (d) f_4. (e) f_5. (f) f_6. (g) f_7. (h) f_8. (i) f_9. (j) f_{10}. (k) f_{11}. (l) f_{12}.

convergence, and hence the computational accuracy of SPSO, SPSO2011, CLPSO, QPSO and DE is frequently not as good as the proposed algorithm.

Secondly, to some extent, the truncated mean stabilization policy helps to keep away from some local optima. As local optima are frequently at the two ends of a sorted population, the truncated mean stabilization mechanism in TS-MQHOA generates a new particle without considering information from some local optima. The worst particles are continuously replaced by the truncated mean position (a new position without containing information from some local optima) and hence the diversification of particles is reinforced. Meanwhile, the differential evolution mechanism which helps to share information of the leader particle is beneficial to fast iteration and convergence.

For DE algorithm, it generates new candidate solutions by utilizing a differential mechanism: $Y = X_a + F(X_b - X_c)$, where $Y = y_1, y_2, \ldots, y_n$ is the new candidate solutions, n is the dimension size, X_a, X_b and X_c are three candidate solutions randomly selected from the population. As the crossover and mutation mechanisms diversifying the population in DE, it converges more efficiently than SPSO, SPSO2011, CLPSO and QPSO in Fig. 5. Similarly, in artificial bee colony algorithm, the algorithm generates a new onlooker bee V_{ik} by the mechanism as follows.

$$V_{ik} = X_{ik} + \phi_{ik}(X_{ik} - X_{jk}) * (rand - 0.5) * 2 \tag{11}$$

where $X_{ik} = x_{i,1}, x_{i,2}, \ldots, x_{i,n}$ is the old onlooker bee, n is the dimension size, ϕ_{ik} is a random number from (0, 1), X_j is a candidate solution randomly selected from a neighbor of the onlooker bee. As both of the ϕ and the X_j are randomly generated or selected, the diversification of the population is reinforced. Meanwhile, the message come from the neighbors help to fast converge in function evaluations.

And thirdly, the mechanism of expanding the current search space when the algorithm stagnates for a long period helps the proposed algorithm to jump out from local optima. For different nature of optimization problems, the expansion coefficient should be decided by experimental trials.

VI. CONCLUSION

This paper proposes a new multi-scale quantum harmonic oscillator algorithm with truncated mean stabilization strategy (TS-MQHOA) to improve the convergence performance and diversify the population. The proposed approach is theoretically and experimentally analyzed to be more efficient to search for the global optima in function evaluations compared with the IS-MQHOA. Simulations on difficult multidimensional problems reveal the superiority of the proposed algorithm. The computational results are compared with several well-known heuristic algorithms such as the standard PSO, SPSO2011, CLPSO, QPSO, ABC and DE. The experimental results reveal the competitiveness or superiority of the proposed algorithm. Meanwhile, the performances improved by the truncated mean stabilization strategy indicate the positive impacts of it on improving the convergence performance of the proposed algorithm. Moreover, the universal improvements of the performance in TS-MQHOA imply a promising mechanism which can be easily implemented in other heuristic techniques based on swarm intelligence.

In the near future, the truncated mean mechanism applying to other population-based algorithms and the application of TS-MQHOA to real-world optimization problems deserve our researching.

REFERENCES

[1] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proc. 6th Int. Symp. Micro Mach. Human Sci.*, New York, NY, USA, vol. 1, Oct. 1995, pp. 39–43.

[2] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Netw.*, Nov. 1995.

[3] M. Omran. (2011). *Spso 2011, MATLAB*. [Online]. Available: http://www.particleswarm.info/Programs.html

[4] H. Hakli and H. Uğuz, "A novel particle swarm optimization algorithm with Levy flight," *Appl. Soft Comput.*, vol. 23, pp. 333–345, Oct. 2014.

[5] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Trans. Evol. Comput.*, vol. 10, no. 3, pp. 281–295, Jun. 2006.

[6] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proc. IEEE Conf. Cybern. Intell. Syst.*, vol. 1, Dec. 2004, pp. 111–116.

[7] J. Sun, W. Fang, X. Wu, V. Palade, and W. Xu, "Quantum-behaved particle swarm optimization: Analysis of individual particle behavior and parameter selection," *Evol. Comput.*, vol. 20, no. 3, pp. 349–393, Sep. 2012.

[8] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Fac. Eng., Erciyes Univ., Kayseri, Turkey, Tech. Rep. TR06, 2005.

[9] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," *Appl. Math. Comput.*, vol. 214, no. 1, pp. 108–132, 2009.

[10] R. Vatankhah, S. Etemadi, A. Alasty, G. R. Vossoughi, and M. Boroushaki, "Active leading through obstacles using ant-colony algorithm," *Neurocomputing*, vol. 88, no. 7, pp. 67–77, 2012.

[11] G. Wang and L. Guo, "A novel hybrid bat algorithm with harmony search for global numerical optimization," *J. Appl. Math.*, vol. 2013, 2013, Art. no. 696491.

[12] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Proc. Int. Conf. Adv. Swarm Intell.*, 2010, pp. 355–364.

[13] J. Li and Y. Tan, "The bare bones fireworks algorithm: A minimalist global optimizer," *Appl. Soft Comput.*, vol. 62, pp. 454–462, Jan. 2018.

[14] W. Peng and H. Yan, "Physical model of multi-scale quantum harmonic oscillator optimization algorithm," *J. Frontiers Comput. Sci. Exploration*, vol. 9, no. 10, pp. 1271–1280, 2015.

[15] P. Wang, X. Ye, B. Li, and K. Cheng, "Multi-scale quantum harmonic oscillator algorithm for global numerical optimization," *Appl. Soft Comput.*, vol. 69, pp. 655–670, Aug. 2018.

[16] P. Wang, K. Cheng, Y. Huang, B. Li, X. Ye, and X. Chen, "Multiscale quantum harmonic oscillator algorithm for multimodal optimization," *Comput. Intell. Neurosci.*, vol. 2018, May 2018, Art. no. 8430175.

[17] P. Wang *et al.*, "Multi-scale quantum harmonic oscillator algorithm with individual stabilization strategy," in *Proc. Int. Conf. Swarm Intell.*, 2018, pp. 624–633.

[18] P. Tsakalides, F. Trinic, and C. L. Nikias, "Performance assessment of CFAR processors in Pearson-distributed clutter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 36, no. 4, pp. 1377–1386, Oct. 2000.

[19] J. Quigley and L. Walls, "Conditional lifetime data analysis using the limited expected value function," *Qual. Rel. Eng. Int.*, vol. 20, no. 3, pp. 185–192, 2004.

[20] J.-F. Caron *et al.*, "Improved particle identification using cluster counting in a full-length drift chamber prototype," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 735, pp. 169–183, Jan. 2014.

[21] X. Jiang, "Iterative truncated arithmetic mean filter and its properties," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1537–1547, Apr. 2012.

[22] Z. Miao and X. Jiang, "Weighted iterative truncated mean filter," *IEEE Trans. Signal Process.*, vol. 61, no. 16, pp. 4149–4160, Aug. 2013.

[23] A. Sanz-García, J. Fernández-Ceniceros, F. Antoñanzas-Torres, and F. J. Martinez-de-Pisón-Ascacibar, "Application of genetic algorithms to optimize a truncated mean k-nearest neighbours regressor for hotel reservation forecasting," in *Hybrid Artificial Intelligent Systems* (Lecture Notes in Computer Science), vol. 7208. Berlin, Germany: Springer, 2012, pp. 79–90.

[24] Z. Elkhadir, C. Khalid, and M. Benattou, "An effective network intrusion detection based on truncated mean LDA," in *Proc. Int. Conf. Electr. Inf. Technol.*, Nov. 2017, pp. 1–5.

[25] R. Abbasi *et al.*, "An improved method for measuring muon energy using the truncated mean of dE/dx," *Nucl. Instrum. Methods Phys. Res. A, Accel. Spectrom. Detect. Assoc. Equip.*, vol. 703, pp. 190–198, Mar. 2013.

[26] Z. Miao and X. Jiang, "Additive and exclusive noise suppression by iterative trimmed and truncated mean algorithm," *Signal Process.*, vol. 99, pp. 147–158, Jun. 2014.

[27] A. Crispin and A. Syrichas, "Quantum annealing algorithm for vehicle scheduling," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2013, pp. 3523–3528.

[28] F. Schwabl, *Quantum Mechanics*, R. Kates, Ed. Berlin, Germany: Springer, 2007.

[29] D. I. Blokhintsev, *Quantum Mechanics*, M. J. Kearsley and J. B. Sykes, Eds. Dordrecht, The Netherlands: Springer, 1964.

[30] J. Brooke, D. Bitko, and G. Aeppli, "Quantum annealing of a disordered magnet," *Science*, vol. 284, no. 5415, pp. 779–781, 2001.

[31] S. Meshoul and M. Batouche, "A novel quantum behaved particle swarm optimization algorithm with chaotic search for image alignment," in *Proc. IEEE Congr. Evol. Comput.*, Jul. 2010, pp. 1–6.

[32] B. Yuan and M. Gallagher, "Experimental results for the special session on real-parameter optimization at CEC 2005: A simple, continuous EDA," in *Proc. IEEE Congr. Evol. Comput.*, vol. 2, Sep. 2005, pp. 1792–1799.

[33] N. H. Awad, M. Z. Ali, P. N. Suganthan, and R. G. Reynolds, "An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC2014 benchmark problems," in *Proc. IEEE Congr. Evol. Comput. (CEC)*, Vancouver, BC, Canada, Jul. 2016, pp. 2958–2965.

[34] N. H. Awad, M. Z. Ali, J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization," Nanyang Technol. Univ., Singapore, Tech. Rep., Nov. 2016.

[35] R. C. Eberhart and Y. Shi, "Comparison between genetic algorithms and particle swarm optimization," in *Proc. Int. Conf. Evol. Program. VII*, vol. 1447. Berlin, Germany: Springer, 1998, pp. 611–616.

[36] E. Davoodi, M. T. Hagh, and S. G. Zadeh, "A hybrid improved quantum-behaved particle swarm optimization–simplex method (IQPSOS) to solve power system load flow problems," *Appl. Soft Comput.*, vol. 21, no. 2, pp. 171–179, 2014.

[37] S. Das, A. Abraham, U. K. Chakraborty, and A. Konar, "Differential Evolution Using a Neighborhood-Based Mutation Operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526–553, Jun. 2009.

[38] B.-Y. Qu, P. N. Suganthan, and J.-J. Liang, "Differential evolution with neighborhood mutation for multimodal optimization," *IEEE Trans. Evol. Comput.*, vol. 16, no. 5, pp. 601–614, Oct. 2012.

[39] Y. Zhu, "Nonparametric density estimation based on the truncated mean," *Statist. Probab. Lett.*, vol. 83, no. 2, pp. 445–451, 2013.

XINGGUI YE (SM'18) received the B.E. and M.E. degrees in electrical engineering and automation from Fuzhou University, Fuzhou, China, in 2008 and 2011, respectively. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences, Beijing. He was with China Unicom as a Cloud Computing Engineer and an Information and Communication Technology solution Engineer.

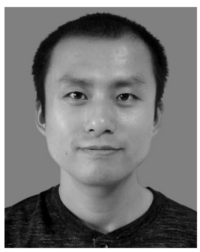
His research interests include computational intelligence, swarm intelligence, quantum computing, cloud computing, smart cities, and electrical engineering and automation.

PENG WANG received the B.E. and M.S. degrees from Sichuan University, in 1998 and 2001, respectively, and the Ph.D. degree from the Institute of Computer Application, Chinese Academy of Sciences, in 2004. He is currently a Full Professor with Southwest Minzu University and a Ph.D. Advisor with the Institute of Computer Application, Chinese Academy of Sciences, Chengdu. He is an Inventor of Multi-scale Quantum Harmonic Oscillator Algorithm.

He holds 15 invention patents. His research interests include computational intelligence, cloud computing, high-performance computing, and quantum-inspired algorithms. He has published more than 110 papers in authoritative journals and conferences in above areas and has authored/co-authored 14 books. He has been a Commissioner of the High Performance Computing of China Computer Federation, since 2008, and a Commissioner of Cloud Computing of The Chinese Institute of Electronics, since 2010.

GANG XIN received the B.E. degree in electrical engineering and automation from Chongqing University, Chongqing, China, in 2006, and the M.E. degree in microelectronics and solid-state electronics from the Dresden University of Technology, Dresden, Germany, in 2011. He is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences and the Chengdu Institution of Computer Application, Chengdu, China. His current research interests include quantum heuristic algorithm and high-performance computing.

JIN JIN received the B.E. degree in computer science and technology from Henan Normal University, Xinxiang, China, in 2010, and the M.E. degree in computer technology from the Chengdu University of Information Technology, Chengdu, China, in 2013. She is currently pursuing the Ph.D. degree with the University of Chinese Academy of Sciences and the Chengdu Institution of Computer Application, Chengdu. Her current research interests include quantum heuristic algorithm, evolutionary computation, and global optimization.



YAN HUANG received the B.S. degree in computer science and technology from Jiangsu University, Nanjing, China, in 2003, the M.S. degree in computer application from Hohai University, Nanjing, China, in 2006, and the Ph.D. degree in computer software and theory from the Chengdu Institution of Computer Application, China Academy of Science, Chengdu, China, in 2016. He is currently a Lecturer with Huaiyin Normal University. His current research interests include quantum heuristic algorithm, evolutionary computation, and global optimization.