QUANTUM-INSPIRED SATIN BOWERBIRD ALGORITHM WITH BLOCH SPHERICAL SEARCH FOR CONSTRAINED STRUCTURAL OPTIMIZATION

SEN ZHANG
College of Information Science and Engineering
Guangxi University for Nationalities
Nanning 530006, China

GUO ZHOU*
Department of Science and Technology Teaching
China University of Political Science and Law
Beijing 102249, China

YONGQUAN ZHOU AND QIFANG LUO
Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis
Key Laboratory of Guangxi High Schools Complex System and Computational Intelligence
Nanning 530006, China

(Communicated by Changzhi Wu)

Abstract. To enhance the optimization ability of the satin bowerbird optimization (SBO) algorithm, in this paper, a novel quantum-inspired SBO with Bloch spherical search is proposed. In this algorithm, satin bowerbirds are encoded using qubits described on the Bloch sphere, each satin bowerbird occupies three locations in the search space and each location represents an optimization solution. Using the search method of general SBO to adjust the two parameters of the qubit, qubit rotation is performed on the Bloch sphere, which simultaneously updates the three locations occupied by a qubit and quickly approaches the global optimal solution. Finally, the experimental results of five examples of structural engineering design show that the proposed algorithm is superior to other state-of-the-art metaheuristic algorithms in terms of the performance measures.

1. Introduction. Quantum computation is a novel inter-discipline that includes quantum mechanics and information science. Since D. Deutsch first proposed the Deutsch–Jozsa algorithm in 1985 [11], quantum computation many researchers have paid attention to quantum computation. For example, in 2000, Han K M, et al. proposed a new intelligent algorithm that combines quantum coding with the genetic algorithm (GA) [15], which uses quantum bits to represent individuals and quantum revolving gates to update individual positions.

In 2003, Zhang G, et al. proposed a novel parallel quantum GA [43]. Hui C, et al. proposed a chaos updating rotated gates quantum-inspired GA [17]. In 2004, Wang
L, et al. used a hybrid GA based on quantum computing for numerical optimization and parameter estimation [40]. In 2008, Li Panchi, et al. used a quantum-inspired evolutionary algorithm for continuous space optimization based on the Bloch coordinates of qubits [25]. Additionally, the rotation angle of the quantum revolving gate is fixed, and it is not sufficiently flexible to solve some complex problems. Some quantum intelligent algorithms are expressed using vectors, and the quantum characteristics are not sufficiently obvious. In this paper, a satin blue gardener optimization algorithm using Bloch spherical coding is proposed, which contains two adjustable parameters. To verify its superiority, it is applied to solve constrained structural optimization problems.

With the development of intelligent algorithms, many scholars have proposed many new intelligent algorithms, such as the GA [10], particle swarm optimization (PSO) [21] ant colony optimization (ACO) [12], differential evolution (DE) [39] and artificial bee colony algorithm (ABC) [19]. In recent years, new metaheuristic algorithms have been proposed, such as cuckoo search (CS) [14], grey wolf optimization (GWO) [32], dragonfly algorithm (DA) [31] and flower pollination algorithm (FPA) [42]. Seyed H. S. M, et al. proposed the satin bowerbird optimization algorithm in 2017 [33]. To enhance the optimization ability of the satin bowerbird optimization (SBO) algorithm, in this paper, a novel quantum-inspired SBO with the Bloch coordinates of quantum bits (QBSBO) is proposed. In the proposed algorithm, satin bowerbirds are encoded by qubits described on the Bloch sphere, each satin bowerbird occupies three locations in the search space and each location represents an optimization solution. Using the search method of general SBO to adjust the two parameters of the qubit, qubit rotation is performed on the Bloch sphere, which simultaneously updates the three locations occupied by a qubit and quickly approaches the global optimal solution. The experimental results of five examples of application structural engineering design show that the proposed algorithm is superior to other state-of-the-art metaheuristic algorithms in terms of the performance measures.

The remainder of this paper is organized as follows: In Section 2, the basic principle of SBO algorithm is introduced. In Section 3, the basic principle of Bloch spherical coding is introduced. QBSBO is used to solve constrained structural optimization problems in Section 4. In Section 5, this paper is summarized and future work is proposed.

2. Satin bowerbird optimizer (SBO). According to the habits of satin bowerbirds, males need to build nests to attract females and reproduce their offspring. The location of the nest determines its attractiveness to females. In the process of nesting, males destroy the nests built by other males. All males tend to learn from the best nests. To summarize, the SBO algorithm has the following steps.

2.1. Random initialization of the bowerbird’s position. The number of individual nests is randomly generated between the upper and lower bounds. Each nest is a D-dimensional vector. The dimension of parameter D is the same as the number of parameters required to solve the optimization problem.

2.2. Calculation of the attraction probability for each bowerbird. The nests built by males have an attraction probability, which determines whether females are attracted to them. The higher the probability, the easier it is for nests to attract females. The probability formula is
SATIN BOWERBIRD ALGORITHM

\[
\text{prob}_i = \frac{\text{fit}_i}{\sum_{\text{NB}=1}^{\text{fit}}} (1)
\]

where \( \text{fit}_i \) is the fitness of the \( i \)-th solution and \( \text{NB} \) is the number of bowers. The value of \( \text{fit}_i \) is obtained using

\[
\text{fit}_i = \begin{cases} 
\frac{1}{1+f(x_i)} & f(x_i) \geq 0 \\
1+|f(x_i)| & f(x_i) < 0 
\end{cases} (2)
\]

where \( f(x_i) \) is the value of the cost function of the \( i \)-th position or \( i \)-th bird. The cost function is a function that should be optimized.

2.3. Bowerbird position update. Many intelligent algorithms adopt an elite strategy, which retains the position and fitness of the best individual in each iteration of the algorithm. For the SBO algorithm, as the number of nests built by males increases, the experience of nest building also increases. Experienced males are better at building attractive nests. In each iteration, the most experienced males are preserved, and other males learn from the most experienced birds. Elite hero birds affect other male nests.

In each cycle of the algorithm, new changes at any bower are calculated according to

\[
x_{\text{new}}^{ik} = x_{\text{old}}^{ik} + \lambda_k \left( \frac{x_{jk} + x_{\text{elite},k}}{2} - x_{\text{old}}^{ik} \right) (3)
\]

where \( x_i \) is the \( i \)-th bower or solution vector and \( x_{ik} \) is the \( k \)-th member of this vector. \( x_i \) is determined as the target solution among all solutions in the current iteration. In Eq. 3, the value \( j \) is calculated based on probabilities derived from positions. In fact, the value \( j \) is calculated using the roulette wheel procedure, which means that the larger the probability, the greater chance the solution has to be selected as \( x_j \). \( x_{\text{elite}} \) indicates the position of the elite individual, which is saved in each cycle of the algorithm. In fact, the position of the elite is the position of the bowerbird whose fitness is the highest in the current iteration.

Parameter \( \lambda_k \) determines the attraction power of the goal bowerbird, where \( \lambda_k \) determines the number of step, which is calculated for each variable. This parameter is

\[
\lambda_k = \frac{\alpha}{1+p_j} (4)
\]

where \( \alpha \) is the greatest step size and \( p_j \) is the probability obtained by Eq. 1 using the goal bower.

2.4. Bowerbird mutation. In nature, when males are busy building a bower on the ground, they may be attacked by other animals or be completely ignored. In many cases, stronger males steal materials from weaker males, or even destroy their bowers. Hence, at the end of each cycle of the algorithm, random changes are applied with a certain probability. In this step, random changes are applied to \( x_{ik} \) with a certain probability. In this case, for the mutation process, a normal distribution \((N)\) is used with an average of \( x_{\text{old}}^{ik} \) and variance of \( \sigma^2 \)

\[
x_{\text{new}}^{ik} = N(x_{\text{old}}^{ik}, \sigma^2) (5)
\]
To help understanding and calculation, the probability formula is transformed into the following formula

$$N(x_{ik}^{old}, \sigma^2) = x_{ik}^{old} + (\sigma * N(0, 1))$$

(6)

where the value of $\sigma$ is a proportion of the space width and calculated as

$$\sigma = z * (\text{var}_{\text{max}} - \text{var}_{\text{min}})$$

(7)

where $\text{var}_{\text{max}}$ and $\text{var}_{\text{min}}$ are the upper bound and lower bound assigned to variables, respectively. The $z$ parameter is the percentage difference between the upper and lower limits, which is variable.

3. Principle of QBSBO.

3.1. Spherical description of qubits. In quantum computing, a qubit is a two-level quantum system, which is described by a two-dimensional complex Hilbert space as shown in Figure 1. From the superposition principles, any state of the qubit can be expressed as follows

$$|\phi\rangle = \cos \frac{\theta}{2} |0\rangle + e^{i\phi} \sin \frac{\theta}{2} |1\rangle$$

(8)

where $0 \leq \theta \leq \pi$, $0 \leq \phi \leq 2\pi$.

Therefore, unlike the classical bit, which can only be set to 0 or 1, the qubit resides in a vector space parameterized by the continuous variables $\theta$ and $\phi$. Thus, continuous states are allowed. The Bloch sphere representation is useful for considering qubits because it provides a geometric picture of the qubit and the transformations that can be performed on the state of the qubit. Due to the normalization condition, the qubit’s state can be represented by a point on a sphere of unit radius called the Bloch sphere. This sphere can be embedded in a three-dimensional space of Cartesian coordinates ($x = \cos \phi \sin \theta, y = \sin \phi \sin \theta, z = \cos \theta$). By definition, a Bloch vector is a vector whose components ($x, y, z$) single out a point on the Bloch sphere. We can say that the angles and define a Bloch vector, as shown in Fig. 1.

In Fig. 1 if $x = \cos \phi \sin \theta, y = \sin \phi \sin \theta$ and $z = \cos \theta$, then the quantum state can be written as
This sphere can be embedded in a three-dimensional space of Cartesian coordinates described as $|A\rangle = [1, 0]^T$, $|B\rangle = [1, 0]^T$, $|C\rangle = |E\rangle = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]^T$, $|D\rangle = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$, $|F\rangle = [\frac{1}{\sqrt{2}}, -\frac{1}{\sqrt{2}}]^T$ and $|G\rangle = [\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}}]^T$.

3.2. **Encoding method for QBSBO.** In QBSBO, all satin bowerbirds are encoded by qubits described on the Bloch sphere. Set the swarm size to $NB$ and the encoding method for QBSBO.

3.3. **Satin bowerbird position update.** Nest renewal is achieved by the quantum revolving gate $U$, and the optimal solution is obtained by updating the location of the nest on the sphere. The formula of the quantum revolving gate is as follows

$$U = \begin{bmatrix}
\cos \Delta \varphi \cos \Delta \theta & -\sin \Delta \varphi \cos \Delta \theta & \sin \Delta \theta \cos (\varphi + \Delta \varphi) \\
\sin \Delta \varphi \cos \Delta \theta & \cos \Delta \varphi \cos \Delta \theta & \sin \Delta \theta \sin (\varphi + \Delta \varphi) \\
-\sin \Delta \theta & -\tan(\frac{\varphi}{2}) \sin \Delta \theta & \cos \Delta \theta
\end{bmatrix}$$

with

$$U \begin{bmatrix} \cos \varphi \sin \theta \\ \sin \varphi \sin \theta \\ \cos \theta \end{bmatrix} = \begin{bmatrix} \cos(\varphi + \Delta \varphi) \sin(\theta + \Delta \theta) \\ \sin(\varphi + \Delta \varphi) \sin(\theta + \Delta \theta) \\ \cos(\theta + \Delta \theta) \end{bmatrix}$$

In Eq. 12, the rotation of the quantum revolving gate $U$ updates the quantum bit through $\Delta \varphi$ and $\Delta \theta$. The parameters $\Delta \varphi$ and $\Delta \theta$ are updated using

$$\Delta \varphi_{ik}^{\text{new}} = \Delta \varphi_{ik}^{\text{old}} + \lambda_k \left( \frac{\Delta \varphi_{j,k} + \Delta \varphi_{\text{elite},k}}{2} - \Delta \varphi_{ik}^{\text{old}} \right)$$

$$\Delta \theta_{ik}^{\text{new}} = \Delta \theta_{ik}^{\text{old}} + \lambda_k \left( \frac{\Delta \theta_{j,k} + \Delta \theta_{\text{elite},k}}{2} - \Delta \theta_{ik}^{\text{old}} \right)$$

3.4. **Satin bowerbird mutation.** To improve the individual diversity of QBSBO from iteration period to the later stage and prevent it from falling into a local optimum, the satin bowerbird’s nest mutates according to a certain probability. The quantum non-gate $V$ mutates an individual qubit as follows

$$V = \begin{bmatrix} 0 & \tan(\frac{\varphi}{2} - \theta) & 0 \\ \tan(\frac{\varphi}{2} - \theta) & 0 & 0 \\ 0 & 0 & \tan \theta \end{bmatrix}$$

with

$$V = \begin{bmatrix} \cos \varphi \sin \theta \\ \sin \varphi \sin \theta \\ \cos \varphi \sin \theta \end{bmatrix} = \begin{bmatrix} \cos(\varphi) \cos(\frac{\varphi}{2} - \theta) \\ \sin(\varphi) \sin(\frac{\varphi}{2} - \theta) \\ \cos(\varphi) \cos(\frac{\varphi}{2} - \theta) \end{bmatrix}$$
From Eq. 16, the purpose of $V$ is to make the nest position rotate along the sphere in the late iteration period. The rotation angle is large, which is conducive to determining the optimal position.

3.5. **Projection measurement of qubits.** Because the optimization problem is real, a quantum bit is projected onto the $x$-axis, $y$-axis and $z$-axis. The projection formula is as follows

\[
\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}
\]

(17)

For the $i$th individual $j$th qubit $|\varphi_{ij}\rangle$, the projection of the coordinates of the measurement is

\[
\begin{align*}
    x_{ij} &= \langle \varphi_{ij}|\sigma_x|\theta_{ij}\rangle \\
    y_{ij} &= \langle \varphi_{ij}|\sigma_y|\theta_{ij}\rangle \\
    z_{ij} &= \langle \varphi_{ij}|\sigma_z|\theta_{ij}\rangle
\end{align*}
\]

(18)

where $i = 1, 2, \ldots, NB$ and $j = 1, 2, \ldots, D$. Here, QBSBO individuals are composed of quantum bits, whose coordinates on the spherical $X$, $Y$, $Z$ axes are regarded as three parallel genes. Hence, QBSBO individuals have three gene chains.

3.6. **Solution space transformation.** In QBSBO, each individual contains $3n$ Bloch coordinates of $n$ qubits that can be transformed from unit space $[-1, 1]^n$ to the solution space of the continuous optimization problem. Each Bloch coordinate corresponds to an optimization variable in the solution space. Let the $j$th variable of the optimization problem be $X_i \in [A_i, B_i]$, and $(x_{ij}, y_{ij}, z_{ij})$ denotes the coordinates of the $j$th qubit for the $i$th individual. Then the corresponding variables $(X_{ij}, Y_{ij}, Z_{ij})$ in the solution space are computed as follows, respectively

\[
\begin{align*}
    X_{ij} &= \frac{1}{2}[B_j(1 + x_{ij}) + A_j(1 - x_{ij})] \\
    Y_{ij} &= \frac{1}{2}[B_j(1 + y_{ij}) + A_j(1 - y_{ij})] \\
    Z_{ij} &= \frac{1}{2}[B_j(1 + z_{ij}) + A_j(1 - z_{ij})]
\end{align*}
\]

(19) (20) (21)

where $i = 1, 2, \ldots, NB$ and $j = 1, 2, \ldots, D$.

3.7. **Optimal solution is updated.** The three solutions

\[
(X_{i1}, X_{i2}, \ldots, X_{i,D}), (Y_{i1}, Y_{i2}, \ldots, Y_{i,D}), (Z_{i1}, Z_{i2}, \ldots, Z_{i,D})
\]

described by the $i$th individual are substituted into the fitness function. Let $\text{fit}_{\text{best}}$ denote the best fitness so far. Then the optimal solution is obtained

\[
\text{fit}_{\text{best}} = \min(f(X_{ij}), f(Y_{ij}), f(Z_{ij}))
\]

(22)

The pseudocode of the QBSBO algorithm is shown in Algorithm 1.

**Algorithm 1** QBSBO algorithm

1. Initialize the population $X_i$ according to Eq. 10, where $i = 1, 2, \ldots, NB$
2. Evaluate the fitness of each individual (bird nest) by Eq. 2
3. Find the best $\Delta \varphi$ and $\Delta \theta$
4. while the end criterion is not satisfied
5. Calculate the probability of an individual by Eq. 1
6. for every bower
7. Update $\Delta \varphi$ and $\Delta \theta$ by Eq. 13 and Eq. 14
8. Update the individual position phase using the quantum rotating gate
9. end for
10. Rank individuals according to the fitness values
11. Update the elite individual
12. end while
13. Return the best individual by Eq. 22

4. QBSBO for constrained structural optimization problems. In this section, the QBSBO algorithm is used to solve some constrained structural problems, including pressure vessel design [31], welded beam design [27], tensile/compression spring design [35, 2], cantilever Beam design [4] and reducer design [13].

4.1. Pressure vessel design. The optimization algorithm is used to optimize the parameters of the pressure vessel design problem to minimize the objective function, as shown in Figure 2. The objective function of the pressure vessel design problem and the constraints of related parameters are as follows. Consider

$$\vec{x} = [x_1, x_2, x_3, x_4] = [T_s, T_h, R, L]$$

Minimize

$$f(\vec{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.8x_1^2x_3$$

Subject to

$$g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0$$
$$g_2(\vec{x}) = -x_3 + 0.00954x_3 \leq 0$$
$$g_3(\vec{x}) = -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0$$
$$g_4(\vec{x}) = x_4 - 240 \leq 0$$

with variables in range
In Eq. 23, the main parameters are the thickness of the shell ($T_s$), thickness of the head ($T_h$), inner radius ($R$) and length of the cylindrical section without considering the head ($L$).

$$0 \leq x_1 \leq 99$$
$$0 \leq x_2 \leq 99$$
$$10 \leq x_3 \leq 200$$
$$10 \leq x_4 \leq 200$$

Table 1 shows that by comparing the QBSBO algorithm with the SBO, GA, GSA, PSO and other algorithms, the QBSBO algorithm is more accurate under constraints. The QBSBO algorithm has obvious advantages in the solving pressure vessel design problem. Table 2 shows that the QBSBO algorithm has a smaller average and standard deviation than the original algorithm, and the best results are more accurate than those of the original algorithm, which shows the advantages of the improved algorithm.

**Table 1** Comparison results for the pressure vessel design problem with optimal variables

| Algorithms                  | $T_s$ | $T_h$ | $R$ | $L$ | Optimal cost        |
|-----------------------------|-------|-------|-----|-----|---------------------|
| GSA [37]                    | 1.125 | 0.625 | 55.988 | 84.454 | 8538.835           |
| PSO (He and Wang) [16]      | 0.812 | 0.437 | 42.091 | 176.740 | 6081.077           |
| GA (Coello) [27]            | 0.812 | 0.434 | 40.321 | 200.000 | 6288.745           |
| GA (Deb and Gane) [9]       | 0.937 | 0.500 | 48.329 | 112.679 | 6410.383           |
| ES (Montes and Coello) [29] | 0.812 | 0.437 | 42.098 | 176.664 | 6059.94           |
| DE (Huang et al.) [24]      | 0.812 | 0.437 | 42.098 | 176.640 | 6059.74           |
| ACO (Kaveh and Talataheri) [20] | 0.812 | 0.437 | 42.103 | 176.572 | 6059.08           |
| Lagrangian Multiplier (Kannan) [18] | 1.125 | 0.625 | 58.291 | 43.6900 | 7198.04           |
| Branch-bound (Sandgren) [38] | 1.125 | 0.625 | 58.700 | 117.701 | 8129.10           |
| SBO [33]                    | 0.940 | 0.468 | 46.085 | 116.438 | 6176.02           |
| D-DS [26]                   | 0.8125 | 0.4375 | 42.098 | 176.637 | 6059.71           |
| QBSBO                       | 0.835 | 0.411 | 43.952 | 164.816 | **5998.92**        |

**Table 2** Comparison statistical results for the pressure vessel design problem

| Algorithms | Best | Worst | Average | Std     |
|------------|------|-------|---------|---------|
| D-DS [26]  | 6059.71 | 6410.02 | 6121.42 | 23.81   |
| SBO [33]   | 6176.02 | 7113.5 | 6792.53 | 257.96  |
| QBSBO      | 5998.67 | 7069.8 | 6434.56 | 189.75  |
4.2. Welded beam design problem. The optimization algorithm is used to optimize the parameters of the welded beam design problem to minimize the objective function, as shown in Figure 3. The objective function of the pressure vessel design problem and the constraints of related parameters are as follows. Consider

$$\vec{x} = [x_1, x_2, x_3, x_4] = [h, l, t, b]$$

Minimize

$$f(\vec{x}) = 1.1047x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

Subject to

$$g_1(\vec{x}) = \tau(\vec{x}) - \tau_{\text{max}} \leq 0$$
$$g_2(\vec{x}) = \sigma(\vec{x}) - \sigma_{\text{max}} \leq 0$$
$$g_3(\vec{x}) = \delta(\vec{x}) - \delta_{\text{max}} \leq 0$$
$$g_4(\vec{x}) = x_3 - x_4 \leq 0$$
$$g_5(\vec{x}) = P - P_0(\vec{x}) \leq 0$$
$$g_6(\vec{x}) = 0.125 - x_1 \leq 0$$
$$g_7(\vec{x}) = 1.1047x_1^2 + 0.04811x_3x_4(14.0 + x_2) - 5.0 \leq 0$$

with variables in range

0.1 ≤ x_1 ≤ 2
0.1 ≤ x_2 ≤ 10
0.1 ≤ x_3 ≤ 10
0.1 ≤ x_4 ≤ 2

where $\tau(\vec{x}) = \sqrt{\tau' + 2\tau''x_1x_2} + (\tau''), \tau' = \frac{P}{\sqrt{2x_1x_2}}, \tau'' = \frac{M}{E}x_2, M = P(L + \frac{x_4}{2}), R = \sqrt{\frac{x_2^2}{4} + (\frac{x_4}{2} + x_3)^2}, J = 2\left(\sqrt{2x_1x_2}x_2^2 + (\frac{x_4}{2} + x_3)^2\right), \sigma(\vec{x}) = \frac{6P}{Lx_3x_4}, \delta(\vec{x}) = \frac{6P}{L^3}x_3 \cdot x_4, P_0(\vec{x}) = \frac{4.103E}{L^2} \left(1 - \frac{x_1}{2L} \sqrt{\frac{E}{40}}\right), P = 6000lb, L = 14in, \delta_{\text{max}} = 0.25in, E = 30 \times 10^6 psi, G = 12 \times 10^6 psi, \tau_{\text{max}} = 13.600psi, \sigma_{\text{max}} = 3000psi$.

In Eq. 24, the four variables thickness of the weld h, length of the clamped bar l, height of the bar t and thickness of the bar b need to be optimized.

**Table 3** Comparison results for the pressure vessel design problem with optimal variables

| Algorithms             | h  | t  | f  | b  | Optimal cost |
|------------------------|----|----|----|----|--------------|
| GSA [37]               | 0.1821 | 3.8569 | 10.00 | 0.192 | 1.0879 |
| CPSO [22]              | 0.202 | 3.5442 | 9.0452 | 0.2057 | 1.728 |
| GA(Castro) [5]         | N/A | N/A | N/A | N/A | 2.124 |
| GA(Deb) [7]            | N/A | N/A | N/A | N/A | 2.380 |
| HS(Leand Geem) [23]    | 0.248 | 6.173 | 8.1789 | 0.2534 | 2.433 |
| Random [34]            | 0.244 | 6.223 | 8.2915 | 0.2443 | 2.380 |
| Simplex [44]           | 0.279 | 5.625 | 7.751 | 0.279 | 2.530 |
| David [33]             | 0.243 | 6.255 | 8.291 | 0.244 | 2.384 |
| Approx [34]            | 0.244 | 6.218 | 8.291 | 0.244 | 2.381 |
| SBO [33]               | 0.214 | 3.492 | 8.557 | 0.229 | 1.849 |
| D-DS [26]              | 0.206 | 3.253 | 9.037 | 0.206 | 1.696 |
| QBSBO                  | 0.213 | 3.619 | 8.492 | 0.233 | 1.820 |

**Table 4** Comparison statistical results for the welded beam design problem

| Algorithms | Best | Worst | Average | Std |
|------------|------|-------|---------|-----|
| D-DS [26]  | 1.695 | 1.695 | 1.695   | 1.94e-06 |
| SBO        | 1.849 | 3.046 | 2.532   | 0.4280 |
| QBSBO      | 1.826 | 2.153 | 1.903   | 0.1364 |
Table 3 also shows that the QBSBO algorithm has the same advantages as those for the solving pressure vessel design problem, and has a higher accuracy than the SBO, GA, GSA, PSO and other algorithms. Table 4 shows that the average and standard deviation of the QBSBO algorithm are smaller than those of the original algorithm, which shows the effectiveness of the improved algorithm.

4.3. Tension compression spring design problem. The QBSBO algorithm is used to solve the spring optimization problem as shown in Figure 4, and the performance of objective function minimization under constraints is verified. The objective function and parameter constraints are as follows. Consider

\[ \overrightarrow{x} = [x_1, x_2, x_3, x_4] = [d, D, N] \]

Minimize

\[ f(\overrightarrow{x}) = (x_3 + 2)x_2x_1^2 \]

Subject to

\[
\begin{align*}
g_1(\overrightarrow{x}) &= 1 - \frac{x_2^2}{2785x_1^2} \leq 0 \\
g_2(\overrightarrow{x}) &= \frac{4x_2^3 - x_1x_2}{2560(x_2x_1^2 - x_1^3)} + \frac{1}{5108x_2^2} \leq 0 \\
g_3(\overrightarrow{x}) &= 1 - \frac{45x_1^3x_2}{256x_2^3x_3} \leq 0 \\
g_4(\overrightarrow{x}) &= \frac{x_1 + x_2}{1.5} - 1 \leq 0
\end{align*}
\] (25)

with variables in range

\[ 0.05 \leq x_1 \leq 2.00 \\
0.25 \leq x_2 \leq 1.30 \\
2.00 \leq x_3 \leq 15.0 \]

In Eq. 25, the variables wire diameter \( d \), mean coil diameter \( D \) and length (or number of coils) \( N \) need to be optimized.

Table 5 Comparison results for the compression spring design problem

| Algorithms                        | \( d \)   | \( D \)   | \( N \)   | Optimal cost |
|-----------------------------------|----------|----------|----------|--------------|
| GSA [37]                          | 0.0502   | 0.3236   | 13.525   | 0.0127       |
| PSO(Hs and Wang) [16]             | 0.0517   | 0.3576   | 11.244   | 0.01267      |
| ES(Coello and Montes) [29]        | 0.0519   | 0.3639   | 10.890   | 0.01268      |
| GA (Coello) [27]                  | 0.0514   | 0.3566   | 11.632   | 0.01270      |
| Montes and Coello [24]            | 0.0516   | 0.3556   | 11.397   | 0.0126       |
| Constraint correction (Arora) [20]| 0.0500   | 0.3159   | 14.250   | 0.0128       |
| Mathematical optimization (Belegundu) [2] | 0.0533   | 0.3991   | 9.1854   | 0.0127       |
| SBO [33]                          | 0.0556   | 0.4634   | 7.0054   | 0.0131       |
| D-DS [26]                         | 0.0527   | 0.356    | 11.3434  | 0.0127       |
| QBSBO                             | 0.0513   | 0.357    | 11.28    | 0.0127       |
As shown in Table 5, the QBSBO algorithm also has advantages in solving the pressure spring problem. Table 6 shows that the accuracy of the original algorithm is improved.

4.4. Cantilever beam design problem. The QBSBO algorithm is used to solve the cantilever beam design problem, as shown in Figure 5. The objective function and related parameters are constrained as follows. Consider

$$\vec{x} = [x_1, x_2, x_3, x_4, x_5]$$

Minimize

$$f(\vec{x}) = 0.6224(x_1 + x_2 + x_3 + x_4 + x_5)$$

Subject to

$$g(\vec{x}) = \frac{61}{x_1^4} + \frac{37}{x_2^3} + \frac{19}{x_3^3} + \frac{7}{x_4^3} + \frac{1}{x_5^3} \leq 1$$

with variables in range

$$0.01 \leq x_1, x_2, x_3, x_4, x_5 \leq 100$$

In Eq. 26, the variables $x_1, x_2, x_3, x_4, x_5$ need to be optimized.
Table 7 also shows that the QBSBO algorithm has the same advantages over other commonly used algorithms in solving the cantilever problem. The QBSBO algorithm in Table 8 has a smaller average value and better quality than the original algorithm, which verifies the effectiveness of the improvement.

4.5 Speed reducer design problem. We continue to use the QBSBO algorithm to test and solve the reducer design problem as shown in Figure 6, and verify the effect of its improvement. The objective function is as follows. Minimize

\[
f(\vec{x}) = 0.7854x_1x_2^2(3.3333x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) \]

\[
+ 7.4777(x_3^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \]

Subject to

\[
g_1(\vec{x}) = -x_1 + 0.0193x_3 \leq 0
\]

\[
g_2(\vec{x}) = \frac{397.5 - 1}{x_1x_2x_3} \leq 0
\]

\[
g_3(\vec{x}) = 1.93x_2 - 1 \leq 0
\]

\[
g_4(\vec{x}) = \frac{1.93x_2}{x_2x_3} - 1 \leq 0
\]

\[
g_5(\vec{x}) = \sqrt{\frac{(745x_4/x_3x_5)^2 + 16.9 + 10^6}{110x_4}} - 1 \leq 0
\]

\[
g_6(\vec{x}) = \frac{(745x_5/x_2x_3)^2 + 157.5 + 10^6}{85x_5} - 1 \leq 0
\]

\[
g_7(\vec{x}) = \frac{x_2x_3}{30} - 1 \leq 0
\]

\[
g_8(\vec{x}) = \frac{x_2x_3}{30} - 1 \leq 0
\]

\[
g_9(\vec{x}) = \frac{x_1}{\frac{12x_2}{1.5x_6+1.0}} - 1 \leq 0
\]

\[
g_{10}(\vec{x}) = \frac{1.1x_5+1.7}{x_5} - 1 \leq 0
\]

where 2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 \leq 8.3, 2.9 \leq x_6 \leq 3.9, 5.0 \leq x_7 \leq 5.5.

### Table 9

| Algorithms      | 4.50916 | 0.7090 | 28 | 7.2891 | 7.8594 | 3.4695 | 5.2891 | 3008.08 |
|-----------------|---------|--------|----|--------|--------|--------|--------|---------|
| Mezura-Montes et al. [30] | 4.50916 | 0.7090 | 28 | 7.2891 | 7.8594 | 3.4695 | 5.2891 | 3008.08 |
| HCPS [26]       | 0.5     | 0.7    | 17 | 3.8    | 7.7134 | 3.4592 | 5.3305 | 2994.47 |
| SCA [36]        | 4.48806 | 0.7    | 17 | 3.87602 | 7.7134 | 3.4592 | 5.3305 | 2994.47 |
| ABC [32]        | 8.49999 | 0.7    | 17 | 3.87602 | 7.7134 | 3.4592 | 5.3305 | 2994.47 |
| SBO             | 3.5046  | 0.7090 | 17 | 7.0576 | 7.3000 | 3.5779 | 5.2935 | 2998.9 |
| QBSBO           | 3.60900 | 0.7090 | 17 | 7.30000 | 7.3000 | 3.5902 | 5.28652 | 2988.142 |

### Table 10

| Algorithms | Best | Worst | Average | Std |
|------------|------|-------|---------|-----|
| SBO        | 2998.9 | 3108.540 | 3065.71 | 23.017 |
| QBSBO      | 2985.185 | 3366.970 | 3084.75 | 144.36 |

The experimental results show that the QBSBO algorithm is also effective in solving the speed reducer design problem. Table 10 shows that the QBSBO algorithm has higher convergence accuracy than the original algorithm. Through simulation experiments, the QBSBO algorithm was used to solve five types of engineering optimization problems. A comparison of the accuracy and average value of the results verified that the QBSBO algorithm is superior to other commonly used algorithms. In fact, no algorithm is perfect and can solve all optimization problems. QBSBO still has some shortcomings regarding solving individual problems; however, through simulation experiments for constrained structural problems, it is sufficient to show that QBSBO has the ability to deal with relatively complex problems.
5. **Conclusions and future work.** In this paper, we proposed a quantum coding gardener algorithm based on the Bloch sphere, which was improved based on the SBO algorithm. Quantum coding for an individual in the algorithm improved the convergence speed and accuracy of the original algorithm, improved the diversity of the population and effectively avoided local convergence. The simulation results showed that the QBSBO algorithm has certain advantages in dealing with some complex problems. In future work, the application scope of the QBSBO algorithm will be expanded and more practical problems will be solved that are difficult to solve using more complex common methods.

**Acknowledgments.** This work was supported by the National Science Foundation of China under Grant No. 61563008, the Project of Guangxi Natural Science Foundation under Grant No. 2018GXNSFAA138146, the Project of China University of Political Science and Law Research Innovation under Grant No. 10818441, and the Young Scholar Fund of China University of Political Science and Law under Grant No. 10819144.

**REFERENCES**

[1] S. Akhtar, K. Tai and T. Ray, A socio-behavioural simulation model for engineering design optimization, *Engineering Optimization*, 34 (2002), 341–354.

[2] A. D. Belegundu and J. S. Arora, A study of mathematical programming methods for structural optimization. Part I: Theory, *International Journal for Numerical Methods in Engineering*, 21 (1985), 1583–1599.

[3] M. Y. Cheng and D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, *Computers & Structures*, 139 (2014), 98–112.

[4] H. Chickermane and H. C. Gea, Structural optimization using a new local approximation method, *International Journal for Numerical Methods in Engineering*, 39 (1996), 829–846.

[5] C. A. C. Coello, Constraint-handling using an evolutionary multiobjective optimization technique, *Civil Engineering Systems*, 17 (2000), 319–346.

[6] C. A. C. Coello and E. M. Montes, Constraint-handling in genetic algorithms through the use of dominance-based tournament selection, *Advanced Engineering Informatics*, 16 (2002), 193–203.

[7] K. Deb, Optimal design of a welded beam via genetic algorithms, *AIAA Journal*, 29 (1991), 2013–2015.

[8] K. Deb, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186 (2000), 311–338.

[9] K. Deb, GeneAS: A robust optimal design technique for mechanical component design, in *Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, Heidelberg, 1997, 497–514.

[10] K. Deb, A. Pratap and S. Agarwal, A fast and elitist multiobjective genetic algorithm: NSGA-II, *IEEE Transactions on Evolutionary Computation*, 6 (2002), 182–197.
[11] D. Deutsch, Quantum theory, the Church-Turing principle and the universal quantum computer, Proceedings of the Royal Society of London Series A, 400 (1985), 97–117.
[12] M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization, IEEE Computational Intelligence Magazine, (2006), 28–39.
[13] A. H. Gandomi and X. S. Yang, Benchmark problems in structural optimization, in Computational Optimization, Methods and Algorithms, Springer, Berlin, 2011, 259–281.
[14] A. H. Gandomi, X. S. Yang and A. H. Alavi, Cuckoo search algorithm: A metaheuristic approach to solve structural optimization problems, Engineering with Computers, 29 (2013), 17–35.
[15] K. H. Han and J. H. Kim, Genetic quantum algorithm and its application to combinatorial optimization problem, Proceedings of the 2000 Congress on Evolutionary Computation, 2 (2000), 1354–1360.
[16] Q. He and L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Engineering Applications of Artificial Intelligence, 20 (2007), 89–99.
[17] C. Hui, Z. Jiashu and Z. Chao, Chaos updating rotated gates quantum-inspired genetic algorithm, Communications, Circuits and Systems, 2004 International Conference on Communications, Circuits and Systems, Chengdu, 2 (2004), 1108–1112.
[18] B. K. Kannan and S. N. Kramer, An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, Journal of Mechanical Design, 116 (1994), 405–411.
[19] D. Karaboga and B. Basturk, A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm, Journal of Global Optimization, 2007, 39(4), 459–471.
[20] A. Kaveh and S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Engineering Computations, 27 (2010), 155–182.
[21] J. Kennedy, Particle Swarm Optimization. Encyclopedia of Machine Learning, Springer, Boston, MA, 2011, 760–766.
[22] R. A. Krohling and L. dos Santos Coelho, Coevolutionary particle swarm optimization using Gaussian distribution for solving constrained optimization problems, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), 36 (2006), 1407–1416.
[23] K. S. Lee and Z. W. Geem, A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice, Computer Methods in Applied Mechanics and Engineering, 194 (2005), 3902–3933.
[24] L. J. Li, Z. B. Huang and F. Liu, A heuristic particle swarm optimizer for optimization of pin connected structures, Computers & Structures, 85 (2007), 340–349.
[25] P. Li and S. Li, Quantum-inspired evolutionary algorithm for continuous space optimization based on Bloch coordinates of qubits, Neurocomputing, 72 (2008), 581–591.
[26] J. Liu, C. Wu, G. Wu and X. Wang, A novel differential search algorithm and applications for structure design, Applied Mathematics and Computation, 268 (2015), 246–269.
[27] F. S. Lobato and V. Steffen, Jr., Fish swarm optimization algorithm applied to engineering system design, Latin American Journal of Solids and Structures, 11 (2014), 143–156.
[28] W. Long, W. Zhang, Y. Huang and Y. Chen, A hybrid cuckoo search algorithm with feasibility-based rule for constrained structural optimization, Journal of Central South University, 21 (2014), 3197–3204.
[29] E. Mezura-Montes and C. A. C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, International Journal of General Systems, 37 (2008), 443–473.
[30] E. Mezura-Montes, C. A. C. Coello and R. Landa-Becerra, Engineering optimization using simple evolutionary algorithm, in Proceedings. 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, CA, 2003, 149–156.
[31] S. Mirjalili, Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Computing and Applications, 27 (2016), 1053–1073.
[32] S. Mirjalili, S. M. Mirjalili and A. Lewis, Grey wolf optimizer, Advances in Engineering Software, 69 (2014), 46–61.
[33] S. H. S. Moosavi and V. K. Bardsiri, Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation, Engineering Applications of Artificial Intelligence, 60 (2012), 1–15.
[34] K. M. Ragsdell and D. T. Phillips, Optimal design of a class of welded structures using geometric programming, *Journal of Manufacturing Science and Engineering*, 98 (1976), 1021–1025.

[35] S. S. Rao, Engineering Optimization: Theory and Practice, John Wiley & Sons, Inc., New York, 2009.

[36] T. Ray and K. M. Liew, Society and civilization: An optimization algorithm based on the simulation of social behavior, *IEEE Transactions on Evolutionary Computation*, 7 (2003), 386–396.

[37] E. Rashidi, H. Nezamabadi-Pour and S. Saryazdi, GSA: A gravitational search algorithm, *Information Sciences*, 179 (2009), 2232–2248.

[38] E. Sandgren, Nonlinear integer and discrete programming in mechanical design optimization, *Journal of Mechanical Design*, 112 (1990), 223–229.

[39] R. Storn and K. Price, Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces, *Journal of Global Optimization*, 11 (1997), 341–359.

[40] L. Wang, F. Tang and H. Wu, Hybrid genetic algorithm based on quantum computing for numerical optimization and parameter estimation, *Applied Mathematics and Computation*, 171 (2005), 1141–1156.

[41] D. H. Wolpert and W. G. Macready, No free lunch theorems for optimization, *IEEE Transactions on Evolutionary Computation*, 1 (1997), 67–82.

[42] X. S. Yang, Flower pollination algorithm for global optimization, *Unconventional Computation and Natural Computation*, Springer, Berlin, Heidelberg, 2012, 240–249.

[43] G. Zhang, W. Jin and L. Hu, A novel parallel quantum genetic algorithm, *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, Chengdu, China, 2003, 693–697.

Received December 2019; 1st revision April 2020; 2nd revision May 2020.

E-mail address: guo.zhou@live.com
E-mail address: yongquanzhou@126.com
E-mail address: zhangsen1264@163.com
E-mail address: l.qf@163.com