

7-31-2020

## Design and Implementation of Quantum-Inspired Bee Colony Algorithm

Shuyun Yang

*1. Admission and Employment office, Northeast Petroleum University, Daqing 163318, China; ;*

Panchi Li

*2. School of Computer & Information Technology, Northeast Petroleum University, Daqing 163318, China;*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

## Design and Implementation of Quantum-Inspired Bee Colony Algorithm

### Abstract

**Abstract:** To enhance the performance of artificial bee colony algorithm, a quantum-inspired bee colony algorithm was proposed. In the proposed approach, *the bees were encoded with the qubits described on the Bloch sphere. The evolutionary search was achieved by rotating the qubit about the rotation axis on the Bloch sphere. The Bloch coordinates of qubit can be obtained by measuring with the Pauli matrices, and the optimization solutions can be presented through the solution space transformation.* The highlight advantages of this method are the ability to simultaneously adjust two parameters of a qubit and automatically achieve the best match between two adjustment quantities. The experimental results show that the proposed method obviously outperforms the classical one for some benchmark functions and the water flooded layer identification.

### Keywords

quantum computing, bee colony optimizing, Bloch sphere rotating, algorithm designing

### Recommended Citation

Yang Shuyun, Li Panchi. Design and Implementation of Quantum-Inspired Bee Colony Algorithm[J]. Journal of System Simulation, 2015, 27(7): 1480-1489.

## 量子衍生蜂群算法的设计与实现

杨淑云<sup>1</sup>, 李盼池<sup>2</sup>

(1.东北石油大学 招生就业处, 大庆 163318; 2.东北石油大学计算机与信息技术学院, 大庆 163318)

**摘要:** 为提高人工蜂群算法的优化能力, 提出一种量子衍生蜂群算法。在该算法中, 蜂群采用基于 Bloch 球面描述的量子比特编码; 采用量子比特在 Bloch 球面上的绕轴旋转实现进化搜索; 采用泡利矩阵获得量子比特的 Bloch 坐标; 通过解空间变换获得优化问题的实际解。该方法的突出优点是能够同时调整量子比特的两个参数, 并自动实现两个调整量的最佳匹配。函数极值优化及水淹层识别的实验结果表明, 该方法的优化能力比普通蜂群算法确有明显提高。

**关键词:** 量子计算; 蜂群优化; Bloch 球面旋转; 算法设计

中图分类号: TP183

文献标识码: A

文章编号: 1004-731X(2015)07-1480-10

## Design and Implementation of Quantum-Inspired Bee Colony Algorithm

Yang Shuyun<sup>1</sup>, Li Panchi<sup>2</sup>

(1. Admission and Employment office, Northeast Petroleum University, Daqing 163318, China;

2. School of Computer &amp; Information Technology, Northeast Petroleum University, Daqing 163318, China)

**Abstract:** To enhance the performance of artificial bee colony algorithm, a quantum-inspired bee colony algorithm was proposed. In the proposed approach, the bees were encoded with the qubits described on the Bloch sphere. The evolutionary search was achieved by rotating the qubit about the rotation axis on the Bloch sphere. The Bloch coordinates of qubit can be obtained by measuring with the Pauli matrices, and the optimization solutions can be presented through the solution space transformation. The highlight advantages of this method are the ability to simultaneously adjust two parameters of a qubit and automatically achieve the best match between two adjustment quantities. The experimental results show that the proposed method obviously outperforms the classical one for some benchmark functions and the water flooded layer identification.

**Keywords:** quantum computing; bee colony optimizing; Bloch sphere rotating; algorithm designing

## 引言

人工蜂群算法是由土耳其学者于 2005 年提出的一种模拟蜜蜂群体寻找优良蜜源的仿生智能优化算法<sup>[1]</sup>。与遗传算法、粒子群算法等智能计算方法相比, 该算法的突出优点是每次迭代中都进行全

局和局部搜索, 从而可在较大程度上避免早熟收敛, 加大获得最优解的概率。目前, 该算法已在数值优化<sup>[2-4]</sup>, 神经网络训<sup>[5]</sup>, 数字滤波器设计<sup>[6]</sup>, 分布式系统中的网络重构<sup>[7]</sup>, 最小生成树的构造<sup>[8]</sup>等许多领域的问题中获得成功应用。然而, 在算法改进方面的进展相对缓慢。丁海军等通过重新设计控制参数, 提出了一种求解 TSP 问题的改进人工蜂群算法<sup>[9]</sup>; 康飞等通过引入文化算法的双层进化结构和具有概率突跳特性的模拟退火操作, 提出了一种具有文化导引功能的文化退火人工蜂群算法<sup>[10]</sup>;



收稿日期: 2014-07-11 修回日期: 2014-08-09;  
基金项目: 国家自然科学基金(61170132); 黑龙江省自然科学基金(F2015021);  
作者简介: 杨淑云(1968-), 女, 黑龙江安达人, 讲师。研究方向为神经网络和智能优化算法; 李盼池(1969-), 男, 河北大城人, 教授, 博士, 研究方向为量子神经网络和量子衍生优化算法。

<http://www.china-simulation.com>

• 1480 •

段海滨等提出了一种将人工蜂群算法和量子进化相结合的算法<sup>[11]</sup>。

量子计算是一种全新的计算模型, 以其特有的多态性、叠加性、并行性等优点受到国内外学者的广泛关注。目前与遗传、免疫、蚁群、粒子群等智能优化模型的融合已获得成功应用。在真实的量子系统中, 量子比特是基于 Bloch 球面描述的, 含有 2 个可调参数。然而, 在目前的量子智能优化算法中, 个体编码采用单位圆描述的量子比特编码, 只有一个可调参数, 而进化机制采用的量子旋转门和量子非门, 本质上是使量子比特在单位圆上绕着圆心旋转, 同样只改变量子比特的一个参数, 因此, 量子特性没有得到充分体现。尽管文献[12]提出了一种基于量子比特 Bloch 坐标编码的量子遗传算法, 且该算法中量子比特基于 Bloch 球面描述, 提出的进化算子也能够同时调整量子比特的两个参数, 然而该算法没有考虑两个参数调整量的匹配问题, 即当前比特向目标比特的逼近过程不是沿最短路径, 因此, 优化性能受到影响。本文提出一种新的蜂群个体编码及进化方法。与文献[12]量子遗传算法中的编码方案不同, 本文直接采用基于 Bloch 球面描述的量子比特(不是量子比特的坐标)编码, 采用量子比特在 Bloch 球面上绕轴旋转的方法实现蜂群个体的进化搜索。该方法可自动实现蜂群个体量子比特的两个参数调整量的最佳匹配。在量子衍生蜂群算法的设计中, 详细阐述了算法的设计原理及实施方案。以标准函数极值优化为例, 对比验证了该方法的优越性。

## 1 蜂群算法

设蜜蜂总数为  $N_s$ , 其中采蜜蜂数为  $N_e$ , 跟踪蜂数为  $N_u$ , 个体向量维度为  $D$ , 个体搜索空间为  $S = R^D$ , 采蜜蜂种群空间为  $S^{N_e}$ ,  $X = (X_1, X_2, \dots, X_{N_e})$  为一个采蜜蜂群,  $X(0)$  为初始采蜜蜂群,  $X(n)$  为第  $n$  代采蜜蜂群,  $f: S \rightarrow R^+$  为目标函数。以极小值优化为例, 人工蜂群算法可描述如下。

(1) 随机生成  $N_s$  个可行解, 第  $i$  个可行解  $X_i$  为

$$X_i^j = X_{\min}^j + \text{rand}(0,1)(X_{\max}^j - X_{\min}^j) \quad (1)$$

其中  $j = 1, 2, \dots, D$ 。分别计算目标值, 并将排名前  $N_e$  的解作为初始采蜜蜂种群  $X(0)$ 。

(2) 对于当前代的采蜜蜂  $X_i$ , 在当前位置邻域内按下式搜索新位置。

$$V_i^j = X_i^j + \phi_i^j(X_i^j - X_k^j) \quad (2)$$

其中  $j \in \{1, 2, \dots, D\}$ ,  $k \in \{1, 2, \dots, N_e\}$ ,  $k \neq i$ ,  $j$  和  $k$  均随机产生,  $\phi_i^j$  为  $(0, 1)$  之间随机数。

(3) 采用贪婪选择算子在  $V_i$  和  $X_i$  之间择优选择进入下一代种群, 该算子记作  $T_s: S^2 \rightarrow S$ , 其概率分布为

$$P\{T_s(X_i, V_i) = V_i\} = \begin{cases} 1, & f(V_i) < f(X_i) \\ 0, & f(V_i) \geq f(X_i) \end{cases} \quad (3)$$

(4) 各跟踪蜂依概率在采蜜蜂种群随机选择一个采蜜蜂, 并在其邻域搜索新位置。该算子记作  $T_{s1}: S^{N_e} \rightarrow S$ , 其概率分布为

$$P\{T_{s1}(X) = X_i\} = f(X_i) / \sum_{m=1}^{N_e} f(X_m) \quad (4)$$

(5) 记录种群最小目标函数值  $f_{best}$  及相应个体  $(x_1, x_2, \dots, x_D)$ 。若在该采蜜蜂周围搜索次数  $Bas$  达到阈值  $Limit$  而仍未找到更优位置, 则重新初始化该采蜜蜂。

(6) 若满足停止准则, 则输出  $f_{best}$  及相应个体  $(x_1, x_2, \dots, x_D)$ , 否则转向(2)。

## 2 量子衍生蜂群算法

本文研究一种新的量子衍生搜索方式, 并与蜂群算法融合, 称为量子衍生蜂群算法(quantum-inspired bee algorithm), 简记为QIBC。

### 2.1 量子比特的 Bloch 球面描述

在量子计算中, 一个量子比特是一个可以在二维复希尔伯特空间中描述的两能级量子体系, 根据叠加原理, 量子比特的任何态都可以写成

$$|\varphi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\phi}\sin\frac{\theta}{2}|1\rangle \quad (5)$$

其中:  $0 \leq \theta \leq \pi$ ,  $0 \leq \phi \leq 2\pi$ 。

由于  $\theta$  和  $\phi$  连续, 所以一个量子比特可以描述无穷多个不同的状态。量子比特可以用三维 Bloch 球面上的一个点来描述, 如图 1 所示, 其中  $x = \cos\phi\sin\theta$ ,  $y = \sin\phi\sin\theta$ ,  $z = \cos\theta$ 。这样, 量子比特  $|\varphi\rangle$  也可以用二维复希尔伯特空间中的向量表示为

$$|\varphi\rangle = \left[ \sqrt{\frac{1+z}{2}}, \frac{x+iy}{\sqrt{2(1+z)}} \right]^T \quad (6)$$

此时, 在 Bloch 球面上的任意一点  $P(x, y, z)$  都是一个量子比特  $|\varphi\rangle$  对应。

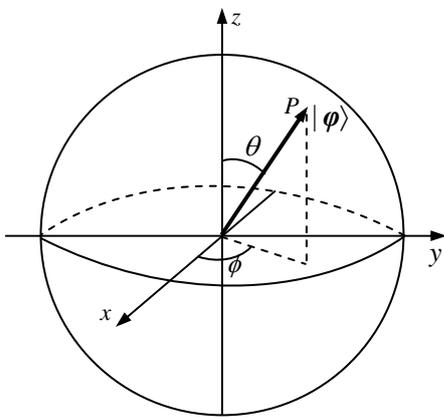


图 1 量子比特的 Bloch 球面描述

### 2.2 量子比特的绕轴旋转

在本文中, 将在 Bloch 球面上建立搜索机制, 即使量子比特在 Bloch 球面上绕着某一固定轴向着目标比特旋转。这种旋转可同时改变量子比特的两个参数  $\theta$  和  $\phi$ , 并可自动实现两个调整量的最佳匹配, 从而可更好的模拟量子行为, 并可明显提高优化能力。实现上述旋转的关键在于旋转轴的设计, 本文给出的旋转轴设计方法可表述为如下定理。

定理 1 记 Bloch 球面上点 P 和 Q 分别记为  $P=[p_x, p_y, p_z]$  和  $Q=[q_x, q_y, q_z]$ , 则在 Bloch 球面上, 量子比特以最短路径由点 P 转向点 Q 的旋转轴为 P 和 Q 的向量积, 即  $R_{axis} = P \times Q$ 。如图 2 所示<sup>[13-14]</sup>。

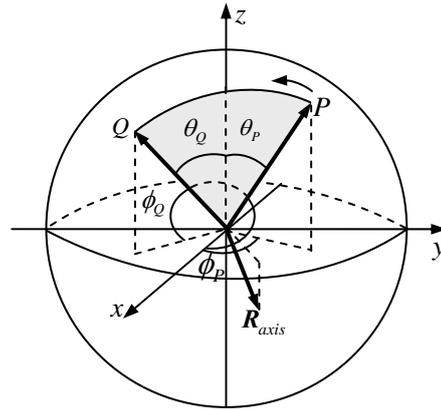


图 2 量子比特的 Bloch 球面旋转轴

令  $|\varphi_{ij}(t)\rangle$  和  $|\varphi_{bj}(t)\rangle$  的 Bloch 坐标分别为  $P_{ij}=[p_{ijx}, p_{ijy}, p_{ijz}]$  和  $P_{bj}=[p_{bjx}, p_{bjy}, p_{bjz}]$ , 根据上述定理,  $|\varphi_{ij}(t)\rangle$  向着  $|\varphi_{bj}(t)\rangle$  旋转的旋转轴为

$$R_{axis}(i, j) = \frac{P_{ij} \times P_{bj}}{\|P_{ij} \times P_{bj}\|} \quad (7)$$

根据量子计算原理, 使量子比特在 Bloch 球面上绕一个沿单位矢量  $n=[n_x, n_y, n_z]$  的轴转动  $\delta$  弧度的旋转矩阵为<sup>[15]</sup>

$$R_n(\delta) = \cos\frac{\delta}{2}I - i\sin\frac{\delta}{2}(n \times \sigma) \quad (8)$$

其中:  $I$  是单位矩阵;  $\sigma=[\sigma_x, \sigma_y, \sigma_z]$  为按下式定义的泡利矩阵。

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (9)$$

因此, 在 Bloch 球面上, 使当前比特  $|\varphi_{ij}(t)\rangle$  绕轴  $R_{axis}(i, j)$  转向  $|\varphi_{bj}(t)\rangle$  的旋转矩阵为

$$M_{ij} = \cos\frac{\delta_{ij}(t)}{2}I - i\sin\frac{\delta_{ij}(t)}{2}(R_{axis}(i, j) \times \sigma) \quad (10)$$

旋转操作为

$$|\varphi_{ij}(t)\rangle = M_{ij} |\varphi_{ij}(t)\rangle \quad (11)$$

其中  $t$  为迭代步数。

### 2.3 QIBC 编码方案

在 QIBC 中, 个体采用基于 Bloch 球面描述的量子比特编码。设种群规模为  $N_s$ , 优化空间为  $D$  维, 记第  $t$  代种群为  $P=[p_1(t), p_2(t), \dots, p_{N_s}(t)]$ , 则种群初始化时第  $i$  个个体  $p_i(0)$  可按如下式编码。

$$p_i(0) = [|\varphi_{i1}(0)\rangle, |\varphi_{i2}(0)\rangle, \dots, |\varphi_{iD}(0)\rangle] \quad (12)$$

其中  $i=1, 2, \dots, N_s$ 。

## 2.4 量子比特的投影测量

根据量子计算原理, 通过在量子比特  $|\varphi\rangle$  上施加泡利矩阵, 可以获得  $|\varphi\rangle$  的 Bloch 坐标  $(x, y, z)$ , 这一过程称为量子比特的投影测量。量子比特  $|\varphi\rangle$  的投影测量计算式为

$$x = \langle\varphi|\sigma_x|\varphi\rangle = \langle\varphi|\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}|\varphi\rangle \quad (13)$$

$$y = \langle\varphi|\sigma_y|\varphi\rangle = \langle\varphi|\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}|\varphi\rangle \quad (14)$$

$$z = \langle\varphi|\sigma_z|\varphi\rangle = \langle\varphi|\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}|\varphi\rangle \quad (15)$$

其中  $i=1, 2, \dots, N_s$ ,  $j=1, 2, \dots, D$ 。

在 QIBC 中, 将量子比特的 3 个坐标看作 3 个并列的基因, 每个个体包含 3 条并列的基因链, 每条基因链代表一个优化解。因此, 每个个体同时代表 3 个优化解。

## 2.5 解空间变换

在 QIBC 中, 每个个体给出的 3 个优化解均由 Bloch 坐标  $(x_{ij}, y_{ij}, z_{ij})$  描述。由于 Bloch 坐标取值于  $[-1, 1]$ , 因此必须通过某种变换将其映射为待优化问题的实际解。记第  $j$  维变量的取值区间为  $[\text{Min}_j, \text{Max}_j]$ , 则解空间变换可描述为如下公式。

$$X_{ij} = [\text{Max}_j(1 - x_{ij}) + \text{Min}_j(1 + x_{ij})]/2 \quad (16)$$

$$Y_{ij} = [\text{Max}_j(1 - y_{ij}) + \text{Min}_j(1 + y_{ij})]/2 \quad (17)$$

$$Z_{ij} = [\text{Max}_j(1 - z_{ij}) + \text{Min}_j(1 + z_{ij})]/2 \quad (18)$$

其中  $i=1, 2, \dots, N_s$ ,  $j=1, 2, \dots, D$ 。

## 2.6 采蜜蜂蜜源搜索

以极小值优化为例, 将初始化后的种群按目标函数值从低到高排序, 取前  $N_e$  个个体组成采蜜蜂群。余下的  $N_u = N_s - N_e$  个个体组成跟踪蜂群。对于第  $i$  只采蜜蜂  $p_i(t)$ , 首先从采蜜蜂群中随机选择蜜蜂  $p_j(t)$  和  $p_k(t)$ , 且使  $i \neq j \neq k$ , 从维度

$1, 2, \dots, D$  中随机选择一维  $d$ ; 然后, 以量子比特  $|\varphi_{id}(t)\rangle$  和  $|\varphi_{jd}(t)\rangle$  的归一化向量积  $R_{axis}(i, j)$  为旋转轴, 以量子比特  $|\varphi_{id}(t)\rangle$  和  $|\varphi_{kd}(t)\rangle$  之间的夹角  $\delta_{ik}(t)$  为旋转角度, 按式(10)构造旋转矩阵, 按式(11)实现  $|\varphi_{id}(t)\rangle$  的绕轴旋转, 旋转之后该蜜蜂记为  $\hat{p}_i(t)$ 。采用贪婪选择算子在  $p_i(t)$  和  $\hat{p}_i(t)$  之间择优选择进入下一代种群, 如下式所述。

$$p_i(t) = \begin{cases} \hat{p}_i(t), & f(\hat{p}_i(t)) < f(p_i(t)) \\ p_i(t), & f(\hat{p}_i(t)) \geq f(p_i(t)) \end{cases} \quad (19)$$

其中

$$f(p_i(t)) = \max(f(X_i(t)), f(Y_i(t)), f(Z_i(t))) \quad (20)$$

$$f(\hat{p}_i(t)) = \max(f(\hat{X}_i(t)), f(\hat{Y}_i(t)), f(\hat{Z}_i(t))) \quad (21)$$

## 2.7 跟踪蜂蜜源搜索

与普通蜂群算法相似, 首先按下式计算采蜜蜂群中各采蜜蜂的选择概率。

$$P(p = p_i) = f(p_i) / \sum_{m=1}^{N_e} f(p_m) \quad (22)$$

对于每只跟踪蜂, 首先按轮盘赌方法选择采蜜蜂  $p_i$ , 用与采蜜蜂相同的方法在其邻域搜索新位置  $\hat{p}_i$ 。若  $\hat{p}_i$  优于  $p_i$ , 则  $p_i = \hat{p}_i$  且置该蜜源搜索次数  $Bas = 0$ ; 否则若  $Bas$  小于限定阈值  $Limit$ , 置  $Bas = Bas + 1$ , 否则重新初始化  $p_i$ , 且置  $Bas = 0$ 。

## 2.8 算法终止条件

该算法中, 所有采蜜蜂和跟踪蜂都实施一次搜索称为一次迭代。每完成一次迭代, 对采蜜蜂群进行投影测量、解空间变换、计算个体目标值, 并记录全局最小目标函数值和全局最优解。该算法的终止条件为限定迭代次数, 即不论算法是否达到预先设置的优化精度, 都将一直运行到限定次数为止。

## 2.9 算法实施方案

(1) 初始化。按式(12)随机生成  $N_s$  个个体组成初始蜂群, 按目标函数值升序排序, 将前  $N_e$  个个体作为采蜜蜂群, 其余个体作为跟踪蜂群。置每只采蜜蜂蜜源跟踪次数  $Bas = 0$ , 设置跟踪次数阈值

Limit, 限定迭代次数  $G$ , 置当前迭代次数  $t = 0$ 。

(2) 对每只采蜜蜂  $p_i$ , 按 2.6 节所述实施蜜源搜索, 按式(19)进行贪婪选择。

(3) 对每只跟踪蜂, 用轮盘赌方法按式(20)定义的概率选择蜜源, 按 2.7 节所述实施蜜源跟踪搜索。

(4) 对采蜜蜂群, 按式(13)~(15)进行投影测量, 按式(16)~(18)解空间变换, 计算各采蜜蜂目标函数值, 保存全局最优解。

(5)  $t = t + 1$ , 若  $t \geq G$ , 保存最优解, 停机; 否则转(2)。

### 3 QIBC 的收敛性

令  $p_t = [p_1^t, p_2^t, \dots, p_{N_e}^t]$  为 QIBC 第  $t$  代采蜜蜂群, 其中第  $i$  个个体  $p_i^t$  的定义为

$$p_i^t = \left[ \begin{array}{c} \cos(\theta_{i1}^t/2) \\ e^{i\varphi_{i1}^t} \sin(\theta_{i1}^t/2) \end{array} \right], \dots, \left[ \begin{array}{c} \cos(\theta_{iD}^t/2) \\ e^{i\varphi_{iD}^t} \sin(\theta_{iD}^t/2) \end{array} \right] \quad (23)$$

关于 QIBC 的收敛性, 有如下结论。

引理: QIBC 的种群序列  $\{p_t, t \geq 1\}$  是有限齐次马尔可夫链。

证明 由于  $\theta_{ij}^t$  和  $\varphi_{ij}^t$  是连续变量, 因此在种群状态空间中, 状态个数是无限的, 然而在实际计算中, 所有  $\theta_{ij}^t$  和  $\varphi_{ij}^t$  都是只有有限位数的离散值。令其位数分别为  $\nu_1$  和  $\nu_2$ , 则种群  $p_t$  所有可能的状态数为  $(\nu_1\nu_2)^{D \times N_e}$ 。因此, 整个种群的状态数是有限的。而算法中采用的几种更新策略能够保证  $p_{t+1}$  仅与前代  $p_t$  有关。因此,  $\{p_t, t \geq 1\}$  是有限齐次马尔可夫链。

定理 2: QIBC 是以概率 1 收敛的。

证明: 以极小值优化为例。

记  $bp^t = \{p^* \mid f(p^*) = \max_{1 \leq i \leq N_e} fit(p_i^t)\}$  为第  $t$  代最佳个体,  $S^* = \{bp^k \mid \min_{1 \leq k \leq (\nu_1\nu_2)^{D \times N_e}} f(bp^k) = f_{best}\}$  为全局最优解集,  $f_{best}$  为全局最小目标值。

令  $T = \{t \mid bp^t \cap S^* = \emptyset\}$ ,  $p_i^k$  表示采蜜蜂群经  $t$  次迭代后处于状态空间的第  $k$  个状态,  $k = 1, 2, \dots, (\nu_1\nu_2)^{D \times N_e}$ 。

下面计算随机过程  $\{p_t, t \geq 1\}$  的一步转移概率  $P_t(i \rightarrow j) = P(p_t^i \rightarrow p_{t+1}^j)$ 。

由于 QIBC 算法采用贪婪策略, 所以  $f(p_{t+1}^j) \geq f(p_t^i)$ 。当  $i \notin T, j \in T$  时, 转移概率  $P_t(i \rightarrow j) = 0$ , 当  $i \in T, j \notin T$  时, 转移概率  $P_t(i \rightarrow j) \geq 0$ 。

设  $P_t(i)$  为  $p_t$  处于状态  $i$  的概率, 记  $P_t = \sum_{i \in T} P_t(i)$ , 由马氏链性质可知,  $p_{t+1}$  处于状态  $j \in T$  的概率为

$$P_{t+1} = \sum_{i \in T} \sum_{j \in T} P_t(i)P_t(i \rightarrow j) + \sum_{i \notin T} \sum_{j \in T} P_t(i)P_t(i \rightarrow j)$$

由  $P_t = \sum_{i \in T} \sum_{j \in T} P_t(i)P_t(i \rightarrow j) + \sum_{i \notin T} \sum_{j \in T} P_t(i)P_t(i \rightarrow j)$  得

$$P_{t+1} = P_t - \sum_{i \in T} \sum_{j \notin T} P_t(i)P_t(i \rightarrow j) \leq P_t, \text{ 故 } \lim_{t \rightarrow \infty} P_t = 0$$

$$\lim_{t \rightarrow \infty} P(f(bp^t) = f_{best}) = 1 - \lim_{t \rightarrow \infty} \sum_{i \in T} P_t(i) = 1 - \lim_{t \rightarrow \infty} P_t = 1$$

即 QIBC 是以概率 1 收敛的。

### 4 实验结果分析

以函数极值优化为例, 并通过与文献[1]中的蜂群算法(bee colony, BC)、文献[12]中的 Bloch 量子遗传算法(Bloch quantum genetic algorithm, BQGA)、带精英保留策略的遗传算法(elite genetic algorithm, EGA)、文献[16]中的量子 Delta 势阱粒子群算法(quantum delta potential-based particle swarm optimization QDPSO)进行对比, 验证 QIBC 的优越性。所有算法均采用 Matlab R2009a 编程, 并在 1.0GB 内存的 P-II 2.0 GHz 微机上实现。

#### 4.1 测试函数

采用如下 10 个标准测试函数验证 QIBC 的性能, 所有 10 个函数全部为最小值优化,  $X^*$  为最小极值点。

$$(1) f_1(X) = \sum_{i=1}^D \left( \sum_{j=1}^i x_j \right)^2,$$

$$-100 \leq x_i \leq 100, x_i^* = 0, f(X^*) = 0$$

$$(2) f_2(X) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2),$$

$$\begin{aligned}
 & -100 \leq x_i \leq 100, \quad x_i^* = 1, \quad f(\mathbf{X}^*) = 0. \\
 (3) \quad & f_3(\mathbf{X}) = \sum_{i=1}^D i * x_i^4 (1 + \text{rand}(0,1)), \\
 & -100 \leq x_i \leq 100, \quad x_i^* = 0, \quad f(\mathbf{X}^*) = 0. \\
 (4) \quad & f_4(\mathbf{X}) = -20e^{-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}} - e^{\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)} + \\
 & 20 + e, \\
 & -100 \leq x_i \leq 100, \quad x_i^* = 0, \quad f(\mathbf{X}^*) = 0. \\
 (5) \quad & f_5(\mathbf{X}) = \frac{D(D+4)(D-1)}{6} + \sum_{i=1}^D (x_i - 1)^2 - \sum_{i=2}^D x_i x_{i-1}, \\
 & -D^2 \leq x_i \leq D^2, \quad x_i^* = i(D - i + 1), \quad f(\mathbf{X}^*) = 0. \\
 (6) \quad & f_6(\mathbf{X}) = \sum_{k=1}^D \sum_{j=1}^D \left( \frac{y_{jk}^2}{4000} - \cos(y_{jk}) + 1 \right), \\
 & y_{jk} = 100(x_k - x_j^2)^2 + (1 - x_j)^2, \\
 & -100 \leq x_i \leq 100, \quad x_i^* = 1, \quad f(\mathbf{X}^*) = 0. \\
 (7) \quad & f_7(\mathbf{X}) = g(x_1, x_2) + \dots + g(x_{D-1}, x_D) + g(x_D, x_1), \\
 & g(x, y) = (x^2 + y^2)^{0.25} [\sin^2(50(x^2 + y^2)^{0.1}) + 1], \\
 & -100 \leq x_i \leq 100, \quad x_i^* = 0, \quad f(\mathbf{X}^*) = 0. \\
 (8) \quad & f_8(\mathbf{X}) = \sum_{i=1}^D x_i^2 + \left( \sum_{i=1}^D 0.5ix_i \right)^2 + \left( \sum_{i=1}^D 0.5ix_i \right)^4, \\
 & -100 \leq x_i \leq 100, \quad x_i^* = 0, \quad f(\mathbf{X}^*) = 0. \\
 (9) \quad & f_9(\mathbf{X}) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \\
 & -500 \leq x_i \leq 500, \quad x_i^* = 0, \quad f(\mathbf{X}^*) = 0. \\
 (10) \quad & f_{10}(\mathbf{X}) = 10D + \sum_{i=1}^D (y_i^2 - 10\cos(2\pi y_i)) \\
 & y_i = \begin{cases} x_i, & |x_i| < 1/2 \\ \text{round}(2x_i)/2, & |x_i| \geq 1/2 \end{cases} \\
 & -100 \leq x_i \leq 100, \quad x_i^* = 0, \quad f(\mathbf{X}^*) = 0.
 \end{aligned}$$

## 4.2 QIBC 参数对性能的影响

QIBC 参数只有 4 个: 采蜜蜂数  $N_e$ , 跟踪蜂数  $N_u$ , 采蜜蜂邻域搜索次数阈值  $Limit$ , 限定迭代步数  $G$ 。下面以  $D = 50$  的函数  $f_6$  为例, 考察各参数变化对 QIBC 优化能力的影响。实验方案是:

$$\begin{aligned}
 (1) \quad & \begin{cases} N_e = 10, 15, \dots, 50 \\ N_u = 30 \end{cases}; \quad Limit = 50; \quad G = 10^4 \\
 (2) \quad & \begin{cases} N_e = 30 \\ N_u = 10, 15, \dots, 50 \end{cases}; \quad Limit = 50; \quad G = 10^4 \\
 (3) \quad & \begin{cases} N_e = 10 \\ N_u = 50 \end{cases}; \begin{cases} N_e = 20 \\ N_u = 40 \end{cases}; \begin{cases} N_e = 30 \\ N_u = 30 \end{cases}; \begin{cases} N_e = 40 \\ N_u = 20 \end{cases}; \\
 & \begin{cases} N_e = 50 \\ N_u = 10 \end{cases}; \quad Limit = 50; \quad G = 10^4 \\
 (4) \quad & \begin{cases} N_e = 20 \\ N_u = 40 \end{cases}; \quad Limit = 10, 20, \dots, 200; \quad G = 10^4 \\
 (5) \quad & \begin{cases} N_e = 20 \\ N_u = 40 \end{cases}; \quad Limit = 50; \\
 & G = 10^3, 5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 2 \times 10^5
 \end{aligned}$$

对于以上每种设置, 均用 QIBC 优化 50 次, 平均优化结果如图 3~6 所示。

从图 3~4 可以看出, 当  $N_e = N_u$  时 QIBC 的优化效果较为理想, 对于 50 维的  $f_6$  而言, 可取  $N_e = N_u = 30$ 。从图 5 可知,  $Limit$  的变化对 QIBC 的优化效果影响不大, 且在  $[50, 150]$  范围内效果较好。从图 6 可以看出, 当限定步数超过 10 000 之后, 优化效果的变化非常缓慢, 如当  $G = 10\,000$  时, 50 次平均结果为 0.605 5,  $G = 200\,000$  时为 0.008 8。

综合以上实验结果,  $N_e$  和  $N_u$  越接近, QIBC 的性能越好;  $Limit$  对 QIBC 的影响不大; 当  $G$  较小时, QIBC 的优化结果随  $G$  的增大快速下降, 但当  $G$  较大时, QIBC 优化结果趋于稳定。

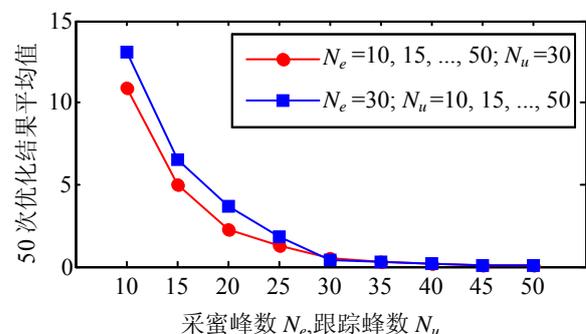


图3 QIBC在不同 $N_e(N_u)$ 下的优化结果对比

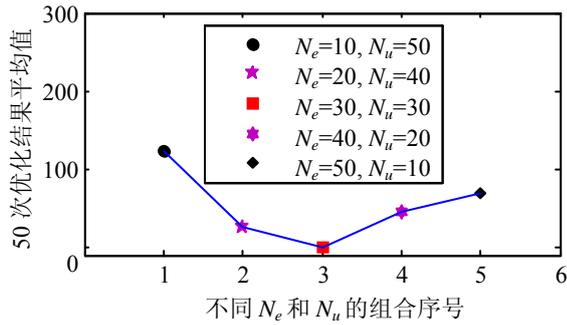


图4 QIBC在不同 $N_e, N_u$ 组合下的优化结果对比

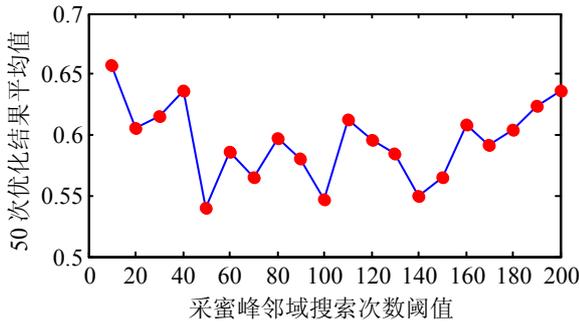


图5 QIBC在不同邻域搜索次数阈值下的优化结果对比

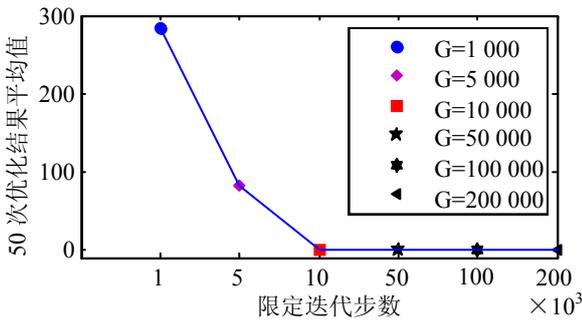


图6 QIBC在不同限定迭代步数下的优化结果对比

### 4.3 仿真参数设置

为便于比较,根据目标函数的复杂程度为每个目标函数分别设置了精度阈值,只有当优化结果小于该阈值时,才认为算法收敛。 $f_1, f_6$  的阈值为 1.0,  $f_2$  的阈值为 10,  $f_3$  的阈值为  $10^{-5}$ ,  $f_4, f_9, f_{10}$  的阈值为  $10^{-10}$ ,  $f_5$  的阈值为  $10^2$ ,  $f_7$  的阈值为  $10^{-8}$ ,  $f_8$  的阈值为  $10^4$ 。若算法收敛,则称收敛时的优化步数为迭代次数,否则迭代次数等于限定次数。

仿真中 10 个函数维度均取  $D = 30$ 。5 种算法种群规模均取 40。对于 QIBC 和 BC, 根据上节实验结果,  $N_e = 20, N_u = 20, Limit = 50$ 。对于

BQGA, 根据文献[12], 转角初值取  $0.05\pi$ , 变异概率取  $10^{-3}$ 。对于 QDPSO, 根据文献[16], 控制参数  $\lambda = 1.2$ 。对于 EGA, 交叉概率取 0.8, 变异概率取 0.01。5 种算法的迭代次数均取  $G = 10^4$ 。

### 4.4 仿真结果对比分析

为增强对比结果的客观性,每种算法均独立运行 50 次, 然后给出统计结果对比。每次迭代的平均时间对比如表 1 所示。为使对比充分, 每种算法除给出平均结果外, 还给出了最差结果和最好结果。收敛次数, 平均步数, 优化结果对比如表 2 所示。

从表 1 可以看出, 就运行时间而言, 5 种算法从长到短的排序为 QIBC, BQGA, BC, EGA, QDPSO。这是因为, 在 QIBC 和 BQGA 中都使用了三链量子比特编码机制, 在每步迭代过程中, 每个个体需要针对 3 条基因链分别进行更新、解空间变换、计算目标函数值等操作, 从而延长了运行时间的缘故。而 QIBC 长于 BQGA 是因在 QIBC 中增加了投影测量和设计旋转轴及旋转矩阵这些新的计算量的缘故。

由表 2 可知, 就优化能力而言, QIBC 明显优于 BC; BQGA 明显优于 EGA; 而 QDPSO 劣于 QIBC, 优于 BQGA, 与 BC 优化能力相当; 同时 QIBC 的平均、最差、最好结果也是差别最小的。5 种算法的优化性能由高到低排序大致为 QIBC, BC, QDPSO, BQGA, EGA。

表 1 每步迭代的平均时间对比 /s

函数	QIBC	BC	BQGA	QDPSO	EGA
$f_1$	0.028 1	0.003 8	0.021 6	0.001 6	0.002 6
$f_2$	0.021 9	0.001 6	0.009 5	0.000 5	0.001 5
$f_3$	0.022 3	0.001 8	0.010 0	0.000 6	0.001 5
$f_4$	0.021 8	0.001 6	0.010 3	0.000 5	0.001 4
$f_5$	0.022 1	0.001 7	0.009 8	0.000 6	0.001 4
$f_6$	0.059 6	0.010 2	0.042 9	0.004 0	0.012 6
$f_7$	0.024 5	0.002 3	0.027 0	0.001 0	0.001 1
$f_8$	0.022 2	0.001 8	0.010 4	0.000 6	0.001 5
$f_9$	0.017 6	0.004 6	0.011 6	0.000 5	0.000 3
$f_{10}$	0.018 3	0.005 0	0.017 8	0.000 7	0.000 5

表 2 5 种算法优化结果对比

函数	算法	收敛次数	平均步数	平均结果	最差结果	最好结果
$f_1$	QIBC	50	3 083	0.157 7	0.573 7	0.0313
	BC	0	10 000	349.70	1 075.1	29.904
	BQGA	0	10 000	131.59	246.14	36.583
	QDPSO	21	7 190	17.893	414.60	2.22e-6
	EGA	0	10 000	456.36	887.14	227.14
$f_2$	QIBC	37	4 643	9.422 1	80.358	0.044 4
	BC	23	6 250	52.786	914.20	0.069 9
	BQGA	0	10 000	1 046	6 992	138.01
	QDPSO	32	5 076	12.440	72.776	0.002 4
	EGA	0	10 000	2 654.1	12 494	301.56
$f_3$	QIBC	48	5 050	1.18e-6	2.10e-5	4.74e-10
	BC	45	8 016	1.36e-6	3.32e-5	9.61e-10
	BQGA	0	10 000	187.72	792.30	21.370
	QDPSO	47	6 471	2.17e-4	0.010 7	5.2e-75
	EGA	0	10 000	1 504.8	8 057.3	219.92
$f_4$	QIBC	47	5 685	3.53e-11	6.29e-10	4.17e-14
	BC	35	7 210	7.84e-10	1.85e-08	5.59e-14
	BQGA	0	10 000	15.894	20.742	10.280
	QDPSO	6	9 730	16.377	20.005	3.81e-14
	EGA	0	10 000	19.287	21.047	3.9204
$f_5$	QIBC	41	3 561	68.415	398.62	2.470 9
	BC	10	9 028	245.01	663.55	488.40
	BQGA	0	10 000	5 427	12 981	435.03
	QDPSO	6	9 061	827.33	2 992.8	3.290 0
	EGA	0	10 000	6 581.4	9 429.7	4 487.2
$f_6$	QIBC	49	2 467	0.1926	8.905 4	1.25e-09
	BC	28	5 838	17.966	106.97	3.47e-08
	BQGA	0	10 000	491.21	829.02	196.59
	QDPSO	0	10 000	171.47	351.05	8.8842
	EGA	0	10 000	22 127	355 958	978.91
$f_7$	QIBC	49	5 763	3.39e-09	1.69e-07	0
	BC	36	8 354	3.67e-08	1.45e-6	1.49e-10
	BQGA	0	10 000	110.73	169.67	32.058
	QDPSO	32	8 640	0.4312	9.2145	2.42e-19
	EGA	0	10 000	164.55	231.46	93.272
$f_8$	QIBC	50	5 454	1 412	5 700	61.335
	BC	0	10 000	50 112	69 884	33 193
	BQGA	0	10 000	4 910.5	12 882	1 006.1
	QDPSO	50	7 392	2 003	5 300	200.81
	EGA	0	10 000	7 843.8	38 137	4 291.6
$f_9$	QIBC	50	2 162	8.65e-17	3.33e-15	0
	BC	36	3 424	0.003 3	0.022 1	0
	BQGA	12	7 780	0.023 9	0.127 5	0
	QDPSO	29	5 040	0.006 2	0.513 6	0
	EGA	0	10 000	3.122 4	7.336 6	1.610 1
$f_{10}$	QIBC	50	2 606	1.13e-14	1.13e-13	0
	BC	49	6 290	4.03e-12	1.89e-10	0
	BQGA	0	10 000	172.42	416.64	26.023
	QDPSO	6	9 284	7.355 3	15.882	0
	EGA	0	10 000	822.60	1 298.1	455.30

这种对比结果可作如下分析。(1) 量子比特三链编码机制的引入有效提高了算法对解空间的遍历性。根据 Bloch 球面的几何特性, 这种编码机制能够扩充全局最优解的数量, 从而增加获得全局最优解的概率, 这是 2 种量子衍生算法优于对应的经典算法的根本原因。(2) QIBC 的优化能力高于 BQGA 的原因在于, 2 种算法中量子比特的调整采用了不同的方法。BQGA 采用直接法, 直接调整量子比特的 2 个参数  $\theta$  和  $\phi$ , 且采用了相同的调整量 (即  $\Delta\theta = \Delta\phi$ ), 这种方法显然不能沿最短路径逼近目标比特, 因为欲沿最短路径逼近,  $\Delta\theta$  和  $\Delta\phi$  必须满足某种匹配关系, 而这种匹配关系是很难用确定的解析式明确表达的。而在 QIBC 中, 个体量子比特的调整采用了间接方法。即直接使量子比特沿 Bloch 球面上的旋转轴在通往目标比特的大圆上旋转 (此时具有最短路径)。尽管该方法没有直接调整量子比特的 2 个参数, 然而却自动实现了 2 个调整量  $\Delta\theta$  和  $\Delta\phi$  的精确匹配。使这种方法具有更高的优化效率。

值得指出, 本文算法虽然具有较强的优化能力, 但计算复杂度明显高于传统算法, 这是本文算法的缺点。这是由于与传统算法相比, QIBC 增加了计算旋转轴、旋转角度、旋转矩阵、解空间变换等新的操作。然而综合运行时间和优化能力两方面, QIBC 正是以牺牲运行时间来换取优化能力提高的, 这与无免费午餐定理是一致的。另外尽管 QIBC 运行时间较长, 但对于大量不关注运行时间的离线优化问题, 仍然具有广阔的应用前景。

下面考察维数及参数变化对算法性能的影响。为简便仅对比 QIBC 和 BC, 并且仅以函数  $f_9$  和  $f_{10}$  为例, 维数  $D=100$ , 根据 4.2 节的结论, 种群取  $N_e = N_u = 40$ ,  $Limit = 100$ ,  $G = 10^4$ 。 $f_9$  和  $f_{10}$  的精度阈值均取  $10^{-10}$ 。2 种算法分别运行 10 次。优化结果对比如表 3 所示。

实验结果表明, 对于高维优化问题, 本文算法同样表现出优于传统算法的优良性能。

表 3 QIBC 和 BC 的优化结果对比

函数	算法	收敛次数	平均步数	平均结果	最差结果	最好结果
$f_9$	QIBC	10	6267	1.11e-16	6.66e-16	0
	BC	0	10000	1.23e-07	6.80e-07	1.5e-08
$f_{10}$	QIBC	7	9065	4.14e-08	4.02e-07	3.4e-13
	BC	2	9758	3.17e-07	1.90e-06	2.4e-11

## 5 工程应用

从数学角度看,任何涉及连续指标优化的工程实际问题(用料最省、时间最短、利润最大等),均可等效为一个多元函数求极值问题,因此都可以用本文算法求解。本节针对油藏测井解释中的复杂水淹层识别问题,研究基于 QIBC 优化的非线性回归模型识别方法。应用结果表明,该方法有较高的预测精度。

### 5.1 建立回归模型

在油藏水淹层测井解释中,表征油层水淹状况的测井特征参数较多。为建立回归模型,首先需要确定输入输出指标集。经过对现场取心井岩心分析资料采用统计方法进行筛选,并结合专家经验,最终选用油层厚度、深侧向电阻率、自然电位、声波时差和微电位等 5 个参数作为输入指标,而水淹等级为输出指标。利用大庆萨北油田高台子油层 5 口取心井资料,进行该区块水淹层自动识别仿真,共选取 100 个小层,部分样本数据如表 4 示。表中,水淹等级 1, 2, 3, 4 分别代表未水淹、弱水淹、中水淹、强水淹。

表 4 水淹层识别部分样本数据

厚度	深侧向电阻率	自然电位	声波时差	微电位	水淹等级
0.60	20.0	5.0	150.0	2.50	2
0.90	11.0	0.5	280.0	2.00	4
0.20	17.0	6.5	148.0	1.40	3
...	...	...	...	...	...
0.50	13.0	3.0	168.0	1.90	1

考虑到水淹层识别问题的复杂性,我们建立的回归模型包括 2 部分:线性部分和非线性部分。其

中非线性部分又称为残差项,其作用是补偿线性部分存在的误差,由于水淹级别分为 4 个等级,所以共有 4 个残差项,均采用高斯函数形式,具体如下式所示。

$$Y = CX + \sum_{d=1}^4 \frac{1}{\sqrt{2\pi}\sigma_d} \exp\left(-\frac{(X - U_d)^T(X - U_d)}{2\sigma_d^2}\right) \quad (24)$$

其中  $X = [X_1, X_2, X_3, X_4, X_5]^T$  为输入,  $C = [C_1, C_2, C_3, C_4, C_5]$ ,  $U_d = [U_{d1}, U_{d2}, U_{d3}, U_{d4}, U_{d5}]^T$  ( $d=1, 2, 3, 4$ ) 和  $\sigma = [\sigma_1, \sigma_2, \sigma_3, \sigma_4]$  为模型参数。

从式(24)可知,回归模型共包括 29 个参数,只有这些参数确定之后,该模型才能真正用于实际预测。下面研究基于量子蚁群优化的参数确定方法。

令  $D$  为回归模型的期望输出,  $N$  为样本数目。目标函数定义为期望输出与实际输出误差的平方。

$$J(C, U, \sigma) = \sum_{i=1}^N (D_i - Y_i)^2 = \sum_{i=1}^N \left( D_i - CX_i - \sum_{d=1}^4 \frac{\exp\left(-\frac{(X_i - U_d)^T(X_i - U_d)}{2\sigma_d^2}\right)}{\sqrt{2\pi}\sigma_d} \right)^2 \quad (25)$$

上式实际上是一个包含 29 个待优化变量的复杂非线性函数,其取全局极小值时的自变量组合即为我们最终所需要的优化结果。因此,应用上式即可将回归模型的建立问题归结为一个多元函数求极值问题,而解决这类问题,恰好是 QIBC 算法的长处。

### 5.2 回归结果及分析

为体现提出方法的优越性,实验中我们将此模型与普通 BP 网络(Common BP, CBP)进行对比。

QIBC 的参数设置:种群规模分别设置为  $\begin{cases} N_e = 20 \\ N_u = 20 \end{cases}$ ,  $\begin{cases} N_e = 20 \\ N_u = 30 \end{cases}$ ,  $\begin{cases} N_e = 30 \\ N_u = 30 \end{cases}$ ,  $D=29$ ;  $Limit=50$ 。

CBP 参数设置:输入层 5 个节点,隐层分别取 5, 10, 15 个节点;输出层 1 个节点;学习速率取 0.8,所有算法限定代数均取 1000,限定均方差取 0.1。采用前 80 个样本用于模型训练,后 20

个样本用于测试模型的泛化能力。QIBC 和 CBP 的训练结果对比如表 5 所示。

表 5 QIBC 和 CBP 训练结果对比题

QIBC			CBP		
种群规模	迭代步数	均方误差	隐层节点	迭代步数	均方误差
(20, 20)	799	0.0898	5	1000	0.1383
(20, 30)	683	0.0835	10	887	0.0589
(30, 30)	488	0.0761	15	1000	0.1293

由表 5 可知, 对于种群规模的 3 种取值, QIBC 优化模型的逼近能力均优于 CBP, QIBC 在种群规模较小时同样表现出了较强的逼近能力。这是因为 QIBC 采用的量子计算机制有效增强了寻优能力, 从而找到了高质量的模型参数组合, 进而提高了模型非线性映射能力的缘故。而对于 CBP, 当隐层节点数为 5 时, 映射能力太弱, 网络不能收敛, 而当隐层节点数为 15 时, 由于误差曲面过于复杂且易于陷入局部极小, 从而使网络也不能收敛。

将预测集中 20 个小层数据提交训练后的网络, QIBC 识别正确 19 个, CBP 识别正确 14 个。QIBC 识别正确率比 CBP 高出 25 个百分点。以上结果表明, QIBC 的预测精度较为理想, 在工程优化方面有一定潜力。

## 6 结论

提出了一种量子衍生蜂群算法, 采用基于 Bloch 球面描述的量子比特实现个体编码, 采用量子比特在 Bloch 球面上的绕轴旋转实现种群进化。实验结果揭示出基于 Bloch 球面描述的量子比特的编码方法, 和使量子比特两参数的调整量具有最佳匹配的绕轴旋转的个体更新方法, 切实能够提高智能优化算法的优化能力。

### 参考文献:

[1] Karaboga D. An Idea Based On Honey Bee Swarm For Numerical Optimization [R]. Technical Report- TR06, Engineering Faculty, Computer Engineering Department, Erciyes University, Kayseri, Turkey, 2005.  
[2] Bahriye A, Dervis K. A modified Artificial Bee Colony algorithm for real-parameter optimization [J]. Information Sciences(S0020-0255), 2012, 192(6): 120-142.

[3] Xiang W L, An M Q. An efficient and robust artificial bee colony algorithm for numerical optimization [J]. Computers & Operations Research(S0305-0548), 2013, 40(5): 1256-1265.  
[4] Li G Q, Niu P F, Xiao X J. Development and investigation of efficient artificial bee colony algorithm for numerical function optimization [J]. Applied Soft Computing(S1568-4946), 2012, 12(1): 320-332.  
[5] Karaboga D, Akay B, Ozturk C. Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks [C]. Proceedings of the 4th International Conference on Computer Science, Kitakyushu, Japan, August 16-18, 2007: 318-329.  
[6] Karaboga N. A new design method based on artificial bee colony algorithm for digital IIR filter [J]. Journal of the Franklin Institute(S0016-0032), 2009, 346(4): 328-348.  
[7] Rao R, Narasimham S, Ramalingaraju M. Optimization of distribution network configuration for loss reduction using artificial bee colony algorithm [J]. International Journal of Electrical Power and Energy Systems Engineering(S0142-0615), 2008, 1(2): 709-715.  
[8] Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem [J]. Applied Soft Computing(S1568-4946), 2009, 9(2): 625-631.  
[9] 丁海军, 李峰磊. 蜂群算法在 TSP 问题上的应用及参数改进[J]. 中国科技信息, 2008, 25(3): 241-243.  
[10] 康飞, 李俊杰, 许青. 改进人工蜂群算法及其在反演分析中的应用[J]. 水电能源科学, 2009, 27(1): 126-129.  
[11] Duan H B, Xu C F, Xing Z H. A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems [J]. International Journal of Neural Systems(S0129-0657), 2010, 20(1): 39-50.  
[12] 李盼池. 基于量子位 Bloch 坐标的量子遗传算法及其应用[J]. 控制理论与应用, 2008, 25(6): 985-989.  
[13] 李盼池, 林晶晶. 基于 Bloch 球面搜索的混沌量子免疫算法[J]. 系统工程与电子技术, 2012, 34(12): 2592-2598.  
[14] 李盼池, 王琪超, 施光尧. 基于 Bloch 球面搜索的量子粒子群优化算法[J]. 计算物理, 2013, 30(3): 454-462.  
[15] Karaboga D, Akay B, Ozturk C. Artificial bee colony (ABC) optimization algorithm for training feed-forward neural networks[C]//Proceedings of the 4th International Conference on Modeling Decisions for Artificial Intelligence, Springer, August 16-18, 2007: 318-329.  
[16] 李盼池, 王海英, 宋考平, 等. 量子势阱粒子群优化算法的改进研究[J]. 物理学报, 2012, 61(6): 060302.