

量子衍生布谷鸟搜索算法^①

李盼池, 杨淑云, 刘显德, 潘俊辉, 肖红, 曹茂俊

(东北石油大学 计算机与信息技术学院, 大庆 163318)

摘要: 为提高布谷鸟搜索算法的寻优能力, 通过在经典布谷鸟搜索算法中引入量子计算机制, 提出了一种量子衍生布谷鸟搜索算法. 该算法采用量子比特编码个体, 采用泡利矩阵确定旋转轴, 采用 Levy 飞行原理确定旋转角度, 采用量子比特在 Bloch 球面上的绕轴旋转实现个体更新. 标准函数极值优化的实验结果表明, 与传统布谷鸟搜索算法相比, 该算法的搜索能力确有明显提升.

关键词: 仿生智能优化; 群智能优化; 布谷鸟算法; 量子衍生优化; 算法设计

引用格式: 李盼池, 杨淑云, 刘显德, 潘俊辉, 肖红, 曹茂俊. 量子衍生布谷鸟搜索算法. 计算机系统应用, 2017, 26(9): 122-127. <http://www.c-s-a.org.cn/1003-3254/5915.html>

Quantum-Inspired Cuckoo Search Algorithm

LI Pan-Chi, YANG Shu-Yun, LIU Xian-De, PAN Jun-Hui, XIAO Hong, CAO Mao-Jun

(School of Computer & Information Technology, Northeast Petroleum University, Daqing 163318, China)

Abstract: In order to improve the search ability of the cuckoo search algorithm, this paper proposes a quantum-inspired cuckoo search algorithm by introducing the quantum computing mechanism into the classical cuckoo search algorithm. In the proposed algorithm, the qubits are used to encode individuals, and the Pauli matrixes are employed to determine rotation axis. The Levy flight principle is applied to obtain rotation angle, and the rotation of the qubits on the Bloch sphere is used to update the individuals. The experimental results of extreme optimization of benchmark test functions show that the proposed algorithm is obviously superior to the classical cuckoo search algorithm in optimization ability.

Key words: bionic intelligent optimization; swarm intelligence optimization; cuckoo algorithm; quantum-inspired optimization; algorithm design

2009年, 英国剑桥大学学者提出了一种新的启发式智能优化算法: 布谷鸟搜索算法(cuckoo search algorithm, CS)^[1]. 该算法基于布谷鸟寻找鸟巢放置鸟蛋的行为和鸟类飞行的 Levy 行为建立搜索机制. 近几年来, 国内学者已对布谷鸟算法展开深入研究. 2011年, 西安工程大学王凡提出了基于高斯扰动的布谷鸟算法^[2], 有效提高了算法的收敛速度. 2012年, 中国矿业大学的李明提出了布谷鸟和差分进化相融合的优化算法^[3], 有效提高了原算法的优化能力. 随后在理论算法方面, 先后提出了基于细菌觅食行为中的趋化搜索策略的布谷

鸟全局优化算法^[4], 具有动态惯性策略的布谷鸟搜索算法^[5], 多目标布谷鸟搜索算法^[6,7], 基于多种群粒子群和布谷鸟搜索的联合寻优算法^[8], 基于梯度的自适应快速布谷鸟搜索算法^[9], 基于种群特征反馈的布谷鸟搜索算法^[10]. 在算法应用方面, 布谷鸟搜索算法已成功应用在飞行器容错控制^[11], 神经网络训练^[12], 水库优化调度^[13], 干扰资源分配^[14]等众多应用领域.

尽管目前国内外文文献已提出多种有关布谷鸟搜索算法性能的改进策略, 然而有关量子计算与布谷鸟搜索的融合研究尚不多见. 本文提出一种新颖的量子衍

① 基金项目: 黑龙江省教育厅科学技术研究项目(12541059)

收稿时间: 2016-12-19; 采用时间: 2017-01-05

生布谷鸟搜索算法,其编码机制为采用带两个相位参数量子比特编码个体,寻优机制为采用量子比特在 Bloch 球面上的绕轴旋转模拟布谷鸟的搜索行为.其中旋转角度采用 Levy 飞行策略决定.采用 CEC2013 的 28 个标准测试函数做为优化对象,仿真结果验证了提出算法的优越性.

1 布谷鸟搜索算法

标准布谷鸟搜索算法是通过模拟布谷鸟借窝产蛋的繁殖习性有效求解最优化问题的全局搜索算法.该算法中,一个鸟巢代表一个候选解,首先基于当前解采用 Levy 飞行得到新解,然后根据鸟蛋被发现的概率丢弃一些现有的解,并重新生成相同数目的新解.最后进行种群评估,更新全局最优解,完成一次迭代.

布谷鸟搜索算法采用 Levy 飞行理论产生新解,具体方法为:

$$X_i^{t+1} = X_i^t + a \oplus L(\lambda) \quad (1)$$

其中 X_i^t 表示第 t 代第 i 个解, \oplus 是点对点乘法, a 是控制步长,通常按下式计算:

$$a = a_0(X_i^b - X_i^t) \quad (2)$$

其中 a_0 是常数(通常取 0.01), X_i^b 是当前最优解.

$L(\lambda)$ 为 Levy 概率分布随机数,通常按下式计算:

$$Levy(\lambda) = \frac{\phi \times \mu}{|\nu|^{1/\lambda}} \quad (3)$$

其中 μ, ν 服从标准正态分布, $\lambda = 1.5$.

$$\phi = \left(\frac{\Gamma(1+\lambda) \times \sin(\pi\lambda/2)}{\Gamma(\frac{1+\lambda}{2}) \times \lambda \times 2^{(\lambda-1)/2}} \right)^{1/\lambda} \quad (4)$$

综合式(1)-式(4),在 Levy 随机搜索中,布谷鸟算法采用下式生成新解:

$$X_i^{t+1} = X_i^t + a_0 \frac{\phi \times \mu}{|\nu|^{1/\lambda}} (X_i^b - X_i^t) \quad (5)$$

按发现概率丢弃部分解之后,采用随机游走方式重新生成与丢弃数目相同的新解:

$$X_i^{t+1} = X_i^t + r(X_j^t - X_k^t) \quad (6)$$

其中 $r \in (0, 1)$ 是缩放因子, X_j^t 和 X_k^t 是第 t 代随机选择的两个解.

2 量子衍生布谷鸟搜索算法

本节提出一种量子衍生布谷鸟搜索算法(Quantum

Inspired Cuckoo Search Algorithm, QICS),包括个体的编码机制和搜索机制两部分,下面分别予以介绍.

2.1 QICS 的编码机制

(1) 基于量子比特的鸟巢编码

在 QICS 中,鸟巢采用基于 Bloch 球面描述的量子比特编码.设鸟巢数量为 N ,空间维数为 D ,记第 t 代种群为 $P(t) = [p_1(t), p_2(t), \dots, p_N(t)]^T$,则第 i 个鸟巢 $p_i(0)$ 的初始编码可写为:

$$\begin{cases} p_i(0) = [|\varphi_{i1}(0)\rangle, |\varphi_{i2}(0)\rangle, \dots, |\varphi_{iD}(0)\rangle] \\ |\varphi_{ij}(0)\rangle = \cos(\theta_{ij}/2)|0\rangle + e^{i\phi_{ij}} \sin(\theta_{ij}/2)|1\rangle \\ \theta_{ij} = rand(0, 1)\pi, \quad \phi_{ij} = rand(0, 1)2\pi \end{cases} \quad (7)$$

(2) 量子鸟巢到经典鸟巢的变换

对于量子比特编码的鸟巢,只有转换为具体的数值向量才能评价解的优劣,这可以通过对量子鸟巢的投影测量及解空间变换获得.具体方法为首先采用泡利矩阵对量子鸟巢实施投影测量得到 Bloch 球面上的坐标鸟巢,然后通过解空间变换,将这些坐标鸟巢映射为具体的鸟巢位置.泡利矩阵的定义式为:

$$\sigma_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \sigma_y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \sigma_z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (8)$$

对于第 i 个鸟巢上的第 j 个量子比特 $|\varphi_{ij}\rangle$,其投影测量计算式为:

$$\begin{cases} x_{ij} = \langle \varphi_{ij} | \sigma_x | \varphi_{ij} \rangle \\ y_{ij} = \langle \varphi_{ij} | \sigma_y | \varphi_{ij} \rangle \\ z_{ij} = \langle \varphi_{ij} | \sigma_z | \varphi_{ij} \rangle \end{cases} \quad (9)$$

其中 $i = 1, 2, \dots, N$, $j = 1, 2, \dots, D$.

此时, $X_i = [x_{ij}]$, $Y_i = [y_{ij}]$, $Z_i = [z_{ij}]$ 即为 Bloch 球面上的坐标鸟巢.

(3) 坐标鸟巢的解空间变换

在 QICS 中,每个量子鸟巢都对对应到三个坐标鸟巢,然而这三个坐标鸟巢并不是彼此独立的.这是因为三者之间存在 $x_{ij}^2 + y_{ij}^2 + z_{ij}^2 = 1$ 的约束关系.因此若一组坐标(例如 x_{ij})较优,则另外两组(y_{ij}, z_{ij})一般会较差.因此在 QICS 中,对于第 i 个量子鸟巢测量之后得到的三个坐标鸟巢 $X_i = [x_{ij}]$, $Y_i = [y_{ij}]$, $Z_i = [z_{ij}]$,我们只采用 $X_i = [x_{ij}]$.

由于 $x_{ij} \in [-1, 1]$,因此需要进行解空间变换.记第 j 维优化空间的取值区间为 $[\text{Min}_j, \text{Max}_j]$,则解空间变换如下式所示:

$$X_{ij} = [\text{Min}_j(1 - x_{ij}) + \text{Max}_j(1 + x_{ij})]/2 \quad (10)$$

其中 $i = 1, 2, \dots, N, j = 1, 2, \dots, D$.

此时 $[X_{i1}, X_{i2}, \dots, X_{iD}]$ 即为第 i 只布谷鸟的实际鸟巢.

2.2 QICS 的搜索机制

在 QICS 中, 搜索机制采用量子比特在 Bloch 球面上的绕轴旋转实现, 其中旋转角度采用 Levy 飞行理论决定, 旋转轴采用泡利矩阵决定. 具体旋转方法如下.

(1) 基于 Levy 飞行理论产生新解

令 $|\varphi_{ij}(t)\rangle$ 和 $|\varphi_{bj}(t)\rangle$ 分别为第 i 个量子鸟巢和当前最优鸟巢中第 j 个量子比特, 在 Bloch 球面上的对应点分别为 $\mathbf{P}_{ij} = [p_{ijx}, p_{ijy}, p_{ijz}]$ 和 $\mathbf{P}_{bj} = [p_{bjx}, p_{bjy}, p_{bjz}]$. 显然, 搜索的目的是让 $|\varphi_{ij}(t)\rangle$ 向着 $|\varphi_{bj}(t)\rangle$ 移动. 根据量子力学原理, 量子比特在 Bloch 球面上的移动可以通过使其绕着某个轴旋转实现. 由文献[15]可知, 旋转轴可按下式定义:

$$\mathbf{R}_{ij} = \frac{\mathbf{P}_{ij} \times \mathbf{P}_{bj}}{\|\mathbf{P}_{ij} \times \mathbf{P}_{bj}\|} \quad (11)$$

根据量子计算原理, 旋转算子为一个二维酉矩阵, 该矩阵由旋转轴、旋转角度共同决定.

在 QICS 中, 旋转角度采用 Levy 飞行理论设计. 令 $\tilde{\theta}_{ij}$ 为 \mathbf{P}_{ij} 和 \mathbf{P}_{bj} 之间的夹角, 则 $|\varphi_{ij}(t)\rangle$ 的旋转角度可按下式设计:

$$\theta_{ij} = a_0 \times \frac{\varphi \times \mu}{|\nu|^{1/\lambda}} \times \tilde{\theta}_{ij} \quad (12)$$

其中 a_0 为控制参数(通常取 0.1), μ, ν 为服从标准正态分布的随机数, ϕ 按式(4)计算.

至此, 当前量子比特 $|\varphi_{ij}(t)\rangle$ 在 Bloch 球面上, 绕轴 \mathbf{R}_{ij} 转向目标比特 $|\varphi_{bj}(t)\rangle$ 的旋转矩阵为:

$$\mathbf{M}(\theta_{ij}) = \cos \frac{\theta_{ij}}{2} \mathbf{I} - i \sin \frac{\theta_{ij}}{2} (\mathbf{R}_{ij} \times \boldsymbol{\sigma}) \quad (13)$$

其中 \mathbf{I} 为单位矩阵, $\boldsymbol{\sigma} = [\sigma_x, \sigma_y, \sigma_z]$.

$|\varphi_{ij}(t)\rangle$ 转向 $|\varphi_{bj}(t)\rangle$ 的旋转操作为:

$$|\varphi_{ij}(t)\rangle = \mathbf{M}(\theta_{ij})|\varphi_{ij}(t)\rangle \quad (14)$$

其中 t 为迭代步数.

(2) 基于发现概率的抛弃解的补充

在 QICS 中, 依鸟巢被宿主发现的概率舍弃一些解, 同时补充相同数量的新解, 以保持种群规模不变. 这一过程本质上是一种类似于差分进化策略的局部搜索. 具体实现方法为, 首先生成一个均匀分布的随机数, 若该随机数小于发现概率, 则抛弃当前解, 同时补充新解. 补充方法如下:

令 $|\varphi_i(t)\rangle$ 为当前鸟巢, $|\varphi_j(t)\rangle$ 、 $|\varphi_k(t)\rangle$ 、 $|\varphi_l(t)\rangle$ 为从当

代种群随机选取的三个鸟巢, 且 $i \neq j \neq k \neq l$, 记 θ_{jk} 为 $|\varphi_j(t)\rangle$ 和 $|\varphi_k(t)\rangle$ 之间的夹角, 以 $|\varphi_i(t)\rangle$ 和 $|\varphi_l(t)\rangle$ 在 Bloch 球面对应向量的张量积为旋转轴, 以 $\theta_{il} = a_0 \times rand \times \theta_{jk}$ 为旋转角度, 使 $|\varphi_i(t)\rangle$ 向着 $|\varphi_l(t)\rangle$ 旋转即可, 其中 $rand$ 为 (0, 1) 内均匀分布的随机数.

2.3 QICS 的终止条件

关于群智能优化算法的终止条件, 通常有多种. 例如精度控制: 当优化结果达到某一精度阈值, 算法终止; 误差控制, 当优化结果与精确结果的绝对误差小于给定阈值, 算法终止; 进化速度控制: 若连续若干代优化结果无变化, 算法终止; 步数控制: 若寻优步数达到限定步数, 算法终止. 本文提出的 QICS 采用步数控制作为终止条件.

2.4 QICS 的实施方案

关于本文提出的 QICS, 具体实施方案如下.

Step1. 初始化: 鸟蛋被巢主鸟发现的概率 Pa; 种群规模 NP; 空间维数 D; 限度步数 M; 初始种群.

Step2. 计算目标函数值, 记录最优解, 置步数 $t=0$.

Step3. 主循环开始

Step3.1. 所有个体执行 Levy 飞行;

Step3.2. 所有个体按概率 Pa 替换为新解;

Step3.3. 更新全局最优解; $t=t+1$; 若 $t>M$, 转

Step4; 否则转 Step3.1.

Step4. 保存全局最优解, 结束.

3 对比实验

3.1 测试函数

采用 CEC2013 定义的 28 个标准函数^[16]作为优化对象, 如表 1 所示. 其中(S)表示该函数的变量是可分离的, (N)表示变量是不可分离的, n 表示该函数由其他 n 个基本函数复合而成, 所有函数均为极小值优化.

3.2 对比方案

为体现所提出的 QICS 算法的优良性能, 所有函数的优化结果将与普通布谷鸟搜索算法(CS)^[1]、基于高斯扰动的布谷鸟搜索算法(Gauss-based Cuckoo Search Algorithm, GCS)^[2]、混合 CS 算法的 DE 算法(CS-DE)^[3]进行对比. 为保证对比结果客观公正, 四种算法采用相同的种群规模. 考虑到群智能优化算法具有随机性, 为尽量提高对比结果的可信度, 所有函数的维数均取 $D=30$ 维, 且每个函数均用 4 种算法各自独立优化 30 次, 取 30 次优化的平均结果作为对比指标.

3.3 参数设置

关于本文提出的 QICS, 除引入量子计算机制外, 并未增加新的控制参数, 其控制参数个数与普通 CS 是相同的. 关于这些算法的参数设置: 发现概率 Pa=0.25; 种群规模 NP=50; 空间维数 D=30; CS-DE 中的交叉概率 Pc=0.9, 加权因子 F=0.5.

表 1 28 个 CEC'13 测试函数简介

序号	名称	最小值
1	Sphere (S)	-1400
2	Rotated High Conditioned Elliptic (N)	-1300
3	Rotated Bent Cigar (N)	-1200
4	Rotated Discus (N)	-1100
5	Different Powers (S)	-1000
6	Rotated Rosenbrock's (N)	-900
7	Rotated Schaffers F7 (N)	-800
8	Rotated Ackley's (N)	-700
9	Rotated Weierstrass (N)	-600
10	Rotated Griewank's (N)	-500
11	Rastrigin's (S)	-400
12	Rotated Rastrigin's (N)	-300
13	Non-Continuous Rotated Rastrigin's (N)	-200
14	Schwefel's (N)	-100
15	Rastrigin's Schwefel's (N)	100
16	Rotated Katsuura (N)	200
17	Lunacek Bi_Rastrigin (S)	300
18	Rotated Lunacek Bi_Rastrigin (N)	400
19	Expanded Griewank's plus Rosenbrock's (N)	500
20	Expanded Scaffer's F6 (N)	600
21	Composition Function 1 (n=5, Rotated) (N)	700
22	Composition Function 2 (n=3, Unrotated)(S)	800
23	Composition Function 3 (n=3, Rotated) (N)	900
24	Composition Function 4 (n=3, Rotated) (N)	1000
25	Composition Function 5 (n=3, Rotated) (N)	1100
26	Composition Function 6 (n=5, Rotated) (N)	1200
27	Composition Function 7 (n=5, Rotated) (N)	1300
28	Composition Function 8 (n=5, Rotated) (N)	1400

搜索范围: [-100, 100]^D

3.4 对比结果

由于本文提出的 QICS, 需要计算旋转矩阵, 计算量较大, 因此首先考察相同迭代步数下 QICS 与 CS 的优化性能对比. 迭代步数取 1000, 每个函数采用两种算法分别优化 30 次的平均结果对比如表 2 所示.

在表 2 中, 粗体数字表示该结果优于对比算法的结果. 由此可知, 全部 28 个函数中, QICS 有 27 个函数的优化结果优于 CS, 只有第 15 个函数的优化结果劣于 CS. 实验结果充分展示了 QICS 的优势. 以函数 5、8、10、24 为例, 两种算法 30 次优化的平均结果随迭代步数的变化情况如图 1 所示.

就平均运行时间而言, 记 T_{QICS}^i 和 T_{CS}^i 分别表示量子衍生算法和原算法第 i 个函数 30 次优化的平均时间, 由表 2 可知, 两种算法优化 28 个函数的平均时间之比为 $\sum_{i=1}^{28} \frac{T_{QICS}^i}{T_{CS}^i} = 4.4196$.

表 2 QICS 和 CS 每个函数 30 次优化的平均结果对比

序号	QICS		CS	
	平均结果	平均时间(s)	平均结果	平均时间(s)
1	-1399.999	7.34918	3089.971	1.67857
2	17679497	8.14407	48626351	1.94689
3	2.40E+09	8.68184	1.00E+10	1.98121
4	65341.76	7.06032	100047.5	1.74097
5	-999.9738	8.03946	592.8172	1.66297
6	-873.4480	7.078	-571.1776	1.71913
7	-693.0851	11.6829	1240.300	2.61146
8	-679.0587	11.0676	-678.9645	2.32129
9	-570.7860	32.6465	-562.6684	7.36325
10	-492.8885	9.87705	40.06197	2.02489
11	-285.8642	8.99333	4.849442	2.10289
12	-53.24979	10.8383	599.9074	2.29633
13	61.02142	10.5667	769.7226	2.30569
14	3171.966	10.074	3597.338	2.15593
15	5378.091	9.61325	5271.607	2.17465
16	202.5119	12.7059	202.5646	2.98274
17	522.4933	8.06497	1245.551	1.88137
18	707.0253	9.04823	1700.191	2.02177
19	517.1357	7.2462	1784.838	1.77841
20	614.9117	9.00951	614.9999	2.13409
21	1033.003	13.4557	2151.111	3.21674
22	4782.996	16.7057	5578.398	3.67226
23	7066.937	16.616	7329.652	3.85634
24	1279.312	44.4765	1346.994	9.08238
25	1399.829	41.5765	1475.156	9.19782
26	1401.114	40.853	1404.609	10.1182
27	2096.309	41.7117	2611.411	9.92478
28	1795.888	22.07	6892.718	4.84539

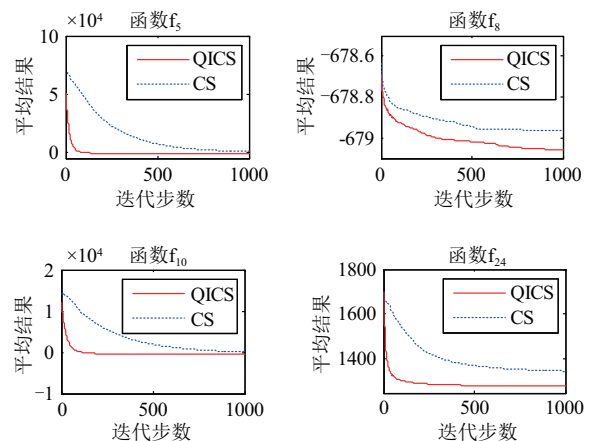


图 1 两种算法 30 次优化的平均结果对比

为充分验证 QICS 的优势,有必要在相同运行时间下对比优化结果.因此,在下面的仿真中,在保持 QICS 迭代步数不变(仍然为 1000 步)的前提下,将 CS、GCS、CS-DE 三种算法的迭代步数均设为 5000 步,然后将每种算法分别独立运行 30 次,四种算法的平均结果对比如表 3 所示.

表 3 四种算法每个函数 30 次优化的平均结果对比

序号	QICS	CS	GCS	CS-DE
1	-1399.999	-1243.473	-1255.088	-1377.138
2	17679497	20291066	18093816	16712608
3	2.40E+09	1.00E+10	1.00E+10	1.00E+10
4	65341.76	91778.11	90639.67	91340.83
5	-999.9738	-860.7984	-881.2796	-969.1241
6	-873.4480	-800.7932	-807.1774	-824.4310
7	-693.0851	-456.0740	-524.6444	-581.1530
8	-679.0587	-679.0046	-679.0140	-679.0412
9	-570.7860	-563.157	-563.5497	-564.4961
10	-492.8885	-444.5689	-454.5131	-472.4276
11	-285.8642	-163.4158	-175.1705	-284.2575
12	-53.24979	500.0011	482.5504	388.2055
13	61.02142	602.3583	566.8238	464.1083
14	3171.966	3140.948	3042.930	2122.688
15	5378.091	4945.689	4820.876	4852.221
16	202.5119	202.4662	202.4182	202.1585
17	522.4933	935.8001	944.8681	610.4416
18	707.0253	1410.824	1393.841	1230.919
19	517.1357	552.6343	549.8897	526.9851
20	614.9117	614.9999	614.9995	614.9946
21	1033.003	1214.398	1183.954	1070.178
22	4782.996	4814.046	4769.329	3414.404
23	7066.937	7029.479	6956.162	6793.155
24	1279.312	1331.425	1330.434	1317.540
25	1399.829	1460.511	1459.150	1447.479
26	1401.114	1401.120	1401.067	1400.974
27	2096.309	2405.466	2355.786	2091.982
28	1795.888	6450.134	6278.773	5576.455

在表 3 中, QICS 列的粗体数字表示该结果是四种算法中最优的.而其他三列中的粗体数字表示该结果优于 QICS 算法.由此可知, QICS 优于 CS 的函数个数为 24 个, 优于 GCS 的函数个数为 22 个, 优于 CS-DE 的函数个数为 20 个.实验结果表明 QICS 明显优于三种对比算法.

3.5 对结果的分析

在 28 个测试函数中,前 5 个函数为单峰函数,这类函数寻优难度较低,除变量不可分离的 F2 外,对于其他 4 个, QICS 均优于对比算法.对于 15 个基本多峰函数函数 F6-F20,除 F11 和 F17 外都是变量不可分离的.这些函数寻优难度较大,除 F14-F16 外,对于其他

12 个, QICS 均优于对比算法.对于最后 8 个变量不可分离的复合函数,寻优难度最大,此时 QICS 仍有 4 个函数优于对比算法.实验结果进一步揭示了量子计算机制的引入对寻优能力提高的促进作用.

QICS 的优良性能在于其编码方式和寻优机制.首先,基于 Bloch 球面描述的量子比特的编码机制,将优化变量的每一维数值,都映射为 Bloch 球面上的某一点.这样就将传统方法中优化变量在优化空间每一维上的线性搜索,转化为 Bloch 球面上的搜索.根据 Bloch 球面性质,优化空间中全局最优解每一维的数值,都对应于 Bloch 球面上的一个圆周.只要搜索到该圆周上的任意一点,都等价于找到了该维的全局最优解.因此这种编码方式在一定程度上提高了 QICS 获得全局最优解的概率.第二, QICS 的搜索机制采用了量子比特绕轴旋转的方法,这种方法可以自动实现量子比特的两个相位参数 θ 和 ϕ 的最佳匹配,从而保证了量子比特在 Bloch 球面上沿着最短路径向着目标位置逼近,而基于 Levy 飞行理论的旋转角度确定方法,有效提高了种群的多样性,一定程度上可以避免早熟收敛,从而进一步提升了 QICS 的搜索能力.

4 结论

经典布谷鸟搜索算法的搜索能力有待于进一步提高.本文尝试将量子计算机制引入经典布谷鸟搜索算法,用量子比特编码鸟巢位置,用量子比特在 Bloch 球面上的绕轴旋转代替经典布谷鸟搜索算法在优化空间中的随机游动,以期较大幅度地提高其优化能力.实验结果表明,这种尝试是成功的,新算法的确呈现出明显优于原算法的性能,从而揭示出,在原算法中引入量子计算机制,是提高其优化性能的有效途径.

参考文献

- 1 Yang XS, Deb S. Cuckoo search via Levy flights. Proc. of World Congress on Nature & Biologically Inspired Computing. India. 2009. 210-214.
- 2 王凡, 贺兴时, 王燕. 基于高斯扰动的布谷鸟搜索算法. 西安工程大学学报, 2011, 25(4): 566-569.
- 3 李明, 曹德欣. 混合 CS 算法的 DE 算法. 计算机工程与应用, 2012, 49(9): 57-60.
- 4 马卫, 孙正兴. 采用搜索趋化策略的布谷鸟全局优化算法. 电子学报, 2015, 43(12): 2429-2439. [doi: 10.3969/j.issn.0372-2112.2015.12.013]

- 5 周欢, 李煜. 具有动态惯性权重的布谷鸟搜索算法. 智能系统学报, 2015, 10(4): 645–651.
- 6 贺兴时, 李娜, 杨新社, 等. 多目标布谷鸟搜索算法. 系统仿真学报, 2015, 27(4): 731–737.
- 7 杨辉华, 谢谱模, 张晓凤, 等. 求解多目标优化问题的改进布谷鸟搜索算法. 浙江大学学报(工学版), 2015, 49(8): 1600–1608.
- 8 高云龙, 闫鹏. 基于多种群粒子群算法和布谷鸟搜索的联合寻优算法. 控制与决策, 2016, 31(4): 601–608.
- 9 李荣雨, 刘洋. 基于梯度的自适应快速布谷鸟搜索算法. 运筹学学报, 2016, 20(3): 45–56.
- 10 贾云璐, 刘胜, 宋颖慧. 基于种群特征反馈的布谷鸟搜索算法. 控制与决策, 2016, 31(6): 969–975.
- 11 董朝阳, 路遥, 江未来, 等. 基于布谷鸟搜索算法的一类变体飞行器容错控制. 航空学报, 2015, 36(6): 2047–2054.
- 12 倪百秀, 张翠翠, 周本达. 基于改进布谷鸟搜索的人工神经网络及其性能仿真. 江汉大学学报(自然科学版), 2015, 43(1): 41–50.
- 13 明波, 黄强, 王义民, 等. 基于改进布谷鸟算法的梯级水库优化调度研究. 水利学报, 2015, 46(3): 341–349.
- 14 李东升, 高扬, 雍爱霞. 基于改进离散布谷鸟算法的干扰资源分配研究. 电子与信息学报, 2016, 38(4): 899–905.
- 15 Benenti G, Casati G, Strini G. Principles of quantum computation and information (Volume I: Basic Concepts). Singapore: World Scientific Publishing, 2004: 100–112.
- 16 Liang JJ, Qu BY, Suganthan PN, *et al.* Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization. http://www3.ntu.edu.sg/home/EPNSugan/index_files/CEC2013/Definitions%20of%20%20CEC%2013%20benchmark%20suite%200117.pdf.