

# 基于自寻优和交叉寻优的量子优化算法

曹茂俊, 尤文菁, 卢玉莹

(东北石油大学 计算机与信息技术学院, 黑龙江 大庆 163318)

**摘要:**为提高群智能优化算法的搜索能力,提出了一种量子优化算法。该算法基于量子位 Bloch 坐标,将种群分为最优个体和普通个体两部分并进行不同处理,对于最优个体,通过使所有量子比特在 Bloch 球面上绕着坐标轴多次旋转,生成多个新个体,并采用贪婪搜索策略选择最优个体;对于普通个体,将当前个体的量子比特向着随机交叉确定的目标位置旋转,生成新个体,并在当前个体和新个体之间通过贪婪选择以实现当前个体的交叉寻优。函数极值优化的仿真结果表明,所提算法在优化能力上,优于简单量子遗传算法、普通遗传算法和人工鱼群算法,从而验证了算法的有效性。该算法在高维能很好地避免陷入局部最优值,具有快速收敛性和良好的全局搜索能力,实验结果揭示出采用量子计算设计优化算法进而提升搜索能力的研究思路是可行的。

**关键词:**优化算法;自寻优;交叉寻优;量子衍生算法;Bloch 球面坐标

中图分类号:TP301

文献标识码:A

文章编号:1673-629X(2022)07-0008-07

doi:10.3969/j.issn.1673-629X.2022.07.002

## Quantum Optimization Algorithm Based on Self-optimization and Cross-optimization

CAO Mao-jun, YOU Wen-jing, LU Yu-ying

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

**Abstract:**To improve the search capability of the group intelligence optimization algorithm, a quantum optimization algorithm is proposed. Based on Bloch coordinates, the algorithm divides the population into optimal individuals and ordinary individuals and performs different processing. For the optimal individual, by rotating all qubits around the coordinate axis, multiple new individuals are generated and using the greedy search strategy, the optimal individual is selected. For current individuals in the population, the cross-optimization of the current individual by determining random crossings with other individuals and rotating the qubits towards the target position generates a new individual with greedy selection between the current individual and the new individual. Simulation results of functional extremal optimization show that the proposed algorithm outperforms the simple quantum genetic algorithm, the common genetic algorithm and the artificial fish swarms algorithm in terms of optimization ability, thus verifying the effectiveness of such algorithm. The algorithm avoids falling into local optimum in high dimensions, with fast convergence and strong global search capability. The experimental results reveal that the optimization algorithm is feasible.

**Key words:** optimization algorithm; self-optimization; cross-optimization; quantum inspired algorithm; Bloch coordinates

## 0 引言

启发式算法常常用来解决某些难以得到精确解的优化问题,根据搜索过程中所依赖的解的数量,启发式算法可分为单解和群解两类。常见的单解启发式算法有模拟退火算法<sup>[1]</sup>、禁忌搜索算法<sup>[2]</sup>等,而群解启发式算法的代表有差分进化算法<sup>[3]</sup>、遗传算法<sup>[4]</sup>、布谷鸟搜索算法<sup>[5]</sup>等。就搜索特性而言,单解和群解算法在开发和探索上各有优劣。自从20世纪80年代 Feynman<sup>[6]</sup>首次提出利用量子计算模拟量子力学过程

的思想<sup>[7]</sup>,Deutsch 提出构建通用量子计算机的可行性<sup>[8]</sup>,人们开始关注如何利用量子特性来解决传统计算机难以解决的问题,而量子计算也逐渐出现了量子算法和量子衍生技术这两个不同的发展方向。而将量子特性引入启发式算法,就出现了许多量子优化算法,如量子遗传算法<sup>[9]</sup>、量子粒子群优化算法<sup>[10]</sup>等。

目前许多启发式算法优化效率低、容易陷入局部最优值、高维优化效果差,对于一些群解算法,还存在着进化时不对种群个体作区分、不考虑个体差异性等

收稿日期:2021-08-01

修回日期:2021-12-02

基金项目:黑龙江省自然科学基金(LH2019F004)

作者简介:曹茂俊(1978-),男,副教授,博士,研究方向为量子衍生优化算法、机器学习、软件工程和地球探测与信息技术等。

问题。而在量子态的描述方式上,多数算法也只在平面单位圆上描述量子态<sup>[11-13]</sup>,削弱了量子计算的特性。对此,在利用 Bloch 球面来对量子态进行描述和变换的前提下,提出一种群解算法。针对种群的最优个体和普通个体分别提出优化策略,对于最优个体采用单解启发算法中涡流搜索算法的自优化策略,对于普通个体则采用群解启发算法中差分进化的交叉策略,即最优个体自寻优策略和普通个体交叉寻优策略。两种寻优策略有机融合可构成一种全新的量子衍生优化模型,基于自寻优和交叉寻优的量子优化算法 (quantum optimization algorithms based on self-optimization and cross-optimization, QOA-SAC)。通过函数极值优化问题进行仿真,并通过与普通遗传算法 (common genetic algorithm, CGA)、简单量子遗传算法 (simple quantum genetic algorithm, SQGA) 和人工鱼群算法 (artificial fish swarms algorithm, AFSA) 对比,验证了 QOA-SAC 的有效性。

## 1 基于量子比特的个体编码和解码

### 1.1 量子个体编码

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,根据量子叠加原理,量子比特的任何状态都可以写成:

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

其中,  $0 \leq \theta \leq \pi$ ,  $0 \leq \varphi \leq 2\pi$ ,  $i = \sqrt{-1}$ 。

量子比特可以借助 Bloch 球面描述,该球为量子比特及其变换提供了几何图像,具体而言,量子比特可以用嵌入三维笛卡尔坐标系中的 Bloch 球面上的一个点来描述。

量子比特  $|\varphi\rangle$  可以写成:  $|\varphi\rangle = [\cos \frac{\theta}{2}, e^{i\varphi} \sin \frac{\theta}{2}]^T = [\frac{1+z}{2}, \frac{x+iy}{\sqrt{2(1+z)}}]^T$ , 因此,

Bloch 球面上的任意一点  $P(x, y, z)$  与一个量子比特  $|\varphi\rangle$  一一对应。令优化空间为  $n$  维,第  $i$  个个体可编码为  $[|\varphi_{i1}\rangle, |\varphi_{i2}\rangle, \dots, |\varphi_{in}\rangle]$ 。

### 1.2 解空间变换

为了减少计算量,提高优化效率,设  $X_i = [x_{i1}, x_{i2}, \dots, x_{in}]$  为问题可行解,其中  $x_{ij} \in [-1, 1]$ ,按下式将  $x_{ij}$  变换为实际优化空间  $[\text{Min}_j, \text{Max}_j]$  中的解  $X_{ij}$ 。

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

## 2 算法设计与实现

### 2.1 最优个体的自寻优策略

对于最优个体上的所有量子比特,分别确定旋转轴、计算旋转角度、实施旋转操作,所有量子比特都被

旋转之后,即可生成一个新个体,重复上述操作,即可生成多个新个体。将这些新个体进行解码、计算目标函数值之后,利用贪婪选择策略即可实现当代最优个体的更新。

#### 2.1.1 旋转轴的确定方法

若在 Bloch 球面上随机选择旋转轴,不仅会显著增加模型的复杂度,而且当选择的轴与  $x$  轴夹角较小时,还会严重降低优化效率。所以在旋转时要根据量子比特在 Bloch 球面上的具体位置选择  $y$  轴或  $z$  轴作为旋转轴。

令最优个体上第  $j$  个量子比特  $|\varphi_j\rangle$  的 Bloch 坐标为  $(x_j, y_j, z_j)$ ,若使  $|\varphi_j\rangle$  绕  $y$  轴旋转,则旋转半径为  $\sqrt{x_j^2 + z_j^2}$ ,若使  $|\varphi_j\rangle$  绕  $z$  轴旋转,则旋转半径为  $\sqrt{x_j^2 + y_j^2}$ ,由于旋转半径越大,对  $x_j$  的调整幅度也越大,所以,若  $|y_j| \leq |z_j|$ ,选择  $y$  轴作为旋转轴,否则选择  $z$  轴作为旋转轴。

#### 2.1.2 旋转角度的确定方法

在确定了量子比特旋转轴之后,还需要确定旋转角度。由于自寻优策略中量子比特的“移动”没有确定的目标位置,所以无法根据目标位置确定转角步长。此时对于最优个体上的所有量子比特,其转角步长的确定可以设计一个不依赖于目标位置的统一原则。考虑到最优个体的量子比特在 Bloch 球面上移动的随机性,转角方向应该可正可负,又考虑到当前个体的最优性,小步长的探索次数应该多于大步长的探索次数。在 QOA-SAC 中,旋转角度拟采用中心为 0 标准差逐代缩小的高斯分布随机产生。对于最优个体上第  $j$  个量子比特  $|\varphi_j\rangle$ ,其旋转角度的计算式如公式(3)所示。

$$\delta_j(t) = \sigma_j(t) \times \text{randn} \quad (3)$$

式中,randn 为服从  $N(0,1)$  分布的正态随机数, $t$  为迭代步数。

下面给出标准差  $\sigma_j(t)$  的计算方法。在算法执行的开始阶段,寻优偏重于全局探索,旋转角度应相对较大,而在后期阶段则偏重于局部开发,旋转角度应相对较小。为实现探索和开发两阶段的平衡,QOA-SAC 拟采用使旋转角度的标准差  $\sigma_j(t)$  随迭代步数逐代减小的策略,具体可采用逆不完全伽马函数实现  $\sigma_j(t)$  的计算。

逆不完全伽马函数是不完全伽马函数  $\gamma(\lambda, a) = \int_0^\lambda e^{-t} t^{a-1} dt$  关于积分上限  $\lambda$  的逆函数,其中  $\lambda \geq 0$  是随机变量, $a > 0$  是分辨率参数。若使  $\lambda$  取定值,而使  $a$  随迭代步数以等步长  $\Delta a$  从 0 线性增加到 1,则逆不完全伽马函数  $(1/\lambda) \text{gammaincinv}(\lambda, 1 - a)$  具有从略大于 1 单调下降到接近于 0 的特性,符合算法对于调整

最优解量子比特转角标准差  $\sigma_j(t)$  的要求。

设  $\lambda = 0.1$ ,  $a = t/\text{MaxItr}$ , 其中  $\text{MaxItr}$  为限定迭代步数, 此时逆不完全伽马函数具有这样一个优良特性: 在迭代步数的前一半, 接近线性下降, 而在后一半, 接近指数下降<sup>[14]</sup>, 如图 1 所示。用这样的函数来动态调整最优解量子比特的转角标准差  $\sigma_j(t)$ , 能够较好地实现探索与开发两阶段之间的平衡, 进而有效避免早熟收敛。根据以上思路, 最优个体量子比特旋转角度的标准差  $\sigma_j(t)$  可按公式(4)计算。

$$\sigma_j(t) = \sigma_j(0) (1/\lambda) \text{gammaincinv}(\lambda, 1 - t/\text{MaxItr}) \quad (4)$$

式中,  $\sigma_j(0)$  为标准差初值。利用(公式 3, 4), 即可获得最优个体量子比特的旋转角度  $\delta_j(t)$ 。

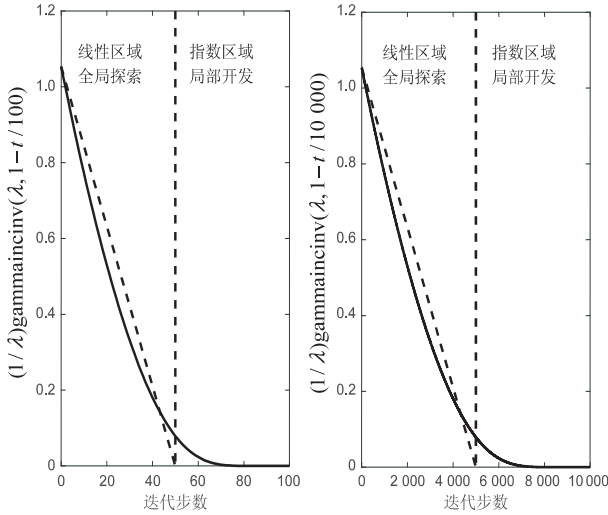


图 1 逆不完全伽马函数在  $\lambda = 0.1$ , 限定步数取 100 和 10 000 时的特性

表 1 QOA-SAC 拟采用的交叉策略

序号	交叉策略	说明
1	$\tilde{x}_{ij} = x_{pj}$	选择随机 $p$
2	$\tilde{x}_{ij} = x_{bj}$	选择最好 $b$
3	$\tilde{x}_{ij} = \text{rnd} \cdot x_{pj} + (1 - \text{rnd})x_{bj}$	随机 $p$ 和最好 $b$ 交叉
4	$\tilde{x}_{ij} = \text{rnd} \cdot x_{pj} + (1 - \text{rnd})x_{qj}$	随机 $p$ 和随机 $q$ 交叉
5	$\tilde{x}_{ij} = \text{rnd}_1 \cdot x_{pj} + (1 - \text{rnd}_1)(\text{rnd}_2 x_{pj} + (1 - \text{rnd}_2)x_{bj})$	随机 $p$ 先与最好 $b$ 交叉, 交叉的结果再与随机 $p$ 交叉
6	$\tilde{x}_{ij} = \text{rnd}_1 \cdot x_{pj} + (1 - \text{rnd}_1)(\text{rnd}_2 x_{qj} + (1 - \text{rnd}_2)x_{rj})$	随机 $q$ 先与随机 $r$ 交叉, 交叉的结果再与随机 $p$ 交叉
7	$\tilde{x}_{ij} = \text{rnd}_1 \cdot x_{bj} + (1 - \text{rnd}_1)(\text{rnd}_2 x_{pj} + (1 - \text{rnd}_2)x_{qj})$	随机 $p$ 先与随机 $q$ 交叉, 交叉的结果再与最好 $b$ 交叉

## 2.2.2 目标位置的确定方法

关于目标位置的确定, 令当前个体某个量子比特在 Bloch 球面上的对应点为  $P_0$ , Bloch 坐标为  $(x_0, y_0, z_0)$ , 应用交叉策略确定的目标位置的横坐标为  $x$ 。过点  $(x, 0, 0)$  作与  $x$  轴垂直的圆周  $C$ , 记平面  $XOP_0$  与圆周  $C$  的交点为  $P$  和  $M$ , 且点  $P$  的坐标  $(x, y, z)$  与  $P_0$  的坐标  $(x_0, y_0, z_0)$  同号。对于圆周  $C$  上的任意一点  $P_1$ ,

## 2.1.3 最优个体自寻优策略的具体实现

确定了旋转轴和旋转角度之后, 即可对  $|\varphi_j\rangle$  实施旋转操作, 绕  $y$  轴和绕  $z$  轴旋转的旋转矩阵分别如公式(5)和公式(6)所示。

$$R_y(\delta) = \begin{bmatrix} \cos(\frac{\delta}{2}) & -\sin(\frac{\delta}{2}) \\ \sin(\frac{\delta}{2}) & \cos(\frac{\delta}{2}) \end{bmatrix} \quad (5)$$

$$R_z(\delta) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\delta} \end{bmatrix} \quad (6)$$

具体的旋转操作可表述为:  $|\varphi_j'\rangle = R_y(\delta_j(t)) |\varphi_j\rangle$  或  $|\varphi_j'\rangle = R_z(\delta_j(t)) |\varphi_j\rangle$ 。

## 2.2 普通个体的交叉寻优策略

对于种群中的当前个体, 首先通过其他个体的随机交叉确定目标位置, 然后使当前个体的量子比特在 Bloch 球面上绕着某一旋转轴, 向着目标位置的量子比特旋转, 即可生成一个新个体, 在当前个体和新个体之间通过贪婪选择, 即可实现当前普通个体的交叉寻优。

### 2.2.1 普通个体寻优中多种交叉策略的设计

记第  $i$  个个体中第  $j$  个量子比特为  $|\varphi_{ij}\rangle$ , 对应的横坐标为  $x_{ij}$ , 最优个体中第  $j$  个量子比特对应的横坐标为  $x_{bj}$ , 第  $p, q, r$  个个体中第  $j$  个量子比特对应的横坐标分别为  $x_{pj}, x_{qj}, x_{rj}$ , 其中  $p, q, r$  为在种群中随机选择的互不相同且不等于  $i$  的三个个体。令  $\tilde{x}_{ij}$  为交叉结果, 即  $\tilde{x}_{ij}$  为  $|\varphi_{ij}\rangle$  希望逼近的目标位置的横坐标。为确定  $\tilde{x}_{ij}$  的具体取值, QOA-SAC 拟采用的交叉策略如表 1 所示。

由于平面  $XOP_0$  垂直于圆周  $C$ , 显然有路径  $P_0P$  的长度小于路径  $P_0P_1$  的长度, 所以, 点  $P$  即为  $P_0$  希望逼近的目标位置, 目标位置  $P$  的纵坐标  $y$  和竖坐标  $z$  可按公式(7)计算。

$$y = y_0 \frac{\sqrt{1-x^2}}{\sqrt{1-x_0^2}}, z = z_0 \frac{\sqrt{1-x^2}}{\sqrt{1-x_0^2}} \quad (7)$$

至此, 普通个体寻优中当前个体的目标位置得以

完全确定。

### 2.2.3 旋转角度的确定方法

将当前个体向目标位置进行旋转,需要确定旋转角度。关于旋转角度,如果采用向量  $\overrightarrow{OP_0}$  和  $\overrightarrow{OP}$  之间的夹角,种群将很快失去多样性并陷入早熟收敛。令点  $P_0$  坐标为  $(x_0, y_0, z_0)$ , 点  $P$  坐标为  $(x, y, z)$ , randn 为服从  $N(0,1)$  分布的正态随机数, QOA-SAC 拟采用公式(8)确定  $|\varphi_{ij}\rangle$  的旋转角度。

$$\delta_{ij} = \arccos(xx_0 + yy_0 + zz_0)(1 + \text{randn}/3) \quad (8)$$

这种确定旋转角度的方法,是先使  $|\varphi_{ij}\rangle$  从  $P_0$  旋转到  $P$ , 然后再从  $P$  点开始旋转一个服从  $N(0, \arccos(xx_0 + yy_0 + zz_0)/3)$  分布的随机转角。

### 2.2.4 旋转轴的确定方法

根据量子计算原理,使量子比特  $|\varphi_{ij}\rangle$  在 Bloch 球面上绕一个沿单位矢量  $n = [n_x, n_y, n_z]$  的轴转动  $\delta$  弧度的旋转矩阵如公式(9)所示。

$$M_n(\delta) = \cos \frac{\delta}{2} I - i \sin \frac{\delta}{2} (n_x \sigma_x + n_y \sigma_y + n_z \sigma_z) \quad (9)$$

式中,  $I$  为单位矩阵,  $\sigma_x$ 、 $\sigma_y$ 、 $\sigma_z$  为泡利矩阵。

根据此公式,即可导出将当前量子比特  $|\varphi_{ij}\rangle$  从  $P_0$  向着  $P$  旋转的旋转矩阵,如公式(10)所示。

$$M_{ij}(\delta_{ij}) = \cos \frac{\delta_{ij}}{2} I - i \sin \frac{\delta_{ij}}{2} \frac{R_{\text{axis}} [\sigma_x, \sigma_y, \sigma_z]^T}{\|R_{\text{axis}}\|} \quad (10)$$

具体的旋转操作可表述为:  $|\varphi'_{ij}\rangle = M_{ij}(\delta_{ij}) |\varphi_{ij}\rangle$ 。

### 2.3 基于自寻优和交叉寻优的量子遗传算法

自寻优策略,就是个体不与种群中任何其他个体交互,完全根据自身特征更新自己,研究最优个体的量子衍生自寻优方法,可以最大限度地发挥最优个体的潜能。

交叉寻优策略则针对普通个体,随机目标的选择有利于增强种群中个体的多样性,避免因过分依赖最优个体而早熟收敛。

将以上两种寻优策略有机融合可构成一种全新的量子衍生优化模型,即基于自寻优和交叉寻优的量子优化算法 (quantum optimization algorithms based on self-optimization and cross-optimization, QOA-SAC)。

设种群规模为  $N_1$ , 最优解自寻优时生成新个体数为  $N_2$ , 优化空间为  $D$  维。首先实施种群编码,解码,计算目标函数值,记录当前最优解。记最优解的量子比特描述为  $[|\varphi_{b1}\rangle, |\varphi_{b2}\rangle, \dots, |\varphi_{bd}\rangle]$ , Bloch 横坐标为  $[x_{b1}, x_{b2}, \dots, x_{bd}]$ , 最优目标函数值为  $F_{\text{best}}$ 。在每一步迭代中,首先进行最优个体的自寻优。具体为:通过将  $[|\varphi_{b1}\rangle, |\varphi_{b2}\rangle, \dots, |\varphi_{bd}\rangle]$  的所有量子比特绕坐标轴

旋转,生成  $N_2$  个新个体,对这些新个体解码,计算目标函数值,采用贪婪策略更新  $[|\varphi_{b1}\rangle, |\varphi_{b2}\rangle, \dots, |\varphi_{bd}\rangle]$ 。然后进行普通个体的交叉寻优。所有  $N_1$  个个体全部执行交叉寻优之后,完成一步迭代。循环上述过程直到满足终止条件。QOA-SAC 的基本流程如图2所示。

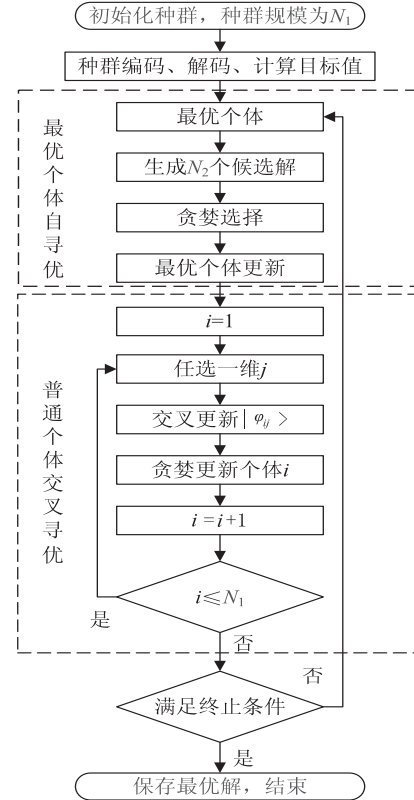


图2 基于自寻优和交叉寻优的量子优化模型

## 3 实验仿真与实验结果

### 3.1 仿真实验环境

该仿真测试环境:操作系统为 Windows10, CPU 为 Intel(R) Core(TM) i7-8550U, 主频 1.80 GHz, 内存为 16 GB, 仿真软件为 Matlab2020b。

### 3.2 测试函数

为验证算法的有效性,选择 8 个标准测试函数进行实验分析,具体为:

$$(1) f_1(x) = \sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|, X \in [-10, 10]^n, f_1(X^*) = 0$$

$$(2) f_2(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1], X \in [-100, 100]^n, f_2(X^*) = 0$$

$$(3) f_3(x) = \exp(1) + 20 - 20\exp(-\frac{1}{5}$$

$$\sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}) - \exp(-\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)),$$

$$X \in [-32, 32]^n, f_3(X^*) = 0$$



$$(4) f_4(x) = \sum_{i=1}^n \frac{x_i^2}{4\,000} - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, X \in [-60, 60]^n, f_4(X^*) = 0$$

$$(5) f_5(x) = \sum_{i=1}^n x_i^2, X \in [-100, 100]^n, f_5(X^*) = 0$$

$$(6) f_6(x) = \sum_{i=1}^n (\lfloor x_i + 0.5 \rfloor)^2, X \in [-100, 100]^n, f_6(X^*) = 0$$

$$(7) f_7(x) = \sum_{i=1}^n \left( \sum_{j=1}^i x_j \right)^2, X \in [-100, 100]^n, f_7(X^*) = 0$$

$$(8) f_8(x) = nA + \sum_{i=1}^n (x_i^2 - A \cos(2\pi x_i)), X \in [-5.12, 5.12]^n, f_8(X^*) = 0$$

其中, (1) 是连续的、平滑多峰函数, 当自变量趋近于无穷大时, 函数会形成大量局部极值区域<sup>[15]</sup>。(2) 是一个偶次多项式, 当自变量为正无穷或负无穷时, 函数值极限值等于无穷。(3) 是 Ackley 函数, 是指数函数叠加上适度放大的余弦而得到的连续型实验函数, 其特征是一个几乎平坦的区域由余弦波调制形成一个个

孔或峰, 从而使曲面起伏不平<sup>[16]</sup>。(4) 是 Griewank 函数, 是智能算法领域经典的测试的函数, 有许多分布广泛的局部极小值, 而这些极小值是有规律分布的。(5) 是球形函数。(6) 是步长函数。(7) 的自变量具有上位性, 因此其梯度方向不会沿着轴线方向变化, 具有较高的寻优难度。(8) 是 Rastrigin 函数, 其特点是高度多峰。

### 3.3 实验结果

许多量子优化算法<sup>[17-18]</sup>实验时均使用低维测试函数, 所以为验证算法的有效性, 先设置维数  $n=2$ , 分别将本算法与 SQGA 和 ASFA 进行比较, 对于测试函数(1)~(8), 当优化函数小于 0.05 时, 认为算法收敛。

三种算法, 种群大小均取 40, 设置最大迭代步数 MaxGen=1 000, 算法进行函数优化时, 若算法收敛, 则退出, 否则迭代到最大步数后退出, 三种算法各自独立执行 30 次, 分别记录执行总时间、目标函数的平均调用次数、均值以及未收敛次数, 实验结果如表 2 所示。

表 2 低维函数优化结果对比(30 次优化)

适应度函数	算法	总用时/s	目标函数平均调用次数	均值	未收敛次数
$f_1$	QGA-SAC	1.310 1	1 255.433 3	0.024 5	0
	SQGA	37.243 8	25 138.666 7	0.085 9	17
	ASFA	4.684 4	15 520.4	0.027 8	0
$f_2$	QGA-SAC	1.834 5	1 783.033 3	0.029 7	0
	SQGA	40.910 8	27 336	3.570 3	19
	ASFA	20.579 3	60 284.233 3	0.028 4	0
$f_3$	QGA-SAC	1.930 6	1 763.433 3	0.028 9	0
	SQGA	59.443 7	40 000	1.362 1	30
	ASFA	15.894 7	56 292.2	0.035 7	0
$f_4$	QGA-SAC	1.266 9	781	0.026 9	0
	SQGA	4.097	2 253.333 3	0.036 4	1
	ASFA	95.166 8	43 657.777 8	0.084 8	14
$f_5$	QGA-SAC	1.988 8	2 109.6	0.026 9	0
	SQGA	47.267 1	37 912	0.730 2	28
	ASFA	7.359 6	17 369.733 33	0.020 6	0
$f_6$	QGA-SAC	1.373 8	1 698.333 333	0.024 5	0
	SQGA	18.783 9	14 314.666 67	0.433 3	7
	ASFA	5.600 5	11 629.766 67	0.000 0	0
$f_7$	QGA-SAC	1.643 5	2 194.833 333	0.026 8	0
	SQGA	44.431 8	35 026.666 67	1.285 3	26
	ASFA	9.008 6	19 206.066 67	0.026 2	0
$f_8$	QGA-SAC	1.402 4	1 717.9	0.028 6	0
	SQGA	14.280 2	11 001.333 33	0.040 4	6
	ASFA	0.923 4	2 867.466 667	0.012 6	0

再设置维数  $n = 30$ , 分别将本算法与 CGA 和 ASFA 进行比较, 由于 SQGA 对高维函数的优化效果过差, 在此不进行比较, 对于测试函数(1)~(8), 当优化函数小于 0.05 时, 认为算法收敛。

三种算法, 种群大小均取 100, 设置最大迭代步数 MaxGen = 5 000, 各自独立运行 10 次, 记录运行总时长、均值、收敛次数和最优值, 实验结果如表 3 所示。

表 3 高维函数优化结果对比(10 次优化)

适应度函数	算法	总用时/s	均值	收敛次数	最优值
$f_1$	QGA-SAC	488.170 7	0.034 7	8	0.003 2
	CGA	1 526.000 0	0.827 5	0	0.626 5
	AFSA	3 889.100 0	0.892 4	0	0.754 7
$f_2$	QGA-SAC	489.232 5	0.066 1	8	0.001 2
	CGA	1 193.000 0	1.471 6	0	0.155 2
	AFSA	3 370.200 0	0.109 6	2	0.045 0
$f_3$	QGA-SAC	476.315 5	0.106 7	4	0.013 5
	CGA	2 155.000 0	18.512 2	0	17.700 5
	AFSA	3 124.500 0	13.445 5	0	12.987 8
$f_4$	QGA-SAC	467.460 3	0.010 9	10	0.000 2
	CGA	2 179.000 0	1.093 2	0	1.069 8
	AFSA	3 470.100 0	1.107 1	0	1.078 0
$f_5$	QGA-SAC	488.214 6	0.175 2	5	0.003 1
	CGA	982.000 0	2.217 6	0	0.854 1
	AFSA	2 801.500 0	415.998 1	0	353.868 8
$f_6$	QGA-SAC	494.089 7	0.200 0	8	0.000 0
	CGA	1 248.000 0	0.100 0	9	0.000 0
	AFSA	2 640.400 0	432.900 0	0	254.000 0
$f_7$	QGA-SAC	404.405 5	0.753 5	0	0.065 4
	CGA	1 809.000 0	4 251.100 2	0	947.058 1
	AFSA	3 399.200 0	5 110.007 7	0	3 836.575 2
$f_8$	QGA-SAC	478.397 9	0.012 0	10	0.000 4
	CGA	1 054.000 0	0.140 8	0	0.059 8
	AFSA	3 293.000 0	3.910 4	0	1.055 3

### 3.4 实验分析

由表 2 可知, 对于低维函数, 相较于 ASFA 和 SQGA, QOA-SAC 有优化时间短、目标函数调用次数少等特点, 同时, 从收敛次数来看, QOA-SAC 表现出极强的稳定性。

由表 3 可知, 对于高维函数, 就优化时间、优化效果和收敛次数而言, 在函数(1)~(5)以及函数(7)~(8)上, QOA-SAC 各方面都表现的最好也最稳定, 仅仅在优化函数(6)时均值和收敛次数相较于 CGA 略有不如, 但考虑到优化时间仅为 CGA 的三分之一, 所以综上, 三种算法中, QOA-SAC 最优, 而 ASFA 和 CGA 互有优劣。QOA-SAC 在函数(7)上优化结果较差, 因为(7)的自变量具有上位性, 因此其梯度方向不会沿着轴线方向变化, 具有较高的寻优难度, 如需更好的优化结果, 需要设定更大的迭代步数。

QOA-SAC 产生优势的具体原因为:

(1) QOA-SAC 基于 Bloch 球面坐标编码, 将优化问题简化为在 Bloch 球面上的搜索问题, 有效增强了解空间的遍历性, 最优解也可以被投射到 Bloch 球面上的一个圆周上, 从而极大地扩充了全局最优解的数量, 该圆周上任意一点被检索到, 都等价于找到了最优解, 提升了算法的搜索能力;

(2) QOA-SAC 的自寻优的核心策略考虑到种群中最优个体本身的优势, 在计算旋转角时采用中心为 0 标准差逐代缩小的高斯分布随机产生, 实现了在算法执行的开始阶段, 寻优偏重于全局探索, 旋转角度较大, 而在后期阶段则偏重于局部开发, 旋转角度较小, 达到探索和开发两阶段间的平衡, 易于跳出局部极值;

(3) QOA-SAC 的普通个体交叉寻优的核心策略通过多种交叉策略的设计以及基于目标点位的随机转

角,保证了种群能持续进化而不失去多样性,多种交叉策略的同时使用,也蕴含了集成学习策略的思想,有助于实现各种交叉策略的优势互补,从而提升算法的搜索能力。

而相较于 CGA、SQGA 和 ASFA 这些经典算法, QOA-SAC 的计算复杂度更高,这是由于 QOA-SAC 在自寻优和交叉寻优时都需要计算旋转轴和旋转角度,并且对于新生成个体要进行贪婪搜索决定的,这也是 QOA-SAC 的缺点,然而该算法正是以此为代价开提升搜索能力的。

#### 4 结束语

针对许多量子优化算法存在的缺点及对高维函数寻优困难等问题,提出了基于自寻优和交叉寻优的量子优化算法。QOA-SAC 首先将种群划分为最优个体和普通个体,提出了不同的寻优策略。对于最优个体,随着迭代过程的进行,搜索范围收束,近距离探索次数增加,能够较好地实现探索与开发两阶段之间的平衡,进而有效避免早熟收敛。其次,对于普通个体采取交叉寻优策略,保证了种群能持续进化而不失去多样性。最后,通过 8 个标准测试函数对各个算法的性能进行检测,仿真结果证明无论在高维还是低维, QOA-SAC 都具有较好的寻优性能。

#### 参考文献:

- [1] KIRKPATRICK S, GELATT C D, VECCHI M P. Optimization by simulated annealing[J]. Science, 1983, 220(4598): 671-680.
- [2] GLOVER F. Future paths for integer programming and links to artificial intelligence[J]. Computers & Operations Research, 1986, 13(5): 533-549.
- [3] STORN R, PRICE K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11(4): 341-359.
- [4] HOLLAND J H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence[M]. [s. l.]: The MIT Press, 2018.
- [5] YANG X S, DEB S. Cuckoo search via levy flights[C]//Proceedings of the world congress on nature & biologically inspired computing. Coimbatore; IEEE, 2009: 210-214.
- [6] FEYNMAN R P. Simulating physics with computers[J]. International Journal of Theoretical Physics, 1982, 21(6-7): 467-488.
- [7] 张毅, 卢凯, 高颖慧. 量子算法与量子衍生算法[J]. 计算机学报, 2013, 36(9): 1835-1842.
- [8] DEUTSCH D. Quantum theory, the church-turing principle and the universal quantum computer[J]. Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences, 1985, 400(1818): 1934-1990.
- [9] 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.
- [10] SHUAI D, LIU Y, SHUAI Q, et al. Self-organizing data clustering based on quantum entanglement model[C]//First international multi-symposiums on computer and computational sciences (IMSCCS'06). Hangzhou; IEEE, 2006: 716-723.
- [11] 马莹, 王怀晓, 刘贺, 等. 一种新的自适应量子遗传算法研究[J]. 计算机工程与应用, 2018, 54(20): 99-103.
- [12] 刘志刚, 杜娟, 许少华, 等. 基于量化正交交叉的量子衍生布谷鸟搜索算法[J]. 信息与控制, 2017, 46(4): 408-414.
- [13] 刘晓红, 曲志坚, 曹雁峰, 等. 基于自适应机制的多宇宙并行量子衍生进化算法[J]. 计算机应用, 2015, 35(2): 369-373.
- [14] 李建平, 宫耀华, 卢爱平, 等. 改进的粒子群算法及在数值函数优化中应用[J]. 重庆大学学报, 2017, 40(5): 95-103.
- [15] 高文欣, 刘升, 肖子雅, 等. 全局优化的蝴蝶优化算法[J]. 计算机应用研究, 2020, 37(10): 2966-2970.
- [16] 柏静. 基于多种混合策略的人工蜂群算法改进研究[D]. 济南: 山东师范大学, 2016.
- [17] 杨俊安, 庄镇泉, 史亮. 多宇宙并行量子遗传算法[J]. 电子学报, 2004, 32(6): 923-928.
- [18] 李絮, 李智勇, 刘松兵, 等. 多宇宙并行量子多目标进化算法[J]. 计算机工程与应用, 2008, 44(27): 37-40.