

DOI: 10.11992/tis.201705007

网络出版地址: <http://kns.cnki.net/kcms/detail/23.1538.TP.20180426.1120.004.html>

求解离散优化问题的元胞量子狼群演化算法

马龙, 卢才武, 顾清华

(西安建筑科技大学管理学院, 陕西 西安 710055)

摘要: 针对离散空间优化问题, 提出了求解离散优化问题的元胞量子狼群演化算法, 首先, 为了提高算法的全局收敛速度, 采用双策略量子位初始化方法和滑模交叉方法, 分别生成量子狼群初始位置和产生头狼, 实现种群多样性; 其次, 为了描述头狼与猎物间的距离以及增强狼群的遍历范围, 采用二进制编码方式和元胞自动机中的演化规则, 分别实现狼群中个体狼与猎物距离的精确描述和量子旋转角的选取调整; 然后, 为了证明该算法的收敛性能, 采用泛函分析方法, 实现了算法全局收敛性能的验证; 最后, 通过 6 个标准测试函数的仿真实验, 并与狼群算法以及量子狼群算法的优化结果进行比较。实验结果表明, 该算法具有较快的收敛速度和较好的全局寻优能力。

关键词: 离散优化; 量子狼群算法; 元胞自动机; 双策略方法; 滑模交叉; 二进制编码; 泛函分析; 狼群算法; 量子旋转角

中图分类号: TP301.6 **文献标志码:** A **文章编号:** 1673-4785(2018)05-0716-12

中文引用格式: 马龙, 卢才武, 顾清华. 求解离散优化问题的元胞量子狼群演化算法[J]. 智能系统学报, 2018, 13(5): 716-727.

英文引用格式: MA Long, LU Caiwu, GU Qinghua. Cellular and quantum-behaved wolf pack evolutionary algorithm for solving discrete optimization problems[J]. CAAI transactions on intelligent systems, 2018, 13(5): 716-727.

Cellular and quantum-behaved wolf pack evolutionary algorithm for solving discrete optimization problems

MA Long, LU Caiwu, GU Qinghua

(School of Management, Xi'an University of Architecture and Technology, Xi'an 710055, China)

Abstract: To solve optimization problems in discrete space, a cellular quantum-inspired wolf pack evolutionary algorithm is proposed for solving discrete optimization problems. First, to speed up the global convergence of the algorithm, when generating the diversity of population, the method fully utilizes the double strategy quantum bit initialization method and the sliding mode crossover method to help generate the initial position and candidate wolf, respectively. Then, to accurately describe the distance between the wolf and the prey as well as enhance the traverse range of wolf pack, the methods of the binary encoding and evolution rules of the cellular automata are used to realize precise description and the selection of the quantum rotation angle, respectively. Then to prove the convergence performance of the algorithm, the method fully utilizes the functional analysis to verify the global convergence. Finally, simulation experiment on six benchmark functions was carried out, and the comparison between the wolf pack algorithm and quantum-inspired wolf pack evolutionary algorithm was provided. The results show that the proposed approach has better convergence speed and great global convergence optimization ability.

Keywords: discrete optimization; quantum-inspired wolf pack algorithm; cellular automata; double strategy method; sliding mode crossover; binary encoding; functional analysis; wolf pack algorithm; quantum rotation angle

收稿日期: 2017-05-08. **网络出版日期:** 2018-04-26.

基金项目: 国家自然科学基金项目 (51774228, 51404182); 陕西省自然科学基金项目 (2017JM5043); 陕西省教育厅专项科研计划项目 (17JK0425).

通信作者: 顾清华. E-mail: qinghuagu@126.com.

在人工智能计算和系统工程等领域中, 许多离散空间优化问题常具有解的多样性、动态性以及目标函数收敛速度慢等特点。为了在有限的空间环境下快速搜寻到优化问题的最优解, 学者们

已对多种智能算法进行融合并展开了广泛的研究和应用。

狼群演化算法 (wolf pack evolutionary algorithm, WPEA) 作为模拟自然界狼群分工协作捕猎的启发式智能优化算法, 1970 年美国著名专家 Mech^[1]在其著作中对狼群的行为特征进行了详细的描述; 2014 年 Mirjalili 等^[2]将该算法与金字塔模型进行结合, 提出了一种具有等级森严的狼群捕猎层次模型, 该算法在解决实数空间问题的应用效果明显。此后针对离散空间优化问题, 文献[3]针对分类特征子集的优化问题, 提出了二进制狼群演化算法; 文献[4-5]提出了一种改进的二进制狼群演化算法, 扩展了狼群演化算法的应用范围; Srikanth 等^[6]在深入分析狼群演化算法的基础上, 针对组合调度优化问题, 提出了二进制量子狼群演化算法 (quantum wolf pack evolutionary algorithm, QWPEA)。QWPEA 算法相对于 WPEA 算法具有原理简单、参数设置少、收敛速度快、较强的全局搜索能力等特点, 在测试函数和实际工程应用中取得了突出的效果^[7-8]。但这些应用都是以固定的搜索空间和种群个体位置的静态更新为基础进行优化求解, 对于离散空间中局部和全局的并行演化以及狼群中的个体狼的量子位置旋转角的动态选取和调整问题, QWPEA 算法并没有一个有效的解决方法。

元胞自动机 (cellar automation, CA) 最早由冯诺伊曼在 1966 年提出^[9], CA 是一种根据简单的并行演化规则和协同更新方法, 采用元胞来模拟复杂的离散系统动力学方法^[10], CA 具有时空动态性、状态离散性和同步性等基本特征。

为了解决有限的离散空间内狼群演化算法的收敛速度慢、搜索能力弱以及易于陷入局部最优等问题, 提出了一种求解离散优化问题的元胞量子狼群演化算法 (cellar quantum-behaved wolf pack evolutionary algorithm, CQWPEA), 该算法主要采用二进制编码方式和元胞自动机中的演化规则, 分别实现量子狼群中的个体狼位置与猎物之间的距离以及量子旋转角的选取和调整, 并给出了头狼与猎物资源位置的编码方式, 同时给出了探狼和猛狼局部搜索算子的编码方式, 分析了编码规则。另外, 采用泛函分析方法对该算法的收敛性进行了证明, 最后通过 6 个测试函数验证了 CQWPEA 算法的有效性和合理性。

1 量子狼群演化算法

为了使量子狼群算法适应离散搜索空间寻优

问题, 受二进制编码量子粒子群演化算法^[11]的启示, 在量子狼群演化算法中引入二进制编码, 融合元胞自动机演化规则, 提出了求解离散优化问题的元胞量子行为狼群演化算法。为分析方便, 首先对元胞自动机原理、二进制量子狼群演化算法中的狼群行为规则、头狼产生规则、狼群更新和变异等演化方程进行描述。

1.1 元胞自动机原理

元胞自动机通过利用大量元胞的并行演化规则来模拟复杂结构和过程, 成为探索复杂系统的有效工具^[12]。CA 协同更新元胞空间中的所有元胞, 协同更新的规则: 下一个时刻元胞 i 的状态是由自身和邻居在前一时刻的状态来决定的。元胞状态集、邻居、元胞空间以及局部演化规则作为 CA 的基本组成要素, 其邻接类型主要有两种, 即 Von-Neumann 型和 Moore 型, 如图 1 所示。

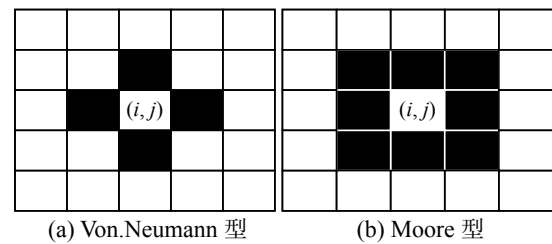


图 1 CA 的邻居类型

Fig. 1 The neighbor types of CA

元胞自动机的动态演化规则以元胞的动态时间状态变化集合为基础, 其可表示为

$$F: S_t^Z \rightarrow S_{t+1}^Z \quad (1)$$

式中: S 表示元胞状态的有限集合; Z 表示取整数集的一维空间; t 表示时间值。

元胞的局部演化函数 f 对于元胞的动态演化规则具有决定性作用, 在取整数的一维空间中, 元胞与其邻居可用 S^{2r+1} 来表示, 元胞的局部演化函数 f 可表示为

$$f: S_t^{2r+1} \rightarrow S_{t+1} \quad (2)$$

元胞局部演化函数 f 都是在有限的元胞状态集上输入和输出的, 将局部演化函数与元胞空间内的每个元胞进行组合优化, 这样就形成了一个全局演化规则:

$$F(c_{t+1}^i) = f(c_t^{i-r}, c_t^{i-r+1}, \dots, c_t^i, \dots, c_t^{i+r}) \quad (3)$$

式中: c_t^i 表示在位置 i 处的元胞, r 表示邻居半径。

CA 由一个标准的四元组构成, 其形式为

$$A = (H, D, S, M, f) \quad (4)$$

式中: A 为一个 CA; H 和 D 为一个元胞空间和维度数; S 为元胞的离散状态有限集; M 为所有元胞 (邻域和中心元胞) 的组合, 其可表示为

$$N = (S_1, S_2, \dots, S_n) \quad (5)$$

式中: n 表示一个元胞空间内的元胞邻居数; S_i 是一个取值为1~ n 的整数集; f 表示局部演化函数在 $S^n \rightarrow S$ 的映射过程; 确定元胞空间中的所有元胞位置采用 Z^d 整数矩阵。

1.2 量子狼群算法

1.2.1 量子狼群编码

在量子狼群演化算法中, 人工狼编码是以一组量子位和二进制表示, 每个量子位的状态可用式表(6)示:

$$|\mu\rangle = \alpha|0\rangle + \beta|1\rangle \quad (6)$$

式中 α 与 β 表示量子位对应状态的概率幅。

人工狼当前位置的编码可通过量子位的概率幅表示, 考虑到狼群初始化时编码的随机性, 假设采用量子方式编码的个体为 p_i , 则编码方式为

$$p = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_d \\ \beta_1 & \beta_2 & \cdots & \beta_d \end{bmatrix} \quad (7)$$

式中: $|\alpha|^2$ 表示量子态被观测为 $|0\rangle$ 态概率, $|\beta|^2$ 为量子态被观测为 $|1\rangle$ 态概率, 满足 $|\alpha_i|^2 + |\beta_i|^2 = 1$; $i = 1, 2, d$; d 为编码位数。

由此可知, 量子狼群可表示为 $Q(q) = (q_1^g, q_2^g, \dots, q_n^g)$, 其中, g 表示进化代数, n 表示个体狼数量; $q_i^g (i = 1, 2, \dots, n)$ 表示第 g 代狼群中第 i 只狼, 即

$$q_i^g = \begin{bmatrix} \alpha_{i1}^g & \alpha_{i2}^g & \cdots & \alpha_{id}^g \\ \beta_{i1}^g & \beta_{i2}^g & \cdots & \beta_{id}^g \end{bmatrix} \quad (8)$$

最后通过对 $Q(q)$ 进行量子测量, 获得一组确定的解 $p(g) = (x_1^g, x_2^g, \dots, x_n^g)$, 其中 $x_i^g (i = 1, 2, \dots, n)$ 为一匹狼通过量子测量到确定状态的二进制序列, 若该二进制序列的位数为 d , 那么 x_i^g 可以表示为 $x_i^g = (x_{i1}^g, x_{i2}^g, \dots, x_{id}^g)$ 。

1.2.2 双策略的初始量子位生成过程

在初始化狼群演化过程中, 狼群中每匹狼位置的所有量子位对应态的概率幅可采用 Logistic 混沌映射产生, 其产生的基本过程为:

- 1) 设狼群规模为 n , 个体狼位置编码的长度为 d ;
- 2) 随机产生 d 个混沌变量初始值 $x_i^j (i = 1, 2, \dots, n; j = 1, 2, \dots, d)$;
- 3) 对式(9)采用迭代计算获得相应的混沌变量序列 $x_k^j (k = 1, 2, \dots, n; j = 1, 2, \dots, d)$, 用这 n 个混沌变量来初始化狼群:

$$x_{k+1}^j = \mu x_k^j (1 - x_k^j), \quad k = 0, 1, \dots, n \quad (9)$$

式中: μ 为混沌因子, $0 \leq \mu \leq 4$; k 为迭代次数。当 $\mu = 4$, $0 \leq x_k^j \leq 1$ 时, Logistic 完全处于混沌状态。

4) 采用2种策略将式(8)中的 α_{id}^g 、 β_{id}^g 分别初始化为 $\cos(2x_k^j\pi)$ 、 $\sin(2x_k^j\pi)$ 和 x_i^j 、 $\sqrt{1 - (x_k^j)^2}$, 由此获得 2^n 个全部个体;

5) 计算全部 2^n 个个体的适应度值并进行排

序, 选取 n 个适应度值高的个体狼构成初始狼群。

1.2.3 头狼产生规则

在初始解空间中, 将最接近猎物资源(或最优目标函数)的个体狼视为头狼, 头狼直接进入迭代过程。算法运行中, 通过比较每匹人工狼的量子位状态 s_j , 获得当前狼群迭代过程中的最优个体狼作为头狼, 最终求解得到头狼的位置和最佳适应度。第 j 个位上的量子位状态可表示为

$$s_j = \begin{cases} 0, & \text{random}[0, 1] > |\alpha_j|^2 \\ 1, & \text{random}[0, 1] \leq |\alpha_j|^2 \end{cases} \quad (10)$$

式中: (s_1, s_2, \dots, s_j) 表示量子状态位对应于人工狼的二进制表示形式; $\text{random}[0, 1]$ 表示在 $[0, 1]$ 之间的随机数, $j = 1, 2, \dots, d$ 。

在传统的进化算法中, 与狼群算法的“优胜劣汰生存”规则和遗传算法的“轮盘赌”规则不同的是, 本文设计了一种基于滑模原理的交叉量子位遗传演化方法, 用于将选择之后产生的候选头狼集合 Z^{od} 中的个体头狼按优秀程度降序排序, 然后从染色体右端低量子位开始与头狼染色体按照滑模方式交叉, 式(11)和式(12)表示交叉参数 ω_i 呈高斯分布, 随着候选个体头狼逐渐陷入局部解, 如图2所示, 交叉点滑模按照式(13)向左移动, 用于分别与新一代头狼迭代, 产生更优秀后代头狼。

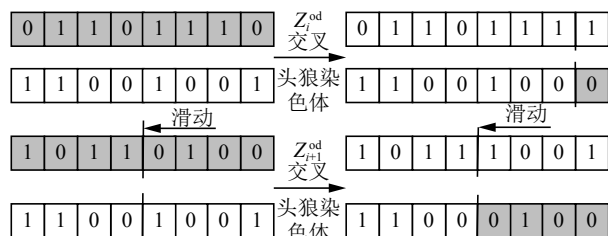


图2 候选头狼染色体量子位滑模交叉方法

Fig. 2 The sliding mode crossover of quantum bits of candidate lead wolf

交叉权值分布函数为

$$f_{gs}(e_i) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(e_i - \mu)^2}{2\sigma^2}\right] \quad (11)$$

式中: $Z_i^{\text{od}} = (Z_{i,1}^{\text{od}}, Z_{i,2}^{\text{od}}, \dots, Z_{i,j}^{\text{od}})$, $j \in L$ 表示第 i 匹候选头狼染色体量子位 j 从最优至次优串排列; $i = 1, 2, \dots, N$; $j = 1, 2, \dots, d$ 。

交叉权值为

$$\begin{cases} \omega_i = \int_{\frac{i-1}{I}\sigma}^{\frac{i+1}{I}\sigma} f_{gs}(e_i) de_i, & i \in I \\ \sum_{i=1}^I \omega_i = 1 \end{cases} \quad (12)$$

式中 I 表示当前候选头狼数量。

滑模位置为

$$j_i^s = \begin{cases} L, & \omega_i \in [0, I^{-1}) \\ \lfloor L \times (1 - \omega_i) \rfloor, & \omega_i \in (I^{-1}, (I-2) \times I^{-1}) \\ 2, & \omega_i \in ((I-2) \times I^{-1}, 1) \end{cases} \quad (13)$$

式中: j_i^s 表示滑模交叉结束量子位。可见, 候选头狼为最优解时 ω_i 参数最小, 滑模位置交叉得到的新解空间越邻近现有值; 反之, ω_i 参数最大, 滑模位置交叉得到的新解空间变化越大。

1.2.4 狼群位置更新

在 QWPEA 中, 当滑模交叉结束后, 采用上次迭代过程中的最优解对狼群中的每一个量子位进行量子旋转门更新, 更新过程如式 (14)。

$$\begin{bmatrix} \cos(\theta_{ij}^{t+1}) \\ \sin(\theta_{ij}^{t+1}) \end{bmatrix} = \begin{bmatrix} \cos(\Delta\theta_{ij}^{t+1}) & -\sin(\Delta\theta_{ij}^{t+1}) \\ \sin(\Delta\theta_{ij}^{t+1}) & \cos(\Delta\theta_{ij}^{t+1}) \end{bmatrix} \times \begin{bmatrix} \cos(\theta_{ij}^t) \\ \sin(\theta_{ij}^t) \end{bmatrix} \quad (14)$$

式中: $\cos(\theta_{ij}^{t+1})$ 和 $\sin(\theta_{ij}^{t+1})$ 表示第 $t+1$ 次迭代的第 j 个量子位的概率幅; $\Delta\theta_{ij}^{t+1}$ 表示第 j 个量子位的旋转角度, $i = 1, 2, \dots, N$, $j = 1, 2, \dots, d$, 其大小和方向是由量子人工狼群的行为规则确定的。

由此可知, 量子旋转门是通过改变描述量子人工狼位置的相位角来实现人工狼在搜索空间位置的同步移动。

1.2.5 量子狼群变异

为了避免算法陷入局部最优解状态, 维持狼群的多样性, 以平衡 d 维猎场空间内随机分布的人工狼和决策变量可行域为基础, 实施智能猎杀行为后, 基于优胜劣汰的生存法则, 会有 R 匹人工狼被淘汰, 并会有新的 R 匹人工狼存活下来, 但存活与淘汰的人工狼数量要相等, 这样既可维持狼群规模数量, 也可避免算法的过早收敛和全局搜索能力差的问题。因此, 狼群中个体狼的变异过程采用量子非门实现, 其表达式为

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos(\theta_{ij}) \\ \sin(\theta_{ij}) \end{bmatrix} = \begin{bmatrix} \cos(\frac{\pi}{2} - \theta_{ij}) \\ \sin(\frac{\pi}{2} - \theta_{ij}) \end{bmatrix} \quad (15)$$

式中: θ_{ij} 表示量子旋转角; $i = 1, 2, \dots, N$; $j = 1, 2, \dots, d$ 。

假设变异概率为 p_m , 每个人工狼在 $(0, 1)$ 之间给定一个随机数 random , 如果 $\text{random} < p_m$, 则随机选择若干个量子比特, 用量子非门交换两个概率幅, 而其旋转角度向量保持不变。

1.2.6 量子人工狼群行为描述

1) 四处游走行为

假设量子人工探狼的当前状态为 p_i , 在其感知的目标猎物资源信息为 $\text{fit}_i = f(x_i)$ 的范围内随机选择一个位置状态 p_j , 如果 $\text{fit}_i < \text{fit}_j$, 这时探狼 i 代替头狼发起召唤行为; 如果 $\text{fit}_i > \text{fit}_j$, 则探狼 i 向 P 个方

向按照游走步长 step_s^d 前进一步继续侦察; 如果此时探狼 i 感知的猎物气味浓度为 fit_i^p , $p \in H, H = \{1, 2, \dots, h\}$, 则自主决策后沿着猎物留下的气味最浓且大于当前位置 p_i 的方向 p^* 前移一步, 同时对探狼 i 的位置 p_i 进行更新。反复执行上述行为, 直到 $\text{fit}_i > \text{fit}_j$, 或者游走次数 T 达到最大游走的次数 T_{\max} 。其中选择的方向 p^* 应满足式 (16):

$$\begin{cases} p^* \in \max\{\text{fit}_i^p, p \in H\} \\ \text{fit}_i^p > \text{fit}_{i_0} \end{cases} \quad (16)$$

在人工狼群中存在的个体探狼有着不同差异, 嗅探猎物资源的方式也不同, 因此可取不同的 h 值, h 可取 $[h_{\min}, h_{\max}]$ 之间的随机整数。而在 d 维空间中, 探狼 i 沿着 p ($p = 1, 2, \dots, h$) 个游走方向前移一步后所处的空间位置为

$$x_{id}^p = x_{id} + \gamma \cdot \sin\left(2\pi \times \frac{p}{h}\right) \times \text{step}_s^d \quad (17)$$

式中: p 表示判断游走的方向数; γ 表示在 $[-1, 1]$ 间均匀分布的随机数; step_s^d 为第 d 维的游走步长; x_{id}^p 表示探狼的位置。

2) 嚎叫召唤行为

假设量子人工头狼的当前状态为 p_i , 头狼采用嚎叫召唤行为召集周围的 M_{num} 匹猛狼向其位置集合, 其中 $M_{\text{num}} = N - S_{\text{num}} - 1$; 接到头狼召唤的猛狼都以较大的奔袭步长 step_b^d 快速逼近头狼所在的位置 p_d , 即在第 d 维空间, 猛狼 j 经历第 $k+1$ 次迭代的位置为

$$x_{jd}^{k+1} = x_{jd}^k + \text{step}_b^d \cdot \left((p_d^k - x_{jd}^k) / |p_d^k - x_{jd}^k| \right) \quad (18)$$

式中: p_d^k 表示第 k 代狼群体中头狼的 d 维空间位置; x_{jd}^k 表示猛狼 j 的当前位置; $\text{step}_b^d \cdot \left((p_d^k - x_{jd}^k) / |p_d^k - x_{jd}^k| \right)$ 表示猛狼向头狼的聚集趋势。

猛狼在向头狼聚拢的过程中, 如果猛狼 j 感知的猎物气味浓度 $\text{fit}_i > \text{fit}_j$, 则令 $\text{fit}_i = \text{fit}_j$, 此时猛狼 j 转换为头狼并发起召唤行为; 如果 $\text{fit}_i < \text{fit}_j$, 则猛狼 j 继续快速奔袭, 且与头狼 fit_j 间的距离 $d_{is} < d_{\text{near}}$ 时, 即转入围攻。则判定距离 d_{near} 可表示为

$$d_{\text{near}} = \frac{1}{D\omega} \cdot \sum_{d=1}^D |\max_d - \min_d| \quad (19)$$

式中: ω 为距离判定因子; D 为空间维数; \max_d 、 \min_d 分别为待寻优的第 d 维空间变量的最大值和最小值。

3) 智能猎杀行为

假设量子人工头狼的当前状态为 p_i , 将距离猎物资源最近的头狼所在位置 p_d 看作猎物的位置。 fit_{id}^k 可视为在第 k 代狼群中在第 d 维空间中猎物的位置信息, 狼群的智能猎杀行为表达为

$$x_{id}^{k+1} = x_{id}^k + \gamma \cdot \text{step}_i^d \cdot \left(\text{fit}_{id}^k - x_{id}^k \right) \quad (20)$$

式中: $\gamma \in [-1, 1]$ 为均匀分布的随机数; step_i^d 表示人

工狼 i (猛狼和探狼)在 d 维空间中智能步长数。

假设在第 d 个变量的取值范围为 $[\max_d, \min_d]$, 则从狼群算法中抽象出的3种智能行为所涉及的游走步长 step_s^d 、奔袭步长 step_b^d 和猎杀步长 step_t^d 之间的关系为

$$\text{step}_s^d = 2 \cdot \text{step}_t^d = \text{step}_b^d / 2 = |\max_d, \min_d| / S \quad (21)$$

式中: S 表示步长因子, S 值越小人工探狼搜索越精细。

2 元胞量子狼群演化算法

在CQWPEA算法中, 人工狼(头狼、探狼和猛狼)的位置表示为一组0、1构成的二进制编码向量, 首先通过使用概率统计方法对个体狼与猎物之间的最优平均位置值进行编码, 并记录编码位中出现的0、1次数, 实现局部搜索算子编码位的计算; 然后, 采用元胞自动机选取和调整人工狼群位置的量子位旋转角, 增强元胞量子狼群演化算法的搜索空间, 加快算法的收敛速度。

CQWPEA的设计思路是, 将散布在猎场空间中的量子狼群中的探狼朝着猎物遗留的气味浓度强和环境信息方向进行嗅探, 一旦发现猎物, 探狼需要判断自身与头狼距离猎物资源的位置, 如果探狼与猎物的量子位置旋转角比头狼与猎物的量子位置旋转角大, 则探狼代替头狼发起嚎叫召唤行为, 否则向头狼报告猎物的位置信息; 猛狼收到嚎叫召唤信号后, 猛狼迅速向头狼所在位置或猎物所在位置靠拢, 同时对猎场的周围环境进行探测, 量子狼群进化算法中的人工狼之间以嚎叫信息实现相互通信, 并通过元胞机中的局部演化规则不断调整人工狼的位置旋转角度, 更好地调节人工狼群的搜索范围和定位猎物的位置, 随着局部搜索过程的不断推进, 人工狼群向着全局最优解逼近, 最终通过计算头狼与猎物的平均最优位置来实现目标函数的优化求解。

2.1 个体狼位置的二进制编码

在二进制编码的元胞量子狼群演化算法中, 为了精确描述狼群中头狼与猎物资源之间的距离关系, 将元胞空间作为量子狼群演化算法的搜索空间, 将距离猎物资源最近的头狼所在位置 p_d 看作猎物资源(目标函数)的位置, 使用量子狼群进化算法中的探狼、猛狼搜索到的局部最优解和猎杀攻击后的全局最优解分别作为元胞空间内一个元胞, 并采用扩展 Moore 邻居类型, 采用数学语言对元胞量子狼群演化算法进行形式化描述。

定义1 设以 $N \times m$ 维欧式空间作为量子狼群

演化算法的搜索空间, 人工狼的位置为

$$X_i = \{x_{i1}, x_{i2}, \dots, x_{im} | i \in \{1, 2, \dots, N\}; j \in \{1, 2, \dots, m\}\} \quad (22)$$

式中: N 表示人工狼总数, m 表示编码长度; x_{ij} 表示人工狼的第 X_i 个位置的第 j 个编码位置, 通过反置赋值操作且只能取0、1; 头狼 p 和猎物资源 q 之间的曼哈顿距离可表示为

$$L(p, q) = \sum_{j=1}^m |x_{pj} - x_{qj}|, \quad p, q \in \{1, 2, \dots, N\} \quad (23)$$

定义2 移动算子^[5], 设人工狼 i 的位置为 $X_i = \{x_{i1}, x_{i2}, \dots, x_{ij}, \dots, x_{im}\}$; M 表示非空的反置编码位, 即表达了人工狼的位置; r 是非空的反置编码位数, 即游走步长; 运动算子 $\theta(X_i, M, r)$ 表示在人工狼 i 的位置 X_i 中, 从 M 个编码位中随机选择 r 个编码位并对其进行反置操作。

定义3 设集合 $C = (c_1, c_2, \dots, c_i, \dots, c_n), c_i \in \{0, 1\}$, c_i 的排列组合构成元胞空间, 其具体形式为

$$L = \{\text{Cell}X = (c_1, c_2, \dots, c_i, \dots, c_n) | c_i \in \{0, 1\}\} \quad (24)$$

式中 CellX 表示由 c_i 所组合的元胞。

定义4^[7] 扩展 Moore 邻居类型

$$N_{\text{moore}} = \{\text{Cell}Y | \text{diff}(\text{cell}Y - \text{cell}X) \leq r, \text{cell}X, \text{cell}Y \in L\} \quad (25)$$

式中: 任意两个 c_i, c_{i+1} 的排序组合的差异度为 $\text{diff}(\text{cell}Y - \text{cell}X) \leq r$, r 为差异度, 本文取 $r = 2$ 。

在二进制编码的元胞量子狼群演化算法中, 为了精确表达头狼和猎物资源之间的距离, 根据定义1, 假设头狼和猎物的位置为 (X_1, X_2) , 它们分别有两个决策变量 (X_{11}, X_{12}) 、 (X_{21}, X_{22}) , 采用6位二进制编码对每个决策变量进行表示, 如图3所示。

X_{11}					X_1			X_{12}		
1	1	0	1	0	1	1	0	0	1	0
X_{21}					X_2			X_{22}		
0	1	0	1	1	0	1	1	0	1	1

图3 头狼与猎物位置的二进制编码

Fig. 3 Location of the lead wolf and prey with binary encoding

在CQWPEA算法中, 定义头狼与猎物(目标函数)含有决策变量的个数即为元胞空间的维数; 例如, X_{id} 表示狼群中第 i 匹头狼的 d 个决策变量, X_i, X_{id} 的二进制编码长度分别用 l 和 l_d 表示, 则

$$l = \sum_{i=1}^d l_i, \quad d = 1, 2, \dots, D \quad (26)$$

根据量子狼群演化算法, 通过修改算法中的头狼和猎物之间的平均最优位置 m_{best} 的值, 生成CQWPEA中的 m_{best} , 并用二进制位串表示种群中全部最优个体狼位置, 通过概率统计方法^[6], 记录二进制编码位中0、1出现的概率次数, 如果0出

现的次数多, 则 m_{best} 对应的值为 0, 否则为 1, 如图 4 所示。

$P_{\text{best}1}$	1	1	0	1	0	1	1	0	0	0	1	0
$P_{\text{best}2}$	0	1	0	1	1	0	1	1	0	1	1	1
$P_{\text{best}3}$	1	0	0	0	1	1	1	0	1	0	0	1
$P_{\text{best}4}$	0	0	1	0	1	1	0	1	0	1	0	0
$P_{\text{best}5}$	1	0	1	0	1	0	0	1	1	0	0	1
m_{best}	1	0	0	0	1	1	1	1	0	0	0	1

图 4 人工狼与猎物之间的平均最优位置值

Fig. 4 The optimal average position between the artificial wolf and prey

从图 4 中可知, 狼群中有 5 个最优位置值, 每个个体狼当前的最优个体信息分别为 $P_{\text{best}1}$ 、 $P_{\text{best}2}$ 、 $P_{\text{best}3}$ 、 $P_{\text{best}4}$ 、 $P_{\text{best}5}$, 其第 1 列对应的二进制位值为 1、0、1、0、1, 依据概率统计方法, 则 m_{best} 对应的第 1 列二进制值为 1, 以此类推。由此获得的 m_{best} 的值为 1000011110001。特别地, 如果对应的每一列中的二进制位数中出现相同的 0 或 1, 则 m_{best} 随机选择 0 或 1。由此可构造出 CQWPEA 算法中 m_{best} 的函数值。

在 QWPEA 算法中, 式 (17)、(18) 是探狼和猛狼在整个搜索空间的局部位置信息, 其值表示了局部搜索算子 p_{id} , $p_{id} \in (p_{\text{best}i}, g_{\text{best}})$, $p_{id} = \{p_{i1}, p_{i2}, \dots, p_{iD}\}$ 位于 $(p_{\text{best}i}, g_{\text{best}})$ 之间的对角线两端的超矩形中, p_{id} 到 $p_{\text{best}i}, g_{\text{best}}$ 的距离需小于对角线的长度, 即

$$|p_{id} - p_{\text{best}i}| \leq |p_{\text{best}i} - g_{\text{best}}| \quad (27)$$

$$|p_{id} - g_{\text{best}}| \leq |p_{\text{best}i} - g_{\text{best}}| \quad (28)$$

通过对 p_{id} 值的计算, 可使种群产生多样性, 并可使算法跳出局部搜索空间。在 QWPEA 算法中, 局部搜索算子 p_{id} 的产生主要取决于上一代种群的 $p_{\text{best}i}$ 和 g_{best} 中的每一个量子位的随机交叉, 形成下一代的搜索算子 p_{id} , 显然 p_{id} 满足定义 1 的曼哈顿距离, 如图 5 所示。

$P_{\text{best}i}$	0	0	1	0	0	1	0	1	1	1	0	0
g_{best}	1	0	1	0	0	0	0	1	0	0	0	1
P_i	1	0	1	0	0	1	0	1	1	1	0	1

图 5 人工狼的多点交叉算子

Fig. 5 Multi-point crossover operator of wolf

探狼和猛狼的局部搜索算子, p_i^d 、 p_j^d 的每一位编码可由式 (29) 与 (30) 计算获得:

$$p_{id} = p_{\text{best}i} \gamma x_{id}^p + g_{\text{best}} \gamma x_{id}^p \quad (29)$$

$$p_{jd} = p_{\text{best}i} x_{jd}^{k+1} + g_{\text{best}} x_{jd}^{k+1} \quad (30)$$

式中 $\gamma \in [-1, 1]$ 内的随机数。由此可构造出直接计算 p_{id} 与 p_{jd} 值的函数^[6]。

2.2 基于 CA 的量子旋转角选取策略

在 QWPEA 算法中, 新一代个体狼的概率幅是由上一代个体的概率幅与量子旋转门 $U(\theta)$ 更新计算获得, 更新过程如式 (31):

$$U(\theta) = \begin{bmatrix} \cos(\Delta\theta_{ij}^{t+1}) & -\sin(\Delta\theta_{ij}^{t+1}) \\ \sin(\Delta\theta_{ij}^{t+1}) & \cos(\Delta\theta_{ij}^{t+1}) \end{bmatrix} \quad (31)$$

$$\begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} = U(\theta) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (32)$$

在 QWPEA 中, 旋转变异的量子旋转角 θ 可通过查表获得^[11]。但查表操作需要多次比较, 运算耗费时间长。因此, 采用元胞自动机演化规则对量子旋转角进行快速调整, 将量子狼群视为 CA 模型, 每匹狼视为 1 个元胞个体, 并将其放入二维空间区域内^[12], 然后采用元胞邻居结构, 对元胞邻域范围内的每匹狼的最优解代替自身最优解。

设每匹狼 i 的 $n-1$ 个邻居构成的集合为 $S_i(t) = \{x_1(t), x_2(t), \dots, x_n(t)\}$ 。在解算个体狼的邻居集合最优解时引入以下演化规则^[13]:

$$\begin{cases} S^t = 1, & \text{则 } S^{t+1} = \begin{cases} 1, & s = 2, 3 \\ 0, & s \neq 2, 3 \end{cases} \\ S^t = 0, & \text{则 } S^{t+1} = \begin{cases} 1, & s = 3 \\ 0, & s \neq 3 \end{cases} \end{cases} \quad (33)$$

式中: S^t 与 S^{t+1} 分别表示 t 与 $t+1$ 时刻元胞状态; s 表示元胞邻居集合中状态为 1 的元胞个数。设第 g 代狼群当前解集 $X_i^g = (x_{i1}^g, x_{i2}^g, \dots, x_{id}^g)$ ($i = 1, 2, \dots, n$), 第 i 匹狼的局部最优解为 $P_i^g = (p_{i1}^g, p_{i2}^g, \dots, p_{id}^g)$ ($i = 1, 2, \dots, n$), 狼群的全局最优解为 $P_g^g = (p_{g1}^g, p_{g2}^g, \dots, p_{gd}^g)$, 则量子门旋转角的更新策略如式 (34) 所示:

$$\Delta\theta_{ij}^{g+1} = \Delta\theta(c_1(p_{ij}^g - x_{ij}^g) + c_2(p_{gj}^g - x_{ij}^g)) \quad (34)$$

式中: g 、 $g+1$ 表示迭代次数; p_{ij}^g 表示第 i 匹人工狼第 j 位在第 g 代邻居集合的局部最优解; p_{gi}^g 表示整体狼群第 j 位在第 g 代的全局最优解; x_{ij}^g 表示第 i 匹人工狼第 j 位在第 g 代的当前解; c_1 和 c_2 表示搜索速度系数, 通常取 $c_1 = 1, c_2 = 2$; $\Delta\theta$ 表示量子旋转角, 该角度根据式 (35) 计算, 可从 $0.01\pi \sim 0.04\pi$ 动态减小。

$$\Delta\theta = 0.04\pi - \frac{(0.04\pi - 0.01\pi)T}{T_{\max}} \quad (35)$$

式中 T 与 T_{\max} 分别表示当前迭代次数和最大进化迭代次数。

为了增强量子狼群演化算法搜索范围, 将量子旋转角进一步扩展到 $[-\lambda, \lambda]$, 使狼群遍历范围呈现双向性。另外, 由于量子态经测量后获得二进制编码, 因此按式 (36) 进行取值:

$$\Delta\theta_{ij}^{g+1} = \{0, \pm\Delta\theta, \pm2\Delta\theta, 3\Delta\theta\} \quad (36)$$

由式(36)可知,本文算法中的量子旋转角扩大了3倍,呈现双向性,避免了混沌序列的迭代计算和多次比较的查表操作,节省了运算时间。

2.3 CQWPEA 算法的步骤

1) 初始化参数,根据1.2.2节的双策略对量子狼群的初始量子位进行初始化,并采用二进制序串的形式初始化狼群中的个体狼位置 x_{id}^k ,使 $p_{best} = x_{id}^k$,计算初始狼群个体的适应度值,同时对算法的基本参数进行初始化操作。

2) 根据 m_{best} 函数计算人工狼与猎物资源之间的平均最优位置值 m_{best} 。

3) 根据方程式(29)、(30)计算局部搜索算子 p_{id}, p_{jd} 的值。

4) 根据初始狼群的适应度函数值计算种群中每个个体狼位置值,并与上一次迭代的局部最优值进行比较,如果适应度值 $fit(X_i) < f(p_{besti})$,则用 $p_{besti} = X_i$ 更新局部最优位置;反之,则不更新;转入步骤5)。

5) 量子人工狼按照式(31)~(36)进行元胞邻域搜索,并计算狼群体中新的适应度值。将狼群的全局最优位置值 g_{bestid} 与上一次的全局最优位置值 $g_{besti-1}$ 进行比较,如果 $f(g_{bestid}) < f(g_{besti-1})$,则替换原最优位置值,反之,不更新。另外在量子旋转角中,如果 $\Delta\theta = 0$,对狼群的量子编码位进行交叉,交换量子位概率幅 α 与 β 。

6) 按头狼产生规则和猎物气息元胞邻居空间的演化规则对头狼位置 p_d 进行更新,邻域搜索同4)。

7) 将局部搜索算子 p_{id}, p_{jd} 的值转换为全局最优位置值。

8) 判断是否满足迭代次数 k_{max} 要求,如果满足,则输出头狼的位置 p_d ,即所求解问题的最优解,否则转入2)。

9) 输出目前的最优解集和最优值。

3 CQWPEA 算法的收敛性分析

本节将用泛函分析方法^[13-14]对CQWPEA算法的收敛性进行分析证明。令二进制编码字符集 $k=\{0,1\}$,采用长度为 D 的二进制位串表示头狼位置编码,编码空间 $S=\{0,1\}^D, |S|=2^D$ 。假设狼群规模为 M ,狼群单钱位置集合为 X ,头狼最佳位置集合为 P ,则狼群的当前位置空间可表示为

$$S^X = \{(X_1, X_2, \dots, X_M), X_j \in S, j=1, 2, \dots, M\} \quad (37)$$

狼群中人工狼的最佳位置空间为

$$S^P = \{(P_1, P_2, \dots, P_M), P_j \in S, j=1, 2, \dots, M\} \quad (38)$$

全局最优解的位置空间为

$$S^g = \{P_g, P_g \in S\} = S \quad (39)$$

根据QWPEA算法的流程图^[6]可知, QWPEA算法的解算过程遵循着反复迭代,逐渐求精的过程,该过程中包括目标函数适应度值的评价、 p_{best} 位置和 g_{best} 位置的选择,以及人工狼的位置更新,从而使狼群达到更优状态。因此,可用随机映射运算过程来抽象表示QWPEA算法的流程。

设 (Ω, A, P) 是狼群搜索的概率空间,可以给出CQWPEA搜索算子的定义,见定义5。

定义5 CQWPEA算法的搜索算子 T 是通过逐步迭代方式对个体狼的当前位置进行迭代搜索,以获得个体狼的最好位置的过程,该过程是从人工狼当前位置的元胞状态空间以及全局位置的元胞状态空间随机映射到个体狼最好位置状态空间,其映射形式为

$$T: \Omega \times S^X \times S^P \times S^g \rightarrow S^P \quad (40)$$

CQWPEA算法在每次迭代解算过程中产生一次映射 T_{CQWPEA} 。设 $X(t)$ 和 $P(t)$ 分别为第 t 次迭代过程中产生的当前位置的种群和个体最好位置种群, $X(t+1)$ 、 $P(t+1)$ 、 $P_g(t+1)$ 分别为第 $t+1$ 次迭代产生的人工狼群位置、个体狼位置以及全局最好位置,则

$$P(t+1) = T(\omega, X(t), P(t), P_g(t)) \quad (41)$$

式中: $t=0, 1, \dots, T-1$; T 为最大游走次数或迭代次数。

实验表明, CQWPEA算法中每一代全局最好解位置的适应度值序列是一个递增序列,即

$$f(P_{g,t-1}) \leq f(P_{g,t}) \leq f(P_{g,t+1}) \quad (42)$$

式中: $P_{g,t-1} \in P(t-1)$ 、 $P_{g,t} \in P(t)$ 、 $P_{g,t+1} \in P(t+1)$ 分别表示第 $t-1$ 、 t 、 $t+1$ 代的全局最好解位置。

在CQWPEA算法中,解算优化问题时,通常只需关注搜索过程中狼群攻击获得的最好解,不妨将迭代狼群体用全局最好位置来代替。由此对CQWPEA算法的映射过程进行重新定义,其形式为

$$P_{g,t+1} = T(\omega, P_{g,t}), \quad P_{g,t} \in P(t), \quad P_{g,t+1} \in P(t+1) \quad (43)$$

在CQWPEA算法中,解算优化问题的过程可视为元胞演化空间之间的映射。以下采用随机映射的方法证明和分析CQWPEA算法的收敛性。

定义6 定义度量 $d: S \times S \rightarrow R$, 其中 d 的表达式为

$$d(P_{g,i}, P_{g,j}) = |c - f(P_{g,i}) - (c - f(P_{g,j}))| = |f(P_{g,i}) - f(P_{g,j})| \quad (44)$$

式中: $c \rightarrow \infty$; $P_{g,i}, P_{g,j} \in S$; (S, d) 为完备可分的度量空间。

证明 首先证明 (S, d) 是度量空间。设 $S \neq \emptyset$, $d \in S \times S$ 的实值函数。对于 $P_{g,i}, P_{g,j}, P_{g,k} \in S$, 对应一个实数 $d(P_{g,i}, P_{g,j}) \in S$ 满足:

1) $d(P_{g,i}, P_{g,j}) \geq 0$, 当且仅当 $P_{g,i} = P_{g,j}$ 时, $d(P_{g,i}, P_{g,j}) = 0$, 满足非负性;

2) $d(P_{g,i}, P_{g,j}) = |c - f(P_{g,i}) - (c - f(P_{g,j}))| = |c - f(P_{g,j}) - (c - f(P_{g,i}))|$, 满足对称性;

3) $(P_{g,i}, P_{g,j}) = |c - f(P_{g,i}) - (c - f(P_{g,j}))| = |c - f(P_{g,i}) - (c - f(P_{g,k})) + (c - f(P_{g,k})) - (c - f(P_{g,j}))| \leq c - f(P_{g,i}) - (c - f(P_{g,k})) + |(c - f(P_{g,k})) - (c - f(P_{g,j}))|$, 满足三角不等式, 因此 (S, d) 为度量空间。

其次证明 (S, d) 为完备度量空间。设 S 是有限的二进制编码位串数, 对当前狼群中最好个体狼位置的柯西序列 $\{P_{g,i}, P_{g,i} \in S\}$, 以及 $\forall \varepsilon > 0, \exists N$, 当自然数 $n > N$ 时, $d(P_{g,n}, P_{g,m}) < \varepsilon$, 当 $n \rightarrow \infty$ 时, $P_{g,n} \rightarrow P_g$ 。因此 (S, d) 是完备的度量空间。

最后证明 (S, d) 是可分的。设 $G \subseteq S$, 因 S 为有限元集合, 所以 G 为可数子集。又因为 G 闭包于 S , 所以 G 在 S 中稠密, 由此得到 (S, d) 是可分的论证。故 (S, d) 是完备可分的度量空间, 证毕。

定义 7 随机搜索算子 $T: \Omega \times S \rightarrow S$ 称为随机压缩算子, 如果 $\exists K(\omega) < 1$ 的非负实值随机变量, 则 $d(\{(\omega, d(T(\omega, P_{g,i}), T(\omega, P_{g,i+1})))K(\omega)d(P_{g,i}, P_{g,i+1})\}) = 1$
 $P_{g,i}, P_{g,i+1} \in S$

定理 1 CQWPEA 算法的迭代过程形成的映射 T_{CQWPEA} 是随机压缩算子。

证明 依据 CQWPEA 原理, 每迭代一次均可产生比上一次更优的个体狼和目标猎物资源, 所以存在一个非负实值随机变量 $K(\omega) \in [0, 1)$, 使得

$$d(T_{CQWPEA}(\omega, P_{g,i-1}), T_{CQWPEA}(\omega, P_{g,i})) = d(P_{g,i}, P_{g,i+1}) = |(c - f(P_{g,i})) - (c - f(P_{g,i+1}))| \leq K(\omega)|(c - f(P_{g,i-1})) - (c - f(P_{g,i}))| = K(\omega)d(P_{g,i-1}, P_{g,i})$$

$$\Omega_0 = \{\omega : d(T_{CQWPEA}(\omega, P_{g,i-1}), T_{CQWPEA}(\omega, P_{g,i-1})) \leq K(\omega)d(P_{g,i-1}, P_{g,i})\} \leq p(\Omega_0) = 1$$

由此可知, CQWPEA 算法迭代过程中形成的映射 $T_{CQWPEA}: \Omega \times S^X \times S^P \times S^g \rightarrow S^P$ 是一个随机压缩算子。

定理 2 (随机压缩映射定理) 设随机搜索算子 $T: \Omega \times S \rightarrow S$ 满足所有的 $\omega \in S$, $T(\omega)$ 均为压缩算子, 即 $\exists \Omega_0 \in \Omega, p(\Omega_0) = 1$, 对 $\forall \omega \in \Omega_0$, 则 $d(T(\omega, P_{g,i}), T(\omega, P_{g,i-1})) \leq K(\omega)d(P_{g,i}, P_{g,i+1})$, 则 $P_{g,i}, P_{g,i-1}, P_{g,i+1} \in S$, $0 \leq K(\omega) < 1$, 则 $T(\omega)$ 有唯一随机不动点 $r(\omega)$, 即 $T(\omega, r(\omega)) = r(\omega)$ 。

证明 首先利用巴拿赫压不动点定理, 对 $\forall \omega \in \Omega_0, \exists h(\omega) \in S$ 作为 $T(\omega)$ 的唯一不动点, 对

$P_{g,i} \in S$, 令 $r(\omega) = \begin{cases} h(\omega), & \omega \in \Omega_0 \\ P_{g,i}, & \omega \in \Omega - \Omega_0 \end{cases}$ 为 $T(\omega)$ 的唯一不动点。

然后证明 $r(\omega)$ 的可测性。对 $\forall P_{g,0} \in S$, 令 $P_{g,1}(\omega) = T_{CQWPEA}(\omega, P_{g,0})$, $P_{g,i+1}(\omega) = T_{CQWPEA}(\omega, P_{g,i}), i = 1, 2, \dots$, 因为 $P_{g,1}(\omega) \rightarrow P_{g,0}$, $T_{CQWPEA}(\omega, P_{g,i}(\omega)) \rightarrow T_{CQWPEA}(\omega, P_{g,0}), P_{g,i} \in S$, 即 $T_{CQWPEA}(\omega)$ 连续。依据符合定理, $\{P_{g,i}\}$ 为一个随机变量序列, 再根据巴拿赫压不动点定理可知, $P_{g,i}(\omega) \rightarrow r(\omega)$, 根据极限随机变量定理的定义, $r(\omega)$ 作为一个随机变量, 因此 $r(\omega)$ 作为 $T_{CQWPEA}(\omega)$ 的随机不动点, 证毕。

4 实验仿真与结果分析

4.1 测试函数

为了检验 CQWPEA 算法的寻优性能, 选取 6 个标准的测试函数进行仿真实验, 6 个测试函数的表达式如下:

1) Sphere 函数

$$f_1(x) = \sum_{i=1}^d x_i^2, \quad -100 \leq x_i \leq 100$$

2) Schwefel 函数

$$f_2(x) = 418.9829 \times d - \sum_{i=1}^d x_i \cdot \sin(|x_i|^{\frac{1}{2}}), \quad -500 \leq x_i \leq 500$$

3) Rosenbrock 函数

$$f_3(x) = \sum_{i=1}^d \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right], \quad -30 \leq x_i \leq 30$$

4) Rastrigrin 函数

$$f_4(x) = \sum_{i=1}^d (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad -5.12 \leq x_i \leq 5.12$$

5) Ackley 函数

$$f_5(x) = -20 \exp \left(-\frac{1}{5} \sqrt{\frac{1}{d} \sum_{i=1}^d x_i^2} \right) - \exp \left(\frac{1}{d} \sum_{i=1}^d \cos(2\pi x_i) \right) + 20 + e, \quad -32 \leq x_i \leq 32$$

6) Griewangk 函数

$$f_6(x) = \frac{1}{4000} \sum_{i=1}^d x_i^2 - \prod_{i=1}^d \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1, \quad -100 \leq x_i \leq 100$$

在上述 6 个标准测试函数中, $f_1(x)$ 、 $f_2(x)$ 和 $f_3(x)$ 为单峰函数, $f_4(x)$ 、 $f_5(x)$ 和 $f_6(x)$ 为多峰函数, 函数的维数均设置为 30, 6 个标准测试函数的全局最优解均为 0。

4.2 仿真结果分析

采用 CQWPEA 算法对 6 个标准测试函数进行解算, 并将其结果与 WPEA、QWPEA 算法的结果进行比较。其中相同的参数设置为: 狼群规模均为 500, 最大迭代次数为 500 次, 收敛精度设置

为 0.000 01。具体到每个不同算法的参数,在 WPEA 算法和 QWPEA 算法中,判定因子 $\omega = 500$,步长因子 $S = 1\,000$,更新因子 $\beta = 6$,最大游走次数 $T_{\max} = 20$,探狼比例因子 $\delta = 4$,滑模交叉 $\mu = 0.2$, $\sigma = 0.3$, $I = 8$, $k_{\max} = 500$ 。在 CQWPEA 算法的控制参数 $\gamma = 0.6$,量子旋转角 $\Delta\theta = 0.05$,实验仿真环境:Windows7 系统,3 GB 内存,4 GHz CPU,算法基于 MATLAB2015a。每种算法的终止条件均为满足算法的寻优目标或达到最大迭代次数,计算结

果见表 1 所示。各取每个算例 20 次的实验数据,记录其最优值、平均值、最差值;并将 CQWPEA 算法的结果与 WPEA 和 QWPEA 算法进行比较,分别记录其平均迭代次数、最大迭代次数、收敛率以及 20 次独立运行消耗的总时间。为了增加算法的可信度,WPEA 和 QWPEA 算法的参数直接来源于参考文献,优化比较结果如表 2 所示。表 2 中的“—”表示对应的寻优成功次数指标无法获得统计结果。

表 1 6 个测试函数的结果比较
Table 1 Experimental results of six test functions

函数 维数	WPEA			QWPEA			CQWPEA		
	最优值	最差值	平均值	最优值	最差值	平均值	最优值	最差值	平均值
$f_1(x)$	10	0.150 7	2.969 1	1.494 8	0	0	0	0	0
	20	0.274 2	3.529 1	3.171 0	0	0	0	0	0
	30	0.126 2	8.845 4	4.009 1	0	0	0	0	0
$f_2(x)$	10	0.020 9	1.287 4	0.843 7	$1.273\,0 \times 10^{-3}$	$2.417\,8 \times 10^{-3}$	$1.276\,6 \times 10^{-3}$	$1.272\,8 \times 10^{-4}$	$1.272\,8 \times 10^{-4}$
	20	0.003 6	1.987 1	0.522 0	$2.546\,0 \times 10^{-3}$	$2.910\,2 \times 10^{-3}$	$2.556\,4 \times 10^{-3}$	$2.545\,5 \times 10^{-4}$	$2.545\,5 \times 10^{-4}$
	30	0.063 0	1.845 6	0.806 3	$3.819\,0 \times 10^{-3}$	$4.199\,6 \times 10^{-3}$	$3.830\,9 \times 10^{-3}$	$3.818\,3 \times 10^{-4}$	$3.818\,3 \times 10^{-4}$
$f_3(x)$	10	0.003 5	0.118 4	0.540 3	20.527 8	$832.290\,0 \times 10^4$	$1.153\,8 \times 10^3$	$9.922\,0 \times 10^{-30}$	$8.999\,9$
	20	0.028 0	0.325 2	0.137 3	128.120 7	$9.031\,5 \times 10^4$	$3.033\,8 \times 10^3$	$2.939\,0 \times 10^{-17}$	$8.996\,9$
	30	0.007 8	0.289 7	0.150 3	165.607 5	$9.016\,8 \times 10^4$	$3.807\,0 \times 10^3$	$2.923\,0 \times 10^{-16}$	$26.223\,4$
$f_4(x)$	10	0.003 0	0.029 1	0.009 7	9.880 0	33.827 4	22.380 0	0	0
	20	0.005 5	0.030 6	0.014 3	72.813 7	137.791 4	100.284 4	0	0
	30	0.013 0	0.065 5	0.030 8	144.402 3	222.547 0	192.987 0	0	0
$f_5(x)$	10	0.020 9	0.059 7	0.033 7	0.019 7	8.346 0	1.346 0	$8.882\,0 \times 10^{-16}$	$8.882\,0 \times 10^{-16}$
	20	0.024 7	0.054 0	0.032 9	0.024 0	7.554 1	2.006 0	$8.882\,0 \times 10^{-16}$	$8.882\,0 \times 10^{-16}$
	30	0.025 1	0.065 9	0.037 0	0.028 2	5.863 8	2.598 5	$8.882\,0 \times 10^{-16}$	$8.882\,0 \times 10^{-16}$
$f_6(x)$	10	0.003 1	0.313 3	0.107 2	0.277 9	0.412 8	0.096 4	0	0
	20	0.008 1	0.480 8	0.265 6	0.257 3	0.851 7	0.503 8	0	0
	30	0.002 0	0.325 3	0.163 9	0.363 7	0.994 9	0.781 6	0	0

从表 1 的结果可知,在满足固定收敛精度下,本文提出的 CQWPEA 算法分别在维度为 10、20 和 30 的基础上进行测试,发现除了 Schwefel 函数、Rosenbrock 函数和 Ackley 函数外,其他 3 种函数经过 20 次实验均能一致性收敛到问题的全局最优解 0。针对函数 $f_1(x)$,CQWPEA 算法和 QWPEA 算法均可获得最优解,而 WPEA 算法的寻优能力较差;针对函数 $f_2(x)$,3 种算法均无法获得最优解,但几乎可达到近似最优解;针对函数 $f_3(x)$,3 种算法虽然无法获得最优解,但 CQWPEA

算法要比其他 2 种算法获得的近似最优解更为接近最优解值 0;针对函数 $f_4(x)$,CQWPEA 算法的寻优能力最强,WPEA 算法的寻优能力要比 QWPEA 算法强,这主要是因为设置算法参数的不同而造成的;针对函数 $f_5(x)$,虽然 3 种算法均没有获得最优解值,但从总体上来看,CQWPEA 算法获得的最优解几乎接近全局最优解,其寻优能力较强,而 QWPEA 算法和 WPEA 算法获得的最优解距离全局最优解相差甚远;针对函数 $f_6(x)$,CQWPEA 算法可获得全局最优解,WPEA 算法的

寻优能力要比 QWPEA 算法强。

由上述分析结果可以得出,从总体上来看,本文提出的求解离散优化问题的元胞量子狼群演化算法在寻优能力均要优于其他狼群演化算法和量子狼群演化算法,但从部分函数测试的结果可

知,本文算法对于部分函数也会存在无法获得全局最优解问题,这主要是因为算法的初始参数设置的随机性、局部寻优结果差、元胞演化规则对量子旋转角的调整速度慢、量子相位角的选择不够精确等原因造成的。

表 2 迭代次数比较
Table 2 Comparison of iterations

函数	维数	算法	优化步数	平均迭代次数	最大迭代次数	收敛率/%	消耗总时间/s
$f_1(x)$	30	WPEA	500	—	—	0	31.545 960
		QWPEA	500	13.40	10	100	3.692 419
		CQWPEA	500	17.95	12	100	0.460 025
$f_2(x)$	10	WPEA	500	909	501	25	91.341 201
		QWPEA	500	28.5	11	80	3.238 189
		CQWPEA	500	7.20	2	100	0.822 512
$f_3(x)$	10	WPEA	500	950.20	502	10	77.387 860
		QWPEA	500	—	—	0	2.615 991
		CQWPEA	500	300.85	249	100	0.511 912
$f_4(x)$	30	WPEA	500	—	—	0	—
		QWPEA	500	—	—	0	5.197 518
		CQWPEA	500	18.75	16	100	0.724 987
$f_5(x)$	30	WPEA	500	—	—	0	33.743 127
		QWPEA	500	907.35	66	10	3.103 910
		CQWPEA	500	27.60	21	100	0.842 867
$f_6(x)$	30	WPEA	500	—	—	0	31.591 887
		QWPEA	500	—	—	0	3.250 854
		CQWPEA	500	21.40	10	100	0.545 058

由表 2 的结果可知,在不同的维数和相同的优化步数下,从收敛率和消耗时间上来看,对于 6 种不同的标准测试函数, CQWPEA 算法收敛率均可到达 100%,且消耗时间明显比 WPEA 算法和 QWPEA 算法少;但对于不同的函数,3 个算法的性能还存在一定的差异。对于 $f_1(x)$ 函数、 $f_4(x)$ 函数、 $f_5(x)$ 函数和 $f_6(x)$ 函数, WPEA 算法的寻优结果不理想,主要是因为当维数大于 3 时,这 4 个典型的凹函数呈现多峰特征,具有大量的局部极值点,因此找到全局最优解较为困难。对于 $f_3(x)$ 函数、 $f_4(x)$ 函数和 $f_6(x)$ 函数, QWPEA 算法的寻优结果的收敛率为 0,其搜索到的最优解与目标函数的最优解的偏差较大,这是因为这 3 个函数是复杂的非线性多峰函数,寻优过程中会使算法陷入局部收敛;对于 $f_2(x)$ 函数和 $f_3(x)$ 函数, QWPEA 算法的收敛率也几乎为 0,其搜索到的最优解偏差也较大。

由上述的比较结果知,与 WPEA 算法和 QWPEA 算法相比,无论从收敛精度还是收敛稳定性方面, CQWPEA 算法的寻优性能明显优于其他 2 种算法。

另外,从算法收敛(达到收敛精度)时间方面, WPEA 算法对 6 个标准测试函数 20 次实验的收敛消耗总时间分别为 31.55、91.34、77.39、33.74 和 31.59 s,而 QWPEA 算法对 6 个标准测试函数 20 次实验的收敛消耗总时间分别为 3.69、3.24、2.62、3.20、3.10 和 3.25 s, CQWPEA 算法对 6 个标准测试函数 20 次实验的收敛消耗总时间分别为 0.46、0.82、0.52、0.72、0.84 和 0.55 s。从收敛消耗总时间来看,与 WPEA 算法和 QWPEA 算法相比, CQWPEA 算法的收敛速度明显加快。从图 6 中可清晰地看出, CQWPEA 算法要比 WPEA 算法和 QWPEA 算法具有较快的收敛速度。

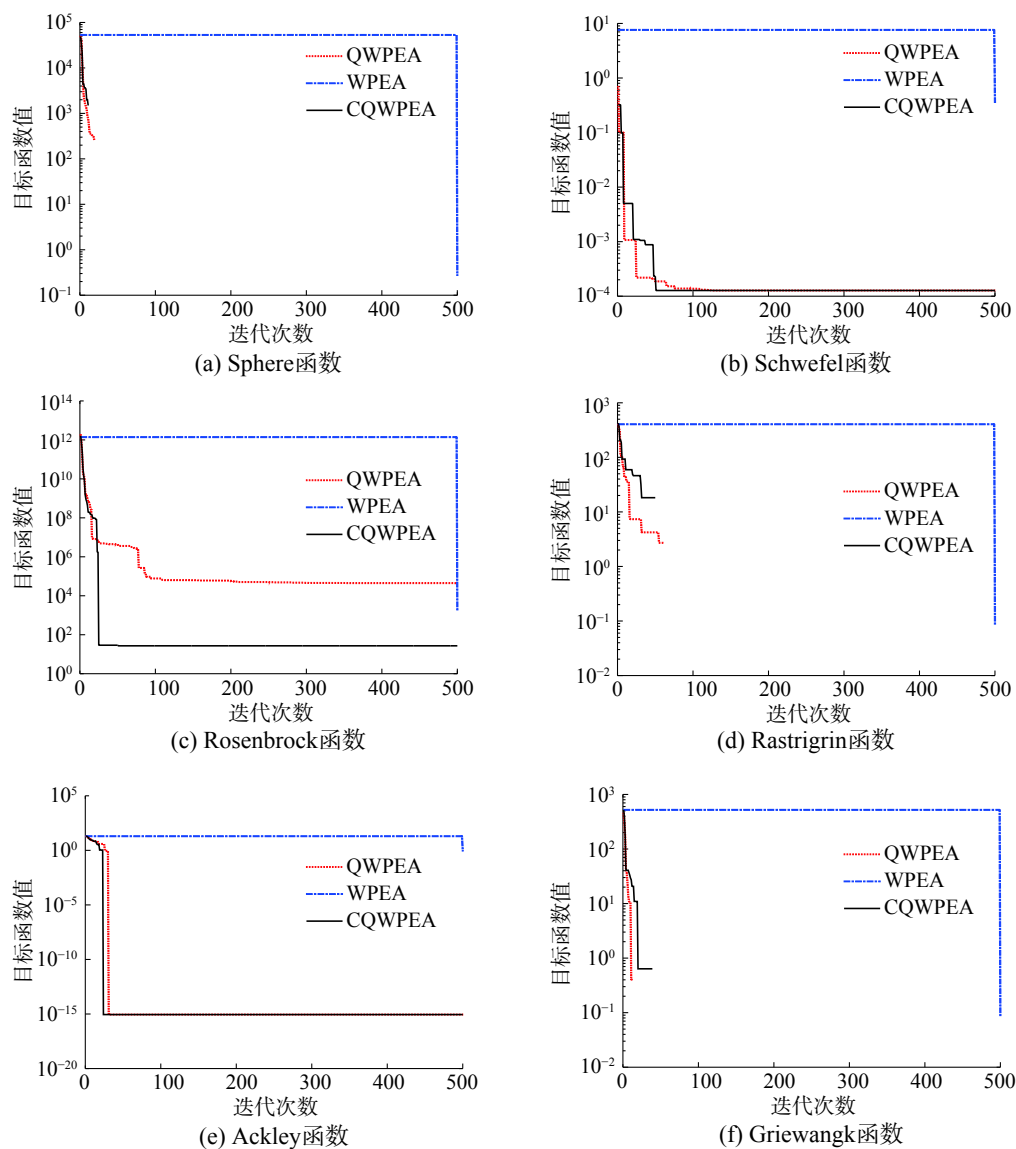


图6 6个测试函数的进化曲线

Fig. 6 Evolutionary curves for six test functions

5 结束语

1) 基于元胞自动机和量子狼群演化算法的原理, 提出一种求解离散优化问题的元胞量子狼群演化算法, 该算法采用双策略方法, 实现量子人工狼群初始位置的生成。

2) 针对狼群中个体狼的位置与猎物资源的关系问题, 采用二进制编码方式和元胞自动机中的演化规则, 分别对狼群中个体狼与猎物间的距离进行精确描述和量子旋转角的选取、调整, 实现狼群个体与猎物资源位置的精确定位和增强狼群的全局搜索空间能力。

3) 元胞量子狼群演化算法是对解决离散空间优化问题的初步探索, 该算法具有寻优精度高、收敛速度快以及较强的鲁棒性, 将CQWPEA算法进行改进以加强算法的寻优能力, 并用于处理复

杂约束优化问题是进一步研究的方向。

参考文献:

- [1] MECH L D. The wolf: the ecology and behavior of an endangered species [M]. New York: Natural History Press, 1970.
- [2] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. Advances in engineering software, 2014, 69: 46–61.
- [3] EMARY E, ZAWBAA H M, HASSANIEN A E. Binary grey wolf optimization approaches for feature selection[J]. Neurocomputing, 2016, 172: 371–381.
- [4] 吴虎胜, 张凤鸣, 战仁军, 等. 利用改进的二进制狼群算法求解多维背包问题[J]. 系统工程与电子技术, 2015, 37(5): 1084–1091.

WU Husheng, ZHANG Fengming, ZHAN Renjun, et al.

- Improved binary wolf pack algorithm for solving multidimensional knapsack problem[J]. Systems engineering and electronics, 2015, 37(5): 1084–1091.
- [5] 吴虎胜, 张凤鸣, 战仁军, 等. 求解 0-1 背包问题的二进制狼群算法[J]. 系统工程与电子技术, 2014, 36(8): 1660–1667.
- WU Husheng, ZHANG Fengming, ZHAN Renjun, et al. A binary wolf pack algorithm for solving 0-1 knapsack problem[J]. Systems engineering and electronics, 2014, 36(8): 1660–1667.
- [6] SRIKANTH K, PANWAR L K, PANIGRAHI B K, et al. Meta-heuristic framework: quantum inspired binary grey wolf optimizer for unit commitment problem[J]. Computers & electrical engineering, 2017, 14(21): 1–18.
- [7] MUANGKOTE N, SUNAT K, CHIEWCHANWAT-TANA S. An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets[C]//Proceedings of 2014 International Computer Science and Engineering Conference. Khon Kaen, Thailand, 2014: 209–214.
- [8] 金杉, 金志刚. 基于量子狼群进化的多目标汇聚节点覆盖算法[J]. 电子与信息学报, 2017, 39(5): 1178–1184.
- JIN Shan, JIN Zhigang. Multi-objective sink nodes coverage algorithm based on quantum wolf pack evolution[J]. Journal of electronics & information technology, 2017, 39(5): 1178–1184.
- [9] VON NEUMANN J, BURKS A W. Theory of self-reproducing automata[M]. Urbana: University of Illinois Press, 1966.
- [10] 奚茂龙, 孙俊, 吴勇. 一种二进制编码的量子粒子群优化算法[J]. 控制与决策, 2010, 25(1): 99–104.
- XI Maolong, SUN Jun, WU Yong. Quantum-behaved particle swarm optimization with binary encoding[J]. Control and decision, 2010, 25(1): 99–104.
- [11] 张丽娟, 张艳芳, 赵宜宾. 基于元胞自动机的智能疏散模型的仿真研究[J]. 系统工程理论与实践, 2015, 35(1): 247–253.
- ZHANG Lijuan, ZHANG Yanfang, ZHAO Yibin. Simulation research on intelligent evacuation model based on cellular automation[J]. Systems engineering—theory & practice, 2015, 35(1): 247–253.
- [12] 赵知劲, 郑仕链, 尚俊娜, 等. 基于量子遗传算法的认知无线电决策引擎研究[J]. 物理学报, 2007, 56(11): 6760–6766.
- ZHAO Zhijin, ZHENG Shilian, SHANG Junna, et al. A study of cognitive radio decision engine based on quantum genetic algorithm[J]. Acta physica sinica, 2007, 56(11): 6760–6766.
- [13] 鲁宇明, 黎明, 李凌. 一种具有演化规则的元胞遗传算法[J]. 电子学报, 2010, 38(7): 1603–1607.
- LU Yuming, LI Ming, LI Ling. The cellular genetic algorithm with evolutionary rule[J]. Acta electronica sinica, 2010, 38(7): 1603–1607.
- [14] 邱玉霞, 谢克明. 基于泛函分析的思维进化算法收敛性研究[J]. 计算机工程与应用, 2006(22): 36–38.
- QIU Yuxia, XIE Keming. A Study on convergence of mind evolutionary algorithm based on functional analysis[J]. Computer engineering and application, 2006(22): 36–38.

作者简介:



专利 4 项。发表学术论文 9 篇。



版学术著作 2 部。



发表学术论文 30 余篇。

马龙, 男, 1982 年生, 讲师, 博士研究生, 主要研究方向为群智能算法、系统建模与优化。参与国家自然科学基金青年项目 1 项、国家自然科学基金面上项目 1 项、陕西省自然科学基金项目 1 项、陕西省教育厅专项科研项目 1 项, 横向项目 1 项。获得

卢才武, 男, 1965 年生, 教授, 博士, 主要研究方向为系统优化理论、信息技术和信息管理。主持省部级项目多项、横向项目多项。获中国工业大奖提名奖 1 项、中国有色科技进步二等奖 1 项, 授权软件著作权多项, 发明专利多项。发表学术论文 80 余篇, 出

顾清华, 男, 1981 年生, 副教授, 博士, 主要研究方向为系统优化理论、矿业系统工程优化。主持国家自然科学基金 2 项, 省部级项目多项、横向项目多项。获得洛阳市科技进步一等奖 1 项, 西安市科技进步二等奖 1 项, 授权软件著作权多项、发明专利多项。