

# 基于量子蛙跳协同进化的粗糙属性快速约简

丁卫平<sup>1,2</sup>, 王建东<sup>1</sup>, 管致锦<sup>2</sup>

(1. 南京航空航天大学计算机科学与技术学院, 江苏南京 210016; 2. 南通大学计算机科学与技术学院, 江苏南通 226019)

**摘 要:** 属性约简是粗糙集理论研究的重要内容, 现已证明求决策表最小约简是一个典型 NP 难题. 本文提出一种基于量子蛙群协同进化的粗糙属性快速约简算法. 该算法构造一种动态多簇的蛙群结构, 用量子态比特进行蛙群个体编码, 以自适应量子旋转角调整、量子变异和量子纠缠等策略加速蛙群进化收敛, 各簇蛙群以双向协同学习机制共享属性约简中相关信息. 标准 Benchmark 优化函数测试结果表明该算法在保证收敛速度同时具有较强的平衡全局优化与局部细致搜索能力. 在 UCI 数据集上进行属性约简比较实验, 结果验证了本算法在属性约简精度和效率方面具有明显优势.

**关键词:** 属性约简; 量子蛙群进化; 双向协同; 自适应量子旋转角; 动态多簇结构

**中图分类号:** TP301.6      **文献标识码:** A      **文章编号:** 0372-2112 (2011) 11-2597-07

## Efficient Rough Attribute Reduction Based on Quantum Frog-Leaping Co-Evolution

DING Wei-ping<sup>1,2</sup>, WANG Jian-dong<sup>1</sup>, GUAN Zhi-jin<sup>2</sup>

(1. College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China;

2. School of Computer Science and Technology, Nantong University, Nantong, Jiangsu 226019, China)

**Abstract:** Attribute reduction is a key studying point of rough set theory. It has been proven that computing minimal reduction of decision tables is a NP-hard problem. An efficient attribute reduction algorithm based on the quantum frog-leaping co-evolution is proposed. A dynamic multi-cluster frog structure is designed, individuals are represented by multi-state qubits. The self-adaptive adjustment of quantum rotation angle, quantum mutation and quantum entanglement strategies are applied to accelerate the convergence. Cooperative searching information of different clusters during attribute reduction is shared by adopting the bidirectional co-evolutionary method. Experiments on some benchmark problems indicate the proposed algorithm has outstanding ability to balance the global exploitation and local exploration on condition of the good convergence, and results on some UCI data sets validate it is more competitive on the attribute reduction accuracy and efficiency, compared to the traditional evolutionary algorithms.

**Key words:** attribute reduction; quantum frog evolution; bidirectional cooperation; self-adaptive quantum rotation angle; dynamic multi-cluster structure

## 1 引言

粗糙集(Rough Set)理论是 Pawlak Z 教授于 1982 年提出的一种能够定量分析并处理不精确、不一致、不完整信息与知识的数学工具, 其基本思想是通过关系数据库分类归纳形成概念和规则, 通过等价关系分类及对目标近似来实现知识发现等. 基于粗糙集的研究主要集中在属性约简、规则获取及计算智能算法等方面, 其中属性约简已发展为一种新的处理数据挖掘相关问题的有

效方法, 在智能信息处理方面有着广泛应用<sup>[1~3]</sup>. 然而 Wong S K M 和 Ziarko W 已经证明找出决策表最小属性约简是一个 NP-Hard 难题, 至今仍未找到一种通用且高效的解决方法为其求解提供有效途径<sup>[4]</sup>. 传统的属性约简方法是每次选择最重要或分类能力最强具有启发的一个属性加入约简集, 反复进行约简, 效率不高且往往却无法保证找到的约简集为最小属性约简集.

量子计算是利用量子世界的叠加态、相干性和纠缠性等特点使量子信息系统突破了经典信息系统极限, 其

以独特的计算性能成为研究热点,尤其是在通过量子并行计算尝试解决经典计算中的 NP 问题方面取得了较好研究成果,如经典的大整数质因子分解用传统计算机实现时间复杂度约为  $O(2^{n^{1/3}})$ ,而用 Shor P W 提出的量子算法将大质数分解转换为 P 问题,在量子计算机上约为  $O(n^3)$  内就能完成<sup>[5]</sup>,这种办法虽不能给出对所有 NP 问题都可进行有效解答的方法,但启发人们思考 NP 问题有可能存在更深层结构可以用量子计算去快速求解. Shor P W 提出的量子算法极大地促进了量子计算的发展,进一步激发人们去寻求其他 NP 问题可能存在的量子算法热潮.

蛙跳算法(SFLA)是 2003 年 Eusuff M M 和 Lansey K E 提出的一种新的基于群智能启发式的协同进化算法,该算法融合了基于遗传基因的 Memetic 算法和基于群体觅食行为的 PSO 算法优点,具有概念简单、参数少、易于编程实现和寻优能力强等优势<sup>[6]</sup>. 近年来 SFLA 算法不断得到完善和工程应用<sup>[7,8]</sup>,但该算法目前尚未成熟,有些关键问题值得研究:如蛙群结构如何合理划分、不同蛙群间采用何种机制进行寻优经验共享等.

目前国内外已有一些学者将量子计算和各种进化算法相结合提出了相应的量子进化算法:如 Narayanan A 等人将遗传算法和量子理论相结合提出量子遗传算法,它被认为是量子进化算法雏形,但进化算子仍使用传统的交叉和变异操作来产生新个体<sup>[9]</sup>. Han K H 等人采用量子染色体表示个体并引入量子旋转门作为进化算子提出一种新型量子启发式遗传算法<sup>[10]</sup>. 解光军等人将量子门组作为神经网络激活函数提出一种量子神经计算新型模型<sup>[11]</sup>. Licheng Jiao 等人将量子计算应用于人工免疫系统中的克隆算子提出一种基于量子编码免疫克隆算法<sup>[12]</sup>. 邢焕来等人采用动态步长策略调整量子门旋转角度提出了一种新型的改进量子遗传算法等<sup>[13]</sup>. 上述量子进化算法较传统进化算法具有更好性能:如用特殊的量子位表示形式可带来种群进化多样性,较好地提高了算法运行效率等,但在算法收敛性、尤其是量子旋转门自适应调整能力较差. SFLA 算法是一种亚启发式协同搜索群智能算法,具有较好的寻优能力和收敛速度,目前将量子计算相关操作引入到 SFLA 算法进化过程的研究还没有相关文献报道.

如何采用高效的启发式智能算法处理属性约简 NP 问题是一项具有挑战性的研究课题. 本文基于上述研究问题,充分发挥量子计算与蛙跳协同进化算法各自优点,设计了一种具有动态多簇的蛙群结构,将量子态结构特性引入蛙跳算法进行量子比特编码,提出一种基于量子蛙跳协同进化的属性快速约简算法. 该算法以量子旋转门动态自适应调整、量子变异和量子纠缠

等方法实现蛙群内个体进化,各簇蛙群充分进行寻优信息共享,相关仿真实验表明本文算法具有较强的寻优能力,能有效找到最小属性约简集.

## 2 粗糙属性约简

粗糙集中属性约简是指在保证原信息系统分类或决策能力不变的条件下,删除条件属性中冗余属性,找到一个尽可能小的属性子集,简化后的属性集将能提供和原始属性集相同的分类质量.

**定义 1** 设决策系统  $S = (U, C \cup D, V, f)$ , 其中  $C, D$  分别表示条件属性和决策属性,决策属性在条件属性下正域定义为:

$$POS_C(D) = \bigcup_{x \in U/D} \underline{C}(x) \quad (1)$$

$POS_C(D)$  表明根据  $C$  的知识所进行划分  $U/C$ , 能够确切划入  $U/D$  类的对象集合.

**定义 2** 决策属性  $D$  对条件属性  $C$  的依赖度定义为:

$$\gamma_C(D) = \frac{|POS_C(D)|}{|U|} \quad (2)$$

依赖度  $\gamma_C(D)$  表示在条件属性  $C$  下,确切划入决策在  $U/D$  的对象占论域中总对象的相应比率.

**定义 3** 约简是条件属性集合  $C$  的一个子集  $R$ , 且满足  $\gamma_R(D) = \gamma_C(D)$ , 其约简集合  $RED$  定义为:

$$RED = \{ R \subseteq C \mid \gamma_R(D) = \gamma_C(D), \forall B \subseteq R, \gamma_B(D) \neq \gamma_C(D) \} \quad (3)$$

所有  $C$  的属性约简交集记  $Core(C)$ .

**定义 4** 最优属性约简是约简集合  $RED$  中具有最小势约简  $RED_{opt}$ , 定义为:

$$RED_{opt} = \{ R \in RED \mid \forall R' \in RED, |R| \leq |R'| \} \quad (4)$$

## 3 改进的动态多簇蛙群协同算法

协同进化算法是在个体和种群之间引入协作、竞争或更复杂的协同学习机制实现进化. SFLA 算法是典型的协同进化算法,它模拟一群青蛙在一片湿地跳动觅食行为. 其首先随机生成  $P$  只青蛙组成初始群体,第  $i$  只青蛙表示解空间解为  $X_i = (x_{i1}, x_{i2}, \dots, x_{is})$  ( $s$  表示解空间维数),将种群内青蛙个体按适应度降序排列,然后将整个青蛙群体分成  $m$  个簇群,每个簇群包含  $n$  只青蛙,满足  $P = m \times n$ ,其中第 1 只青蛙划分至第 1 簇群,第 2 只青蛙划分至第 2 簇群,第  $m$  只青蛙划分至第  $m$  簇,第  $m+1$  只青蛙划分至第 1 簇群,第  $m+2$  只青蛙划分至第 2 簇群,依次类推,直到所有青蛙进入不同簇群. 对于每一个划分簇群,将其中最好适应度解表示为  $X_b$ ,最差适应度解表示为  $X_w$ ,将所有簇群中具有全局最好簇平均适应度解表示为  $X_B$ ,全局最差簇适应度解表

示为  $X'_w$ . 在每个簇群局部深度搜索,更新策略为:

$$S = \text{Rand}(\ ) \times (X_b - X_w) \quad (5)$$

$$\text{New}X_w = X_w + S \quad (-S_{\text{MAX}} \leq S \leq S_{\text{MAX}}) \quad (6)$$

上式中  $\text{Rand}(\ )$  表示  $[0 \ 1]$  随机数,  $S$  表示青蛙可跳动步长最大值,  $\text{New}X_w$  表示更新后  $X_w$  值, 如果  $\text{New}X_w$  适应值优于  $X_w$  所对应适应值, 则取代原来簇群中相应解; 如果  $\text{New}X_w$  适应值低于  $X_w$  所对应适应值, 则用  $X'_b$  代替  $X_b$ , 重新更新  $X_w$ ; 如果仍没改进, 则随机产生一个新的蛙来代替  $X_w$ . 每次迭代完成后所有簇群内青蛙重新混合、排序和划分, 如此反复至满足终止条件<sup>[6]</sup>.

上述算法在每次混合排后重新划分的簇数  $m$  是固定, 易陷入局部最优. 为了进一步加强 SFLA 算法全局收敛能力, 本文设计了一种改进的动态多簇蛙群结构, 即在每次迭代完成后, 将所有群蛙重新划分的簇数  $m$  逐渐增大, 单个簇群内青蛙个数  $n$  相应减少, 但满足  $P = m \times n$ , 动态簇数  $m$  定义如下:

$$m \leftarrow \left\lceil \frac{m+1}{m} \times \frac{1}{\sqrt{2^m}} \sum_{y=0}^{2^m-1} e^{2\pi y} \right\rceil \quad (7)$$

迭代开始时整个蛙群被划分的簇数  $m$  相对较小, 单个簇群内青蛙个数  $n$  较大, 簇内收敛速度较快, 具有较强的局部搜索能力, 可将蛙群快速引向有希望搜索区域; 随着  $m$  的逐渐变大,  $n$  减少, 则算法收敛速度较慢, 但具有较强的全局搜索能力, 能尽快找到全局最优解, 避免陷入局部最优, 算法的稳定性和鲁棒性较好.

在进化学习过程中, 各簇蛙群内个体各自分别进化, 但不同簇群体间又相互作用, 蛙群簇内和簇间均采用双向的协同学习机制: 对于每一个按适应值排好序蛙簇群, 簇内所有蛙个体既要向适应度最好解  $X_b$  学习, 同时也要向适应度最差解  $X_w$  学习经验, 以防止不犯或者少犯同样错误, 进化性能可得到进一步加强, 在解空间进行局部深度搜索时, 其速度和位置更新公式如下:

$$\begin{aligned} v_{id}(t+1) &= w \times v_{id}(t) + c_1 \times r_1 \times [p_{id}(t) - x_{id}(t)] \\ &+ \rho_j \times c_2 \times r_2 \times [X_b - x_{id}(t)] \\ &+ (1 - \rho_j) \times c_3 \times r_3 \times [X_w - x_{id}(t)] \end{aligned} \quad (8)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1) \quad (9)$$

(公式中  $\rho_j$  和  $(1 - \rho_j)$  为第  $j$  次迭代向最优解和最差解的学习因子比,  $\rho_j = X_b / (X_b + X_w)$ , 其余相关参数和经典 PSO 定义一致)

当每簇蛙群局部深度搜索完成后, 再以簇为群体单元在解空间进行整体协同寻优信息学习, 此时同样既要学习平均适应度最好簇群解  $X'_b$ , 又要学习最差簇群解  $X'_w$ , 更新公式和上述簇内公式类似.

实验结果表明改进的动态多簇蛙群协同算法具有

较强全局信息交换和局部深度搜索的平衡能力, 在簇内充分进行个体间信息交流, 既学习较好的经验, 也从错误中总结教训, 当进化到一定阶段后, 以各簇为群体单元再进行蛙簇间寻优经验共享, 形成了个体与个体、群体与群体两阶段协同进化学习, 该算法具有较高的搜索精度和收敛性, 且具有较强的跳出局部最优能力.

## 4 基于量子蛙跳协同进化的属性快速约简算法

### 4.1 相关量子蛙群算子设计

#### 4.1.1 量子编码与量子旋转门

在量子蛙跳协同算法中采用基于量子比特蛙群个体编码方式, 即用一对复数  $(\alpha, \beta)$  定义一个量子比特位, 其量子态可表示为:

$$|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle \quad (10)$$

式中  $\alpha$  和  $\beta$  是复数系数, 量子态  $|\varphi\rangle$  被测量处于  $|0\rangle$  概率为  $|\alpha|^2$ , 处于  $|1\rangle$  概率为  $|\beta|^2$ . 蛙群个体用多个量子比特的量子染色体存储, 每个个体所表达包含所有可能概率的线性叠加态, 簇内蛙群的任一操作也同时作用于所有可能信息, 这使量子蛙跳算法比经典蛙跳算法拥有更好的多样性特征, 并通过  $|\alpha|^2 + |\beta|^2 = 1$  实现收敛.

量子旋转门  $U(\theta)$  是量子蛙跳算法的实际操作过程, 通过其旋转实现量子比特相应更新, 定义如下:

$$\begin{aligned} \begin{bmatrix} \alpha'_i \\ \beta'_i \end{bmatrix} &= U(\theta) \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \end{aligned} \quad (11)$$

其中  $[\alpha'_i, \beta'_i]^T$  表示量子操作后蛙群个体中第  $i$  位量子比特,  $\theta_i$  表示第  $i$  位量子比特所对应量子旋转门的旋转角. 量子旋转门操作将改变叠加态的概率幅度, 实验表明本量子蛙跳算法有着极强的全局搜索能力, 该性能是由量子旋转门驱动的内在机制所决定.

#### 4.1.2 自适应量子旋转角调整算子

量子旋转角度设置是量子旋转门操作的关键. 传统的旋转角度选取一般采用如文献[14]中所述的查表机制, 旋转角  $\theta_i = s(\alpha_i, \beta_i)\Delta\theta$ ,  $s(\alpha_i, \beta_i)$  代表  $\theta_i$  旋转方向,  $\Delta\theta$  代表  $\theta_i$  旋转大小, 查表机制中参数  $\delta$  为每次调整角步长,  $\delta$  值太小将影响收敛速度,  $\delta$  值太大可能会使结果发散或早熟收敛到局部最优解等, 所以每次旋转角调整时  $\delta$  值合适选取非常难, 范围一般限定为  $[0.005\pi \sim 0.200\pi]$ , 实际应用效果较差.

本文基于文献[13]动态调整旋转角思想, 结合改进的动态多簇蛙群协同算法, 提出了一种自适应动态量子旋转角调整策略, 下面给出量子旋转角  $\theta_i$  形式化公式和调整算子的具体步骤:

设两个量子比特 ( $|\varphi\rangle, |\varphi'\rangle$ ) 之间角距离为  $\Delta\theta_{|\varphi\rangle, |\varphi'\rangle} = \arctan(\alpha'/\beta') - \arctan(\alpha/\beta)$ ,  $|\varphi\rangle = \alpha|0\rangle + \beta|1\rangle$ ,  $|\varphi'\rangle = \alpha'|0\rangle + \beta'|1\rangle$ , 自适应量子旋转角  $\theta_i$  定义为:

$$\theta_i = (1 - \frac{X'_B - X'_W}{X_b - X_w})^m \times \Delta\theta_{|\varphi\rangle, *}, \quad (12)$$

$\Delta\theta_{|\varphi\rangle, *}$  为量子蛙群染色体第  $i$  个量子比特位与基态间角距离, 量子旋转角调整步骤如下:

#### 算法 1 自适应量子旋转角调整

If  $((f(b) \geq f(x)) \wedge ((b_i = 0) \wedge (x_i = 1)))$

$$\Delta\theta_{|\varphi\rangle, *}, * = \Delta\theta_{|\varphi\rangle, |0\rangle} = -\arctan \frac{\alpha_i}{\beta_i}$$

$$\theta_i = -(1 - \frac{X'_B - X'_W}{X_b - X_w})^m \arctan \frac{\alpha_i}{\beta_i};$$

// 自适应增加量子蛙群处于  $|0\rangle$  概率

Else if  $((f(b) \geq f(x)) \wedge ((b_i = 1) \wedge (x_i = 0)))$

$$\Delta\theta_{|\varphi\rangle, *}, * = \Delta\theta_{|\varphi\rangle, |1\rangle} = \frac{\pi}{2} - \arctan \frac{\alpha_i}{\beta_i}$$

$$\theta_i = (\frac{\pi}{2} - (1 - \frac{X'_B - X'_W}{X_b - X_w})^m) \times \arctan \frac{\alpha_i}{\beta_i};$$

// 自适应增加量子蛙群处于  $|1\rangle$  概率

Else  $\Delta\theta_{|\varphi\rangle, *}, * = 0$ .

其中  $x_i$  为  $|\varphi_i\rangle$  测量值,  $b_i$  为当前最优解  $b$  的第  $i$  个量子比特位,  $\alpha_i$  和  $\beta_i$  为  $|\varphi_i\rangle$  概率幅, 随着进化代数增加和蛙群被划分的簇数减少, 动态增加蛙群量子比特处于  $|0\rangle$  或  $|1\rangle$  的概率, 使得量子蛙群染色体进化中所有量子比特都将朝着与最优解  $b$  对应的量子比特基态自适应旋转调整.

#### 4.1.3 量子变异操作算子

为了避免进化陷入局部最优, 本文采用量子变异算子来增加进化蛙群的多样性, 加速全局收敛. 在进化开始尽可能地搜索每个蛙簇空间, 随着进化将减少每个簇内参与变异的量子蛙群个体数目, 其变异概率  $P_{mutation}$  经验公式如下:

$$P_{mutation} = 0.25 + 0.30 \times (1 - \frac{Current\_gen}{Max\_gen})^m \quad (13)$$

其中  $Max\_gen$  为量子蛙群进化总代数,  $Current\_gen$  为当前进化代数, 其具体变异步骤如下:

- ①以  $P_{mutation}$  概率在每个蛙簇内中选取部分个体参与变异;
- ②在选中蛙群个体染色体中随机选取一个量子比特;
- ③将选定量子比特的几率幅对  $(\alpha_i, \beta_i)$  进行前后置换操作.

实验表明  $P_{mutation}$  变异因子能产生均匀变异概率分布数, 量子叠加状态将使原倾向于状态“1”的更易倾向于状态“0”, 或相反.

#### 4.1.4 量子纠缠操作算子

复合系统的一个纯态, 如果不能写成其子系统纯

态的直积, 那它就是纠缠态. 对处于纠缠态的量子位的某几位操作, 不但会改变这些量子位状态, 还会改变与它们相纠缠的其他量子位状态. 量子蛙群经过  $P_{mutation}$  变异后, 再采用量子纠缠算子可较好实行非定域量子门操作, 提高其成功几率, 纠缠算子定义为:

$$U'(\theta) = \cos(\frac{\theta}{2}) I_A \otimes I_B + i \times \sin(\frac{\theta}{2}) \sigma_z^A \otimes \sigma_z^B, \quad (\theta \in [0, \frac{\pi}{2}]) \quad (14)$$

纠缠步骤与文献[15]中 Jonathan D 等人提出的经典两粒子纠缠态转化步骤一致, 实验证明该纠缠算子在粗糙属性约简中纠缠量越大, 成功几率就越大, 其属性约简集就越向最小属性集逼近.

#### 4.2 算法核心步骤描述

本文提出的粗糙属性快速约简算法将充分发挥量子蛙跳协同算法在最小属性约简搜索过程中较强的寻优启发作用, 将尽可能地减少所包含的属性数目, 以获得较理想的最小属性约简集, 其具体步骤如下:

#### 算法 2 基于量子蛙跳协同进化的属性约简

Input: 决策系统  $S = (U, C \cup D, V, F)$

Output: 最小属性约简集  $RED_{opt}$

- (1) 利用粗糙集理论中差别矩阵方法求  $S$  的核属性  $Core(D)$ ;
- (2) 如果  $(\gamma_R(D) = \gamma_C(D))$ , 则  $RED_{opt} \leftarrow Core(D)$  算法结束; 否则 Goto (3);
- (3) 通过  $\xi$  映射:  $\xi: \{0, 1\}^m \rightarrow 2^{|C|}$  将属性  $C_i \in \xi(x) \Leftrightarrow x_i = 1, i = 1, 2, \dots, m$ ;
- (4) 令  $\gamma_{\xi(x)} = \gamma_C, Core(\xi(x)) = \xi(x)$ , 将属性约简优化转换为一个二进制目标优化模型  $F(x) = \min(C(x)) = \sum_{i=1}^m x_i$ ;

- (5) 初始化蛙群规模为  $N$ , 蛙群维数  $H = |D| - |Core(D)|$ , 每簇蛙群数为  $n$ , 初始化  $Current\_gen = 0$ , 用布尔向量  $x = \{x_1, x_2, \dots, x_m\} \in \{0, 1\}^m$  代表蛙簇群,  $x_i (i \in [1, N])$  为第  $i$  个蛙个体;
- (6) 初始化量子蛙群染色体位数为  $m$ , 染色体  $Q(t)$  中  $\alpha'_i, \beta'_i (i = 1, 2, \dots, m)$  初始化为  $1/\sqrt{2}$ , 即所有线性叠加均等概率, 算法初始搜索所有状态均等概率进化;
- (7) 根据  $Q(t)$  中蛙群个体概率幅构造量子蛙群叠加态的观测态  $P(t)$ , 其中  $P(t) = \{x'_1, x'_2, \dots, x'_m\}$  为每一个观测态;
- (8) 对观测态  $P(t)$  计算每簇蛙群中个体适应度和每个簇群平均适应度;
- (9) 保留适应值最佳  $X_b$  和最差  $X_w$  蛙个体以及最佳  $X'_B$  和最差  $X'_W$  簇群;
- (10) 以自适应量子旋转角旋转算子  $\theta_i$  进行每簇蛙群中个体频率幅自适应调整;
- (11) 以量子变异算子  $P_{mutation}$  在每簇蛙群选取概率个体变异, 实现蛙群进化多态性;
- (12) 按量子纠缠算子  $U'(\theta)$  进行各簇蛙群间的量子纠缠操作, 提高约简最小集成功几率;
- (13) 进行量子蛙群当前代簇内个体  $X_b$  和  $X_w$ 、簇间群体  $X'_B$  和  $X'_W$  协同学习;

- (14)  $Current\_gen \leftarrow Current\_gen + 1$ ,按公式(7)逐步增加蛙群簇数  $m$ , 在每个簇内深入探索后增强蛙族群全局收敛能力;
- (15) 计算整个量子蛙群属性适应度,并代入步骤(4)的属性约简目标模型,取整个蛙群中最优蛙适应度为当前全局最优,如果连续 10 次  $|F(x_m) - F(x_{m-1})| \rightarrow 0$ ,则算法终止循环;如果迭代过程中全局最优目标值  $F(x)$ 降低,或  $F(x)$ 出现来回振荡,则取到目前迭代为止每次最优簇适应度的平均值为全局最优属性值,即:

$$\overline{F(x)} = \frac{1}{Current\_gen} \left( \sum_{i=1}^{Current\_gen} X'_{Bi} \right)$$

( $X'_{Bi}$ 为第  $i$  迭代中最优簇平均适应度)

- (16) 如不满足上述情况,则重新循环, Goto (5);
- (17) 输出最小属性约简集  $RED_{opt}$ .

## 5 仿真实验及分析

### 5.1 函数优化测试与分析

为了评价本文提出的基于量子蛙跳协同进化的属

表 1 标准复合测试函数优化比较(平均值 ± 标准方差)

函数	文献[7]算法	文献[8]算法	文献[13]算法	本文算法
$f_1$	$7.3900e-005 \pm 4.7945e-005$	$6.8817e-005 \pm 2.3025e-006$	$5.7988e-007 \pm 4.6009e-007$	<b><math>4.7567e-008 \pm 3.2545e-008</math></b>
$f_2$	$5.7845e+002 \pm 3.5536e+001$	$5.9010e-002 \pm 2.6468e-002$	$6.8090e-001 \pm 3.7909e-001$	<b><math>5.3990e-004 \pm 1.2800e-004</math></b>
$f_3$	$3.3788e+002 \pm 2.8062e+002$	$2.3700e+003 \pm 1.9453e+003$	$4.2661e+002 \pm 6.9504e+002$	<b><math>5.0977e+000 \pm 4.0370e+000</math></b>
$f_4$	$8.9785e-003 \pm 5.3156e-003$	$8.0466e-005 \pm 4.7229e-005$	$2.9345e-003 \pm 4.5551e-003$	<b><math>1.2348e-007 \pm 1.6032e-007</math></b>
$f_5$	$1.3919e-006 \pm 5.2340e-006$	<b><math>2.5090e-011 \pm 1.4539e-011</math></b>	$6.7893e+001 \pm 1.7452e+000$	$6.0458e-010 \pm 5.3423e-010$
$f_6$	$4.0900e-007 \pm 2.9011e-007$	$1.0817e-005 \pm 4.4022e-005$	$5.6909e-011 \pm 2.1471e-011$	<b><math>9.2340e-014 \pm 4.0113e-014</math></b>
$f_7$	$1.3135e+002 \pm 3.2455e+002$	$1.3421e-006 \pm 6.5445e-005$	$4.5694e-003 \pm 1.4543e-002$	<b><math>4.4565e-009 \pm 4.4861e-009</math></b>
$f_8$	$4.3423e-008 \pm 4.1761e-007$	$4.5656e-003 \pm 5.6436e-003$	$2.3003e-004 \pm 5.3424e-004$	<b><math>4.3682e-010 \pm 4.2316e-009</math></b>
$f_9$	$5.4569e-009 \pm 4.1223e-009$	$8.1564e-005 \pm 6.2333e-006$	<b><math>9.2351e-013 \pm 1.5441e-013</math></b>	$5.4574e-010 \pm 2.0098e-010$
$f_{10}$	$7.2345e-003 \pm 0.3687e-004$	$2.6345e-003 \pm 6.1247e-003$	$6.3457e-004 \pm 0.6749e-004$	<b><math>6.1250e-006 \pm 2.1750e-006</math></b>

从表 1 可以看出本文算法在函数  $f_1, f_2, f_3, f_4, f_6, f_7, f_8, f_{10}$  上所得到的最终位置比其他三个算法更接近于全局最优点,其标准方差也相对较低,收敛速度明显快于其他三个算法.在函数  $f_5$  和  $f_9$  虽没有取得最佳值,但也接近于文献[8]和[13]所达到最优结果.这主要由于动态多簇蛙群结构具有较强的全局信息交换和局部深度搜索平衡策略,在各个簇内能充分进行个体间信息传递,并以各簇蛙群为单元再进行簇间混合操作.另外进化过程中以自适应量子旋转角和动态变异、纠缠等策略实现量子蛙群进化,增强其多样性,提升跳出局部最优能力,快速向全局最优收敛,收敛稳定性也相对较好.

### 5.2 UCI 数据集约简实验与分析

为进一步检验算法在属性快速约简方面性能,本文设计基于典型 UCI 机器学习数据库属性约简对比实验,(实验数据库地址 <http://ftp.ics.uci.edu/pub/machine-learning-databases>)实验条件和 5.1 节相同,对比算法选取了文献[16]和文献[17]算法,所有算法均在每个 UCI 数据集上进行 20 次 20 倍交叉验证,实验结果为去

性快速约简算法性能,我们选取国际公认典型的复合难优化函数进行仿真测试,实验环境为 Microsoft Windows 2003 Server, MS SQL Server 2005, Intel 双核 CPU 1.66 GHz, 2GB 内存, 100 GB 硬盘,采用 Matlab7.0 编程实现算法,  $f_1 \sim f_{10}$  优化函数维度、取值范围、最大迭代数及优化目标精度等参数均和参考文献[18]一致.实验中将本文算法与文献[7]、文献[8]以及文献[13]算法进行比较.设置蛙群规模  $N$  为 500, 初始簇数  $m$  为 20, 维数  $D$  为 30 维,自适应动态旋转角  $\theta_{\min}$  和  $\theta_{\max}$  分别取  $0.050\pi$  和  $0.100\pi$ ,量子自适应旋转角、变异概率和纠缠操作算子按 4.1 节公式转换,被比较算法的初始化条件与其对应文献一致,实验中将 4 个算法对  $f_1 \sim f_{10}$  独立运行 50 次,陷入局部最优条件为连续 10 代最优结果无变化.4 个算法平均最优适应函数值及其标准偏差如表 1 所示.

除最初 1 次和最后 1 次,中间 18 次实验的平均结果.图 1 给出了上述 3 个算法在不同数据集上属性约简精度变化比较图,可以看出在不同数据集上,本属性快速约简算法约简精度始终高于文献[16]算法和文献[17]算法,且保持平稳状态.本文算法在约简精度和稳定性方面具有明显优势.

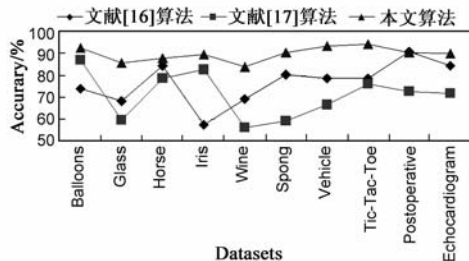


图 1 算法在数据集上约简精度比较

### 5.3 自适应量子旋转角与收敛关系分析

下面分析不同旋转角的调整策略与算法的收敛关系,将自适应量子旋转角范围  $[0.050\pi, 0.100\pi]$  进行划分,选取其中五个固定旋转角,即  $0.060\pi, 0.065\pi, 0.075\pi, 0.085\pi, 0.090\pi$ ,迭代次数为 2000 代,试验次数

仍为 50 次,不同量子旋转角进化中获得的平均值适应度曲线如下图 2 所示,可以看出自适应量子旋转角调整策略在迭代开始由于量子旋转门能自适应调整使其快速跳出局部最优,通过自适应增加蛙群量子比特处于  $|0\rangle$  或  $|1\rangle$  概率,使量子蛙群染色体中所有量子比特都朝最优解对应的量子比特基态进行自适应快速进化,且稳定性较好,而其他固定旋转角调整策略由于  $\delta$  参数影响使其不易跳出局部最优,其全局收敛速度相对较慢.

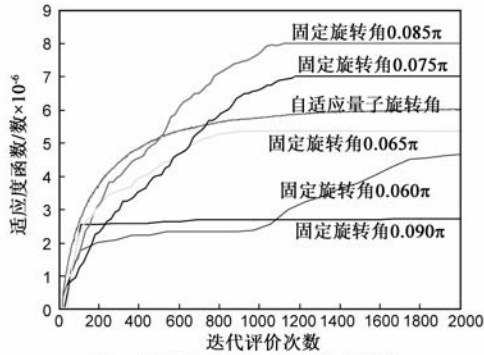


图2 量子旋转角与收敛性关系图

## 6 结论与进一步工作

属性约简是粗糙集理论中的核心问题,为有效获取最小属性约简集,本文将量子计算与蛙群算法融合,设计了基于量子蛙跳协同进化的粗糙属性快速约简算法,该算法利用以量子旋转门动态自适应调整、量子变异策略和量子纠缠方法实现蛙群进化,各簇蛙群内个体与个体、不同簇群与簇群间均采用双向协同学习机制充分进行属性约简中寻信息共享,算法具有较强的全局寻优能力和高效的属性约简性能.相关实验和分析表明本文提出的算法明显优于其他算法,能有效找到全局最小属性约简集,为最小属性约简 NP-Hard 问题解决提供了新的思路.在属性约简过程中如何设计更优秀的量子进化启发算子和协同机制以提高约简性能是我们下一步需要研究的问题.

### 参考文献

- [1] Pawlak Z. Rough sets[J]. International Journal of Computer and Information Sciences, 1982, 11: 341 – 356.
- [2] Peters J F, Skowron A. A rough sets approach to knowledge discovery [J]. International Journal of Intelligent Systems, 2002, 17(2): 109 – 112.
- [3] 胡峰, 王国胤. 属性序下的快速约简算法[J]. 计算机学报, 2007, 30(8): 1429 – 1435.  
Hu F, Wang G Y. Quick reduction algorithm based on attribute order[J]. Chinese Journal of Computer, 2007, 30(8), 1429 – 1435. (in Chinese)
- [4] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of Polish Academy of Science, 1985, 33 (11): 693 – 696.
- [5] Shor P W. Polynomial-time algorithms for prime factorization and discrete logarithms in a quantum computer[J]. SIAM Journal on Computer, 1997, 26 (5): 1484 – 1509.
- [6] Eusuff M M, Lansey K E. Optimization of water distribution network design using the shuffled frog leaping algorithm[J]. Journal of Water Resources Planning and Management, 2003, 129(3): 210 – 225.
- [7] Yang Chengsan, Chuang Liyeh, et al. A combination of shuffled frog-leaping algorithm and genetic algorithm for gene selection [J]. Journal of Advanced Computational Intelligence and Intelligent Informatics, 2008, 12 (3): 218 – 226.
- [8] 罗雪晖, 杨焯, 李霞. 改进混合蛙跳算法求解旅行商问题 [J]. 通信学报, 2009, 30(7): 130 – 135.  
Luo Xuehui, Yang Ye, Li Xia. Modified shuffled frog-leaping algorithm to solve traveling salesman problem [J]. Journal on Communication, 2009, 30 (7): 130 – 135. (in Chinese)
- [9] Narayanan A, Moore M. Quantum-inspired genetic algorithm [A]. Proc. of IEEE International Conference on Evolutionary Computation [C]. Piscatawa: IEEE Press, 1996. 61 – 66.
- [10] Han K H, Kim J H. Quantum-inspired evolutionary algorithms with a new termination criterion, gate, and two-phase scheme [J]. IEEE Trans on Evolutionary Computation, 2004, 8(2): 156 – 169.
- [11] 解光军, 范海秋, 操礼程. 一种量子神经计算网络模型 [J]. 复旦大学学报(自然科学版), 2004, 43(4): 700 – 703.  
Xie Guangjun, Fan Haiqiu, Cao Licheng. A quantum neural computational network model [J]. Journal of Fudan University (Natural Science), 2004, 43 (4): 700 – 703. (in Chinese)
- [12] Licheng Jiao, Yangyang Li, et al. Quantum inspired immune clonal algorithm for global numerical optimization [J]. IEEE Transactions on System, Man and Cybernetics, Part B, 2008, 38(5): 1234 – 1253.
- [13] 邢焕来, 潘炜, 邹喜华. 一种解决组合优化问题的改进型量子遗传算法 [J]. 电子学报, 2007, 35 (10): 1999 – 2002.  
Xing Huanlai, Pan Wei, Zou Xihua. A novel improved quantum genetic algorithm for combinatorial optimization problems [J]. Acta Electronica Sinica, 2007, 35(10): 1999 – 2002. (in Chinese)
- [14] 汪鹏君, 李辉, 等. 量子遗传算法在多输出 Reed-Muller 逻辑电路最佳极性搜索中应用 [J]. 电子学报, 2010, 38(5): 1058 – 1063.  
Wang Pengjun, Li Hui, et al. Application of quantum genetic algorithm in searching for best polarity of multi – output Reed – Muller logic circuits [J]. Acta Electronica Sinica, 2010, 38 (5): 1058 – 1063. (in Chinese)
- [15] Jonathan D, Plenio M B. Minimal condition for local pure-

state entanglement manipulation[J]. Phys Rev Lett, 1999, 83 (2):1455 – 1458.

- [16] Liangjun Ke, Zuren Feng, Zhigang Ren. An efficient ant colony optimization approach to attribute reduction in rough set theory [J]. Pattern Recognition Letters, 2008, 29: 1351 – 1357.
- [17] Dongyi Ye, Zhaojiong Chen, Jiankun Liao. A new algorithm for minimum attribute reduction based on binary particle swarm optimization with vaccination[A]. PAKDD 2007[C]. Nanjing: LNAI 4426, 2007. 1029 – 1036.
- [18] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281 – 295.

### 作者简介



丁卫平 男, 1979 年 1 月出生于江苏金坛. 南京航空航天大学博士研究生, CCF 和 ACM 会员. 主要研究方向机器学习、进化计算和数据挖掘.

E-mail: ding\_wp@nuaa.edu.cn



王建东 男, 1945 年出生于江苏南京, 南京航空航天大学教授、博士生导师. 主要研究方向人工智能和知识工程.



管致锦 男, 1962 年 10 月出生于江苏连云港, 南通大学计算机科学与技术学院教授. 主要研究方向可逆计算、逻辑综合和信息安全.