

一种新的混合量子进化算法^{*})

解 平 李 斌 庄镇泉

(中国科学技术大学电子科学与技术系 合肥 230027)

摘 要 量子进化算法(QEA)用于多峰函数优化时,容易陷入局部最优。本文提出一种新的混合量子进化算法,通过双编码机制(经典二进制编码和量子概率编码),以及经典交叉和量子概率编码更新策略,实现了经典遗传算法与量子进化算法的有机结合,在发挥经典遗传算法全局优化能力的同时,利用量子概率搜索提高了算法的局部搜索能力。通过一组典型函数优化实验对该算法的性能进行了考察,并与 QEA 进行了比较。结果表明,本文算法在解的质量和收敛速度上都要优于 QEA。

关键词 量子进化算法,混合量子进化算法,函数优化

A New Hybrid Quantum Evolutionary Algorithm

XIE Ping LI Bin ZHUANG Zhen-Quan

(Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230027)

Abstract When quantum inspired evolution algorithm(QEA) is used for the optimization of continuous functions with many local optima, it's easy to be trapped into the local deceptive optima. In this paper, a new hybrid quantum evolution algorithm is proposed to overcome the shortcoming of the QEA. The new hybrid QEA combines the merits of classic GA and QEA by using double coding mechanism(classic binary coding and quantum probabilistic coding), and combining the classical crossover and quantum probabilistic search. It not only has the global searching capacity of classic GA, but also improves the local searching capacity of algorithm by using quantum probabilistic search. Experiments on 23 test functions of diverse complexities are implemented and compared with QEA in this paper. The result indicates that the new hybrid QEA is better than QEA in both quality of final result and the convergence rate.

Keywords Quantum inspired evolution algorithm, Hybrid quantum inspired evolution algorithm, Function optimization

1 简介

量子进化算法(QEA)^[1]是一种基于量子计算概念的进化算法,它吸取了量子计算中的叠加态等思想,采用与众不同的编码形式和交叉变异方法,在组合优化问题上取得了较好的实验效果。但是 QEA 在处理多峰函数优化,特别是高维多峰函数优化问题时,容易陷入局部最优,且运算效率不高。

针对 QEA 的上述不足,本文提出将遗传算法的全局优化能力与基于量子概率模型的局部搜索能力相结合,实现了一种新的混合量子进化算法。该算法采用二进制编码和量子概率向量编码的双编码机制,同时采用经典遗传算法的交叉算子和量子演化算法的更新策略,有效地提高了量子演化算法的全局搜索能力。在一组典型测试函数上的优化实验验证了本文方法的有效性。

本文后续内容安排如下:第 2 节简单介绍量子进化算法的基本原理;第 3 节介绍本文提出的混合量子进化算法;第 4 节介绍实验结果与分析,最后总结全文。

2 量子进化算法(QEA)

量子比特是量子计算中的基本信息单元。与经典比特不同的是,量子比特不仅表示 0 或 1,而且表示 0 和 1 的叠加态,

可以表示成: $|\Psi\rangle = \alpha|0\rangle + \beta|1\rangle$, 其中 $|0\rangle$ 和 $|1\rangle$ 分别表示经典比特中的 0 和 1, $|\alpha|^2$ 、 $|\beta|^2$ 分别是量子比特处于状态 0 和 1 的概率,满足 $|\alpha|^2 + |\beta|^2 = 1$ 。因此,一个量子比特可同时存储和表达两个态的信息。一个长度为 N 的量子比特串可同时存储和表达 2^N 个不同的解。

在 QEA^[1]中,第 t 代群体为 $Q(t) = \{q_1, q_2, \dots, q_n\}$, n 为群体规模。每个个体可以表示为

$$q_j = \begin{bmatrix} \alpha_{j1} & \alpha_{j2} & \dots & \alpha_{jm} \\ \beta_{j1} & \beta_{j2} & \dots & \beta_{jm} \end{bmatrix}, j=1, 2, \dots, n \quad (1)$$

其中 m 为每个个体的编码长度。

经典比特状态的变化可以用逻辑门描述,对量子比特也可以用量子门来改变状态。对于单量子比特的量子门,可以由 2×2 矩阵给出。因为量子状态 $\alpha|0\rangle + \beta|1\rangle$ 满足归一化条件 $|\alpha|^2 + |\beta|^2 = 1$, 量子门作用后的状态也要满足这个条件,所以要求量子门相应的矩阵 U 是酉性的,即 $U^+U = I$ 。

在 QEA 中,个体更新是用量子旋转门实现的,如下所示:

$$U(\Delta\theta_i) = \begin{bmatrix} \cos\Delta\theta_i & -\sin\Delta\theta_i \\ \sin\Delta\theta_i & \cos\Delta\theta_i \end{bmatrix} \quad (2)$$

其中, $\Delta\theta_i$ 为调整的旋转角,其取值事先设定,通过查表获得。

QEA 的一般步骤^[1]:

^{*})国家自然科学基金(60401015, 60572012)和安徽省自然科学基金(050420201)资助项目。解 平 硕士研究生,研究方向为演化算法、量子计算;李 斌 博士,副教授,研究方向为演化算法、量子计算、智能信息处理;庄镇泉 教授,博士生导师,研究方向为智能信息处理。

- (1) 初始化群体 $Q(t)$, $t=0$;
- (2) 由 $Q(t)$ 生成 $P(t)$;
- (3) 评价 $P(t)$ 的适应度, 并将 $P(t)$ 中最好的解存入 $B(t)$;
- (4) $t=t+1$, 由 $Q(t-1)$ 生成 $P(t)$, 并评估 $P(t)$;
- (5) 用量子门更新 $Q(t)$;
- (6) 把 $B(t-1)$ 和 $P(t)$ 中最好的解存入 $B(t)$, $B(t)$ 中最好的解存入 b ;
- (7) 判断是否满足终止条件, 若满足则算法结束, 若不满足, 则转到第 4 步。

3 混合量子进化算法

3.1 编码

传统遗传算法一般采用的编码方式是一个个体对应搜索空间中的一个候选解, QEA 中采用的是量子概率编码。而在混合量子进化算法中, 每个个体由两部分组成: 一个是二进制编码, 另一个是量子概率向量编码, 如下所示。

$$\begin{array}{ccccccccc} 0 & 1 & 1 & 0 & 0 & \cdots & & & \\ \alpha_1 & \alpha_2 & \alpha_3 & \alpha_4 & \alpha_5 & \cdots & & & \\ \beta_1 & \beta_2 & \beta_3 & \beta_4 & \beta_5 & \cdots & & & \end{array} \quad (3)$$

取二进制编码的目标函数值作为个体的适应度值。在交叉操作中, 两个向量作为一个整体参与交叉, 二进制编码同时作为量子概率向量的当前演化目标参与概率向量的演化更新中。

3.2 遗传操作

在混合量子进化算法中, 我们采用了与传统遗传算法相

同的遗传操作, 即选择、交叉和变异。类似于传统遗传算法, 选择就是从群体中根据个体的适应度选取个体放入配对池中, 以产生下一代个体。本文采用联赛选择, 即每次从父代群体中随即选出 T_n 个个体, 然后进行适应度比较, 把其中最好的个体放入配对池中。如此循环, 直到配对池满了为止。在联赛选择中, T_n 的选取起着很重要的作用: 大的 T_n 可以增强选择能力, 加快收敛的速度, 但是有早熟的风险; 小的 T_n 可以减少早熟的风险, 但是也减慢了收敛进程, 从而增加了运算的复杂性。所以要权衡 T_n 的取值, 本文中 T_n 取 2。

交叉作为一个主要的遗传操作, 其作用是将已有的优良基因遗传给下一代个体, 希望获得更好适应度的个体。交叉通过交换两个父代个体的遗传信息来产生新的个体。本文算法采用单点交叉(如图 1 所示), 从配对池中取两个父个体, 随机产生一个交叉点 s , 以这个交叉点为界, 互相交换一侧的染色体片断。由于个体编码由两部分组成(二进制编码和量子概率向量编码), 经交叉操作后, 两部分都发生了变化, 新个体的适应度值由二进制编码个体的目标函数值确定。交叉操作是在一个确定概率 p_c 下进行的, 文中实验 $p_c=0.85$ 。

变异是对染色体上某位基因发生突变, 以改变染色体的结构。其作用是增加群体的多样性, 以提高算法跳出局部最优的能力。本文算法采用单点变异策略(如图 2 所示), 即按均匀分布随机选择一位, 该位的二进制编码由 1 变成 0 或者由 0 变成 1, 对应的量子概率编码则 α 和 β 互换。变异也是在一个确定概率 p_m 下进行的。 p_m 应取一个较小的值, 否则会破坏个体中有利的信息, 使搜索趋向于随机搜索。文中实验 $p_m=0.02$ 。

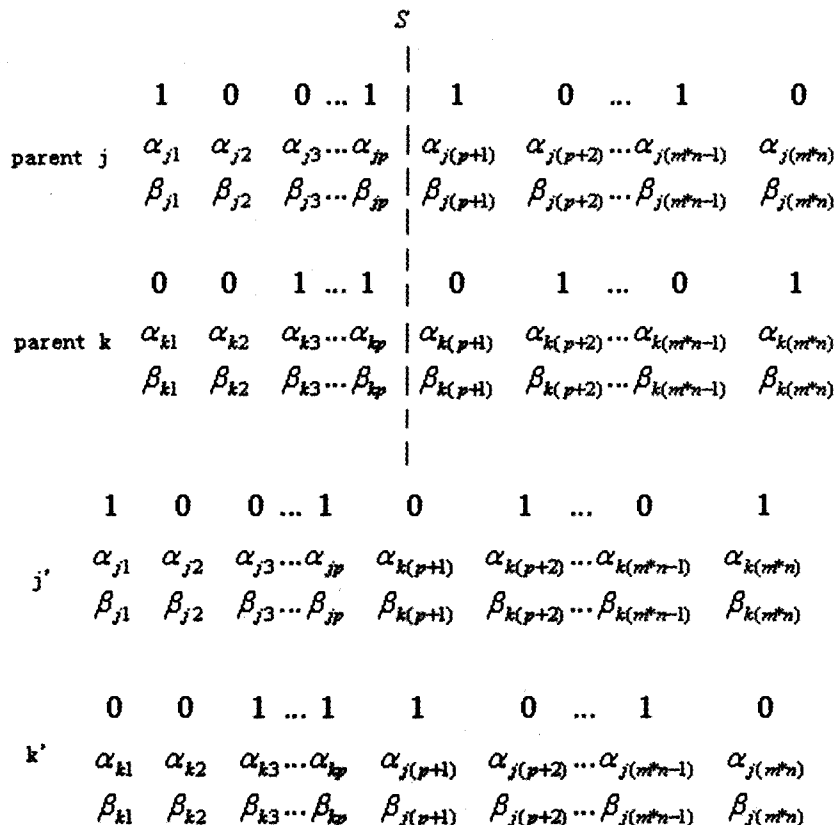


图 1 交叉操作

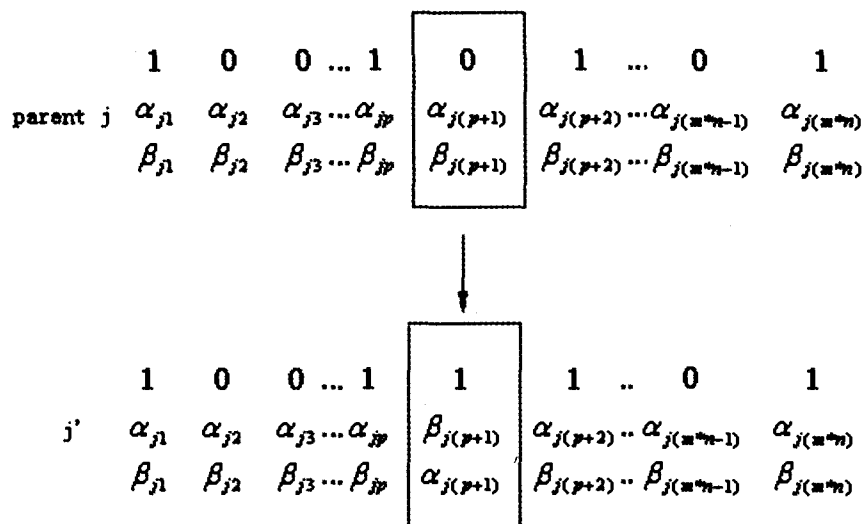


图2 变异操作

3.3 量子个体的更新

QEA的个体更新是每个个体相关位向当前最优解的方向旋转,旋转角度 $\Delta\theta$ 控制了算法的收敛速度; $\Delta\theta$ 大,则算法的收敛速度快,但是同时也很容易出现早熟现象,即陷入局部最优; $\Delta\theta$ 小,则收敛速度慢,算法的复杂度大。

本文算法采用如下更新策略:个体经过遗传操作后,对每个个体的概率向量采样 N 次,保存每个个体的最优解以及整个群体的当前最优解。每个个体有两次机会得到更好的解,分别是遗传操作和采样。然后利用这些信息对个体进行更

新,个体更新分为2个部分:

(1)全局更新:和QEA一样,每个个体中的概率向量向当前最优解旋转,旋转角为 $\Delta\theta_1$ 。

(2)局部更新:若每个个体采样得到的最好解没有原来这个个体的适应度高,即采样没有找到更好的解,则个体中的概率向量朝着其相反方向进行旋转,旋转角度为 $\Delta\theta_2$;若 N 次采样后找到了比原来这个个体适应度更高的解,则个体朝着这个更好的解旋转角度 $\Delta\theta_2$ 。

表1 测试函数

测试函数	n	S	$f_{\min}$
$f_1(x) = \sum_{i=1}^n x_i^2$	30	$[-100, 100]^n$	0
$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	$[-10, 10]^n$	0
$f_3(x) = \sum_{i=1}^n (\sum_{j=1}^n x_{ij})^2$	30	$[-100, 100]^n$	0
$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	30	$[-100, 100]^n$
$f_5(x) = \sum_{i=1}^n [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	30	$[-30, 30]^n$	0
$f_6(x) = \sum_{i=1}^n (4 - x_i + 0.5)^2$	30	$[-100, 100]^n$	0
$f_7(x) = \sum_{i=1}^n ix_i^4 + \text{random}[0, 1]$	30	$[-1.28, 1.28]^n$	0
$f_8(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	30	$[-500, 500]^n$	-12569.5
$f_9(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	$[-5.12, 5.12]^n$	0
$f_{10}(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos 2\pi x_i\right) + 20 + e$	30	$[-32, 32]^n$	0
$f_{11}(x) = 1/4000 \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(x_i/\sqrt{i}) + 1$	30	$[-600, 600]^n$	0
$f_{12}(x) = \pi/n [10 \sin^2(\pi y_i) + \sum_{i=1}^n (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1}) + (y_n - 1)^2] + \sum_{i=1}^n u(x_i, 10, 100, 4)]$	30	$[-50, 50]^n$	0
$f_{13}(x) = 0.1 \{\sin^2(3\pi x_1) + \sum_{i=1}^n (x_i - 1)^2 [1 + \sin^2(3\pi x_{i+1}) + (x_n - 1) [1 + \sin^2(2\pi x_n)]] + \sum_{i=1}^n u(x_i, 5, 100, 4)\}$	30	$[-50, 50]^n$	0
$f_{14}(x) = [1/500 + \sum_{i=1}^{25} (j + \sum_{i=1}^n (x_i - a_{ij})^6)^{-1}]^{-1}$	2	$[-65.536, 65.636]^n$	1
$f_{15}(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i (b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	4	$[-5, 5]^n$	$3.075e-4$
$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	-1.0316285
$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10(1 - 1/8\pi) \cos x_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^n$	3
$f_{19}(x) = -\sum_{i=1}^4 c_i \exp[-\sum_{j=1}^4 a_{ij} (x_j - p_{ij})^2]$	4	$[0, 1]^n$	-3.86
$f_{20}(x) = -\sum_{i=1}^6 c_i \exp[-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2]$	6	$[0, 1]^n$	-3.32
$f_{21}(x) = -\sum_{i=1}^5 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10
$f_{22}(x) = -\sum_{i=1}^4 [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10
$f_{23}(x) = -\sum_{i=1}^{10} [(x - a_i)(x - a_i)^T + c_i]^{-1}$	4	$[0, 10]^n$	-10

3.4 算法流程

- (1) 初始化群体 $Q(t)$, $t=0$;
- (2) 评价 $Q(t)$ 的适应度, 记录 $Q(t)$ 中每个个体的最好解;
- (3) $t=t+1$, 进行遗传操作(选择、交叉和变异);
- (4) 对每个个体中的概率向量进行采样;
- (5) 记录每个个体得到的最好解以及当前群体最优解;
- (6) 用量子门更新个体(包括全局更新和局部更新), 得到新个体, 产生 $Q(t)$;
- (7) 判断是否满足终止条件, 若满足则算法结束, 若不满足, 则转到第 3 步。

4 函数优化实验

4.1 测试函数

本文的函数优化实验是找函数的最小值, 可以归结为寻找 $f_{\min} \in S$, 满足

$$\forall x \in S: f(x_{\min}) \leq f(x) \quad (4)$$

这里 f 是一个有界的函数, S 是 f 的定义域。

本文使用文[3]中的 23 个测试函数对算法性能进行考察(如表 1 所示)。这些测试函数可分为 3 类: $f_1 \sim f_7$ 是高维单峰函数, $f_8 \sim f_{13}$ 是高维多峰函数, 有很多的局部最优解, 局部最优解的数量随着问题维数的增长呈指数增长。 $f_{14} \sim f_{23}$ 是低维多峰函数, 有少量的局部最优解。对于单峰函数相对比较容易优化, 主要检测函数的收敛速度。而多峰函数则比较困难, 最终的优化结果反映了一个算法摆脱局部最优寻找全局最优的能力。通过这 23 个测试函数, 可以比较全面地考察算法的性能。

4.2 实验参数设置

参数设置: 实验中, 对于每一维变量用 15 位的二进制数来编码, 群体规模为 50, 交叉概率 $P_c = 0.85$, 变异概率 $P_m = 0.02$, 联赛选择中每次选择个体的个数 $T_n = 2$, 采样次数 $N = 1$, 旋转角度 $\Delta\theta_1 = 0.01\pi$, $\Delta\theta_2 = 0.005\pi$ 。

实验结果与 QEA 比较, 如表 2 所示, 每个实验独立运行

表 2 新的 Hybrid QEA 和 QEA 优化结果的比较(运行 30 次的平均结果)

	fitness evaluation times($\times n$)	Hybrid QEA		QEA	
		mean best	std dev	mean best	std dev
f_1	3500	2.79e-4	8.45e-6	3.54e-4	7.32e-5
f_2	3500	0.1346	5.78e-3	0.549	4.98e-3
f_3	3500	2.79e-4	3.35e-6	3.54e-4	3.42e-2
f_4	3500	0.954	0.327	13.15	0.265
f_5	9000	20.5	21.6	27.8	32.5
f_6	1500	0	0	0	0
f_7	1000	0.00797	0.0048	0.026	0.0073
f_8	2000	-10834	46.071	-11422.1	102.7
f_9	2000	47.58	8.51	54.74	17.4
f_{10}	2000	0.351	0.0779	2.013	0.486
f_{11}	2000	0.0821	0.0848	0.197	0.123
f_{12}	4000	0.653	2.47e-2	0.770	0.494
f_{13}	3000	0.307	0.5227	1.58	0.548
f_{14}	90	1.2107	0.0958	1.17	0.156
f_{15}	480	1.79e-3	3.757e-4	8.74e-4	4.46e-4
f_{16}	90	-1.0312	4.072e-4	-1.025	0.0126
f_{17}	90	0.399	3.48e-4	0.399	1.24e-3
f_{18}	90	3.00	2.46e-5	3	2.33e-5
f_{19}	60	-3.86	3.82e-6	-3.86	5.75e-5
f_{20}	300	-3.22	0.053	-3.23	0.048
f_{21}	240	-6.63	1.64	-5.03	2.68
f_{22}	300	-6.87	2.74	-5.52	2.73
f_{23}	300	-6.69	2.699	-5.93	3.36

30 次, 得到平均最优解和标准差, 平均最优解表示一个算法的最终优化结果, 30 次最优解的标准差衡量算法的一致性, 标准差小说明算法一致性越好。

4.3 实验结果

在同等适应度评估次数下对 23 个函数进行优化实验, 并与 QEA 比较, 实验结果如表 3 所示, 部分函数的收敛曲线见图 3。

由实验结果可见, 对于单峰函数 $f_1 \sim f_7$, HQEA 性能优于 QEA, 尤其是 f_3 和 f_7 ; 对于高维多峰函数 $f_8 \sim f_{13}$, 除了 f_8 之外, HQEA 也优于 QEA; 对于低维多峰函数 $f_{14} \sim f_{23}$, HQEA 性能与 QEA 接近。

总结 一般来说, 进化算法具有两个方面的能力: 开发能力和探索能力。开发能力是算法利用现有积累的信息寻找最优解的能力, 探索能力是算法探索未知空间的能力。一个好的进化算法在处理开发和探索之间的矛盾时须采取折中的策略, 既要利用现有信息开发当前空间, 也要兼顾探索未知空间。也就是既要加快收敛速度, 又要避免陷入局部最优。

QEA 中, 所有的个体都朝着当前最优解按照同样的量子门旋转, 这难免会导致进化过程后期搜索空间会局限在一个很小的范围, 从而跳不出局部最优。本文提出的混合量子进化算法 HQEA 结合了传统遗传算法和 QEA 的优点, 所以提高了搜索效率, 同时也提高了算法跳出局部最优的能力。

参考文献

- 1 Han Kuk-Hyun, Kim Jong-Hwan. Quantum-Inspired Evolutionary Algorithm for a Class of Combinatorial Optimization. IEEE Transactions on Evolutionary Computation, 2002, 6(6): 580~593
- 2 Tu Zhenguo, Lu Yong. A Robust Stochastic Genetic Algorithm (StGA) for Global Numerical Optimization. IEEE Transactions on Evolutionary Computation, 2004, 8(5): 456~470
- 3 Yao Xin, Liu Yong, Lin Guangming. Evolutionary Programming Made Faster. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82~102
- 4 Grover L K. A fast quantum mechanical algorithm for database search. In: Proc. 28th ACM Symp Theory of Computing, 1996. 212~219
- 5 Shor P W. Quantum computing. Doc Mathematica, vol. Extra Volume ICM, 467~486
- 6 Yao Xin. Getting Most out of Evolutionary Approaches. In: Proceedings of the 2002 NASA/DOD Conference on Evolvable Hardware, 2002. 8~14
- 7 Talbi H, Draa A, Batouche M. A New Quantum-inspired Genetic Algorithm for Solving the Traveling Salesman Problem. In: 2004 IEEE International Conference on Industrial Technology(ICIT), December, 2004. 8~10
- 8 Narayanan A, Moore M. Quantum-inspired genetic algorithms. In: Proceedings of the 1996 IEEE International Conference on Evolutionary Computation(ICEC96), Nagaya, Japan, 1996. 41~46
- 9 Han K H, Kim J H. On Setting the Parameters of Quantum-inspired Evolutionary Algorithm for Practical Application. In: Proc. 2003 Congress on Evolutionary Computation. Piscataway NJ: IEEE Press, 2003
- 10 Yuan Bo, Gallagher M. On the Importance of Diversity Maintenance in Estimation of Distribution Algorithms. In: Proceedings of the 2005 Genetic and Evolutionary Computation Conference, ACM, Washington D C, 719~726

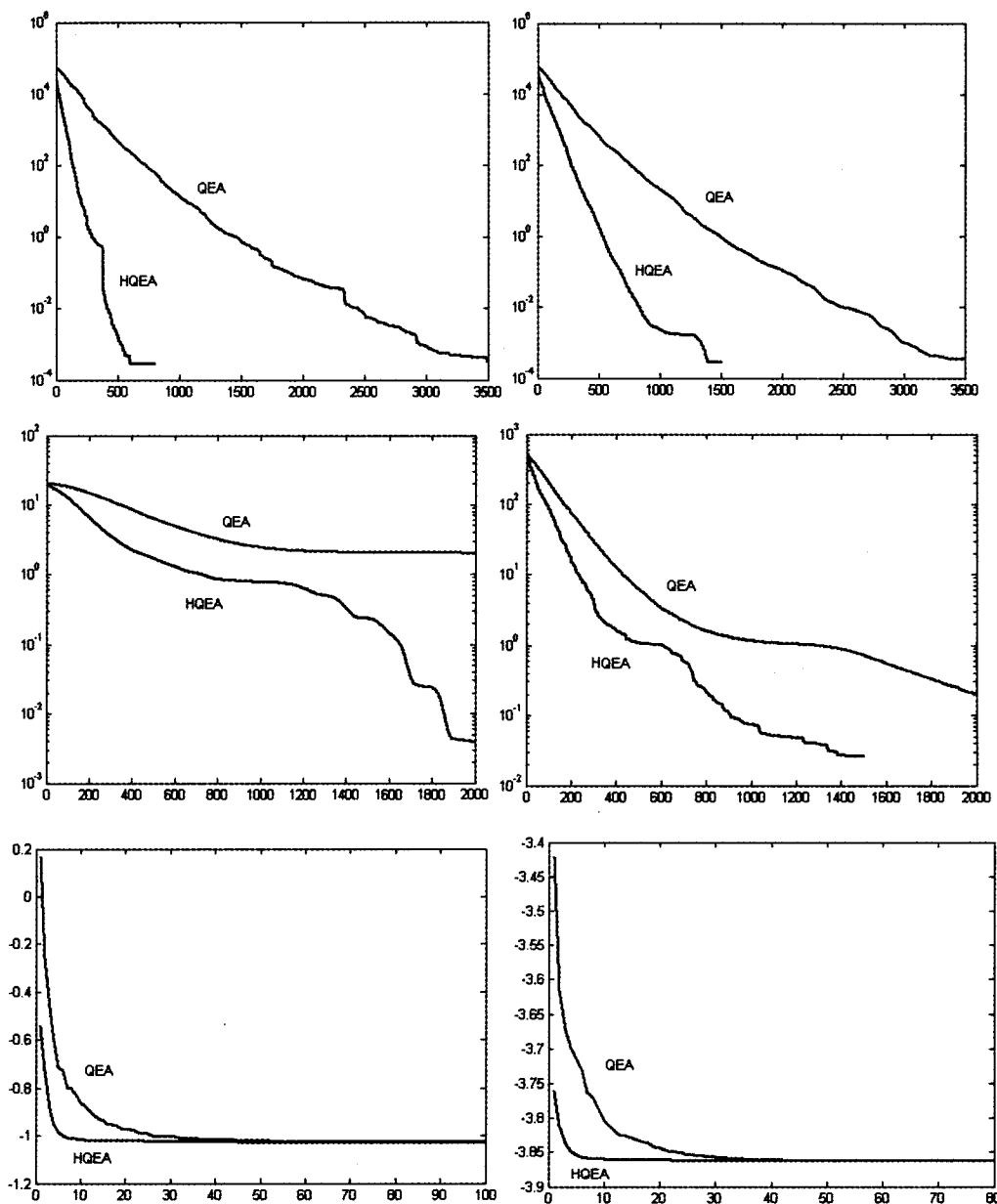


图3 部分函数的收敛曲线
(运行30次后的平均结果,横坐标是演化代数,纵坐标是适应度值)

(上接第156页)

(T_1)偏低,说明学生意见较大,需尽快改进,学校人事处应该尽快安排老师轮流外出进修来提高教师的专业素质,同时要多招些新教师。总之,在课堂教学质量评估指标体系中,教学内容和教学方法两方面非常重要,每位教师都应引起高度重视。显然这与已有的研究成果和实际相吻合。

结束语 本文使用模糊数据挖掘发现评估数据库中教师课堂教学质量评估等级同评估指标之间的规则知识,依据该规则知识对挖掘结果进行有效的评价,并且在分析、预测方面有着很大的优势,从而帮助决策者做出决策。当然对于该教学质量数据挖掘来说,这只是一部分工作,如何进一步优化该

系统是下一步研究的主要工作。

参考文献

- 1 Luo J. Integrating Fuzzy Logic with datamining Methods for intrusion Detection [D]. Mississippi State University, 1999
- 2 王景英. 教育评价理论与实践[M]. 长春: 东北师范大学出版社, 2002
- 3 李鸿吉. 模糊数学基础及实用算法[M]. 科学出版社, 213~215
- 4 谢季坚. 模糊数学方法及其应用[M]. 华中科技大学出版社, 96~97